

**Федеральное государственное бюджетное образовательное учреждение
высшего образования**

«Российский государственный гуманитарный университет»

Институт информационных наук и технологий безопасности



На правах рукописи

Моляков Андрей Сергеевич

**МОДЕЛИ И МЕТОДЫ ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ
БЕЗОПАСНОСТИ СТАЦИОНАРНЫХ И БОРТОВЫХ
СУПЕРКОМПЬЮТЕРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ**

Специальность 2.3.6 – Методы и системы защиты информации,
информационная безопасность

**Диссертация на соискание ученой степени
доктора технических наук**

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
ГЛАВА 1. АНАЛИЗ МЕТОДОВ И ТЕХНОЛОГИЙ ЗАЩИТЫ ИНФОРМАЦИИ СУПЕРКОМПЬЮТЕРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И ПОСТАНОВКА ЗАДАЧ ИССЛЕДОВАНИЙ.....	36
1.1 Анализ особенностей организации защиты информации в стационарных и бортовых суперкомпьютерах	36
1.2 Обзор инновационных проектов в области создания суперкомпьютерных платформ	44
1.3 Проблемы и недостатки применения традиционных методов и технологий защиты информации в вычислительных комплексах, использующих суперкомпьютерные технологии	56
1.4 Тактико-технические характеристики перспективных суперкомпьютеров эволюционного и революционного путей развития технологий	70
1.5 Постановка задач исследований.....	82
ГЛАВА 2. ТЕОРЕТИЧЕСКИЕ ПОДХОДЫ И ПРИНЦИПЫ В ОБЛАСТИ СОЗДАНИЯ ЗАЩИЩЕННЫХ СТАЦИОНАРНЫХ И БОРТОВЫХ СУПЕРКОМПЬЮТЕРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ С ИСПОЛЬЗОВАНИЕМ СРЕДСТВ ВИРТУАЛИЗАЦИИ	84
2.1 Алгебра операций с объектами, обрабатываемыми на суперкомпьютерах.	84
2.2 Оптимизация, нормировка характеристической функции вычисления дескрипторов и модальные правила политик безопасности.....	101
2.3 Интеграция алгебраической структуры с архитектурными доработками суперкомпьютеров революционного пути развития	107
2.4 Выводы по разделу	114
ГЛАВА 3. МОДЕЛИ БЕЗОПАСНОЙ ОБРАБОТКИ ИНФОРМАЦИИ С УЧЕТОМ ОСОБЕННОСТЕЙ СТАЦИОНАРНЫХ И БОРТОВЫХ СУПЕРКОМПЬЮТЕРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ	115
3.1 Базовые характеристики современных суперкомпьютеров и проблемы их достижения.	115

3.2	Модель угроз целостности среды исполнения процессов суперкомпьютеров с реализацией новой парадигмы написания правил политики безопасности и новых сущностей – «субъект», «объект» и «дескриптор оценки безопасности состояний»	124
-----	--	-----

3.3	Описание информационных процессов с использованием модели безопасных операций.....	149
-----	--	-----

3.4	Выводы по разделу:	163
-----	--------------------------	-----

ГЛАВА 4. МЕТОДЫ ГИБРИДНОЙ (РЕАКТИВНОЙ И ПРОАКТИВНОЙ) ЗАЩИТЫ СТАЦИОНАРНЫХ И БОРТОВЫХ СУПЕРКОМПЬЮТЕРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ С УЧЕТОМ САМООРГАНИЗАЦИИ И АДАПТАЦИИ К ТИПУ КОНФИГУРАЦИИ И ПОТОКУ ОБРАБАТЫВАЕМЫХ ДАННЫХ	165
--	------------

4.1	Пример оценки результативности новых решений и технологий.....	167
-----	--	-----

4.2	Разработка прототипа с реализацией новых архитектурных принципов проектирования и механизмов интегрированной защиты суперкомпьютерных комплексов для АС разного класса с гибкой конфигурацией окружения и контекстом вызовов сервисов, сбором метрики приложений.....	174
-----	---	-----

4.3	Метод реактивной защиты, основанный на виртуализации среды исполнения процессов и контроле запросов на выделение ресурсов в соответствии с правилами политики безопасности в виде модальных структур темпоральной логики .	181
-----	--	-----

4.4	Метод реконфигурации среды выполнения программ проактивной защиты, основанный на парадигме темпоральной логики с вычислением доверительная вероятности в качестве показателя защищенности	199
-----	---	-----

4.5	Выводы по разделу:	217
-----	--------------------------	-----

ГЛАВА 5. РАЗРАБОТКА ПРОГРАММНО-ТЕХНИЧЕСКИХ РЕШЕНИЙ ПО ПОВЫШЕНИЮ УРОВНЯ ЗАЩИЩЕННОСТИ СОВРЕМЕННЫХ СУПЕРКОМПЬЮТЕРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ В ВИДЕ МНОГОМОДУЛЬНОГО ЗАЩИЩЕННОГО МУЛЬТИДОМЕННОГО ГИПЕРВИЗОРА.....	219
--	------------

5.1	Краткое описание экспериментального стенда	219
-----	--	-----

5.2	Повышение уровня безопасности вычислительных и сервисных узлов суперкомпьютеров при использовании традиционных средств защиты за счет реализации методики тестирования с использованием средств виртуализации.....	221
-----	--	-----

5.3	Синтез структуры системы контроля и реализация алгоритма идентификации угроз на основе набора маркеров в многоуровневой песочнице под управлением защищенного мультидоменного гипервизора.....	227
5.4	Разработка и реализация программно-технических решений, результаты тестовых испытаний	234
5.5	Выводы по разделу:	257
ЗАКЛЮЧЕНИЕ		259
ПЕРЕЧЕНЬ СОКРАЩЕНИЙ.....		264
СПИСОК ЛИТЕРАТУРЫ		266
ПРИЛОЖЕНИЕ 1 СВИДЕТЕЛЬСТВО О РЕГИСТРАЦИИ ПРОГРАММЫ ЭВМ.....		281
ПРИЛОЖЕНИЕ 2 ПАТЕНТЫ НА ИЗОБРЕТЕНИЯ		282
ПРИЛОЖЕНИЕ 3 СХЕМА И НАБОР ИНСТРУМЕНТАЛЬНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ЭКСПЕРИМЕНТАЛЬНОГО СТЕНДА.....		284
ПРИЛОЖЕНИЕ 4 СПРАВКИ И АКТЫ О ВНЕДРЕНИИ		289

ВВЕДЕНИЕ

Актуальность работы. В данной работе рассмотрены два типа суперкомпьютеров: бортовые (например, “Трифон” или “Комдив”) – это компактные высокопроизводительные вычислительные платформы для работы в режиме реального времени и жестких условиях эксплуатации на борту летательных аппаратов, стационарные (“Ангара” ЕС1740, СТ-2, Tsubame 3.0) – это высокопроизводительные вычислительные платформы для промышленных объектов, центров обработки данных и т.п. Профиль защиты зависит от типа и архитектуры суперкомпьютера. По сравнению со стационарными суперкомпьютерами бортовые супер-ЭВМ обладают ограниченными ресурсами (оперативная память, кол-во ядер процессора, кол-во вычислительных блоков), поэтому набор функций безопасности и режимов защиты для них базовый (минимальный). Для стационарных суперкомпьютеров в зависимости от набора конфигураций заданы три профиля защиты: для малого промышленного кластера уровень защиты средний, для суперкластера крупного технологического объекта – уровень защиты высокий, для масштабных разведывательных или информационно-аналитических центров – уровень защиты максимальный.

В настоящее время в России различными предприятиями ведутся работы по созданию и поставкам вычислительных систем кластерного типа высокой производительности, основу которых составляют коммерчески доступные компоненты и сборочные единицы импортного производства. Подавляющее большинство аппаратных средств, используемых для создания супер-ЭВМ, зарубежного производства, либо сборка конструктивов осуществляется по иностранной лицензии на территории РФ, реализуемая обвязка состоит из недоверенных элементов, импортируемых из-за рубежа.

Высокопроизводительная элементная база вместе с поставляемым системным программным обеспечением (ПО) суперкомпьютеров имеет огромное количество компонентов и представляет собой недоверенную аппаратуру, что повышает вероятность применения средств скрытого информационного воздействия (ССИВ) и приводит к изменению функциональности, утечке конфиденциальной информации или нарушению работы суперкомпьютеров. Факты обнаружения вредоносного аппаратного и программного обеспечения в США, России и других странах только подкрепляют эти опасения. При проведении установки, наладки, монтажа, пусконаладочных и иных видов работ, при изменении конфигурации или при обновлении и установке дополнительных программных, технических и программно-технических средств высока угроза внедрения вредоносного ПО локально или по сети. В результате выполнения непреднамеренных или умышленных действий со стороны пользователей, администраторов, обслуживающего технического персонала также высока вероятность выведения из строя либо простоя в функционировании, что недопустимо для современных суперкомпьютеров при выполнении разного класса инженерно-расчетных высоконагруженных задач. В качестве примера можно привести доклад группы исследователей из компании Positive Technologies, специализирующейся в области информационной безопасности, которая в 2017 году обнаружила уязвимость в модуле Intel Management Engine, существующую с 2008 года, которая позволила получить полный доступ к системе и установить вредоносное программно-аппаратное обеспечение, которое не удавалось обнаружить в течение 9 лет.

Актуальность исследования в области создания защищенных национальных суперкомпьютерных платформ подтверждается и «Приоритетными направлениями развития науки, техники и технологий Российской Федерации», утвержденными Указом Президента РФ от 16 декабря 2015 г. № 623.

Следует отметить, что с 2018 года в России действует закон 187-ФЗ «О безопасности критической информационной инфраструктуры», обязывающий организации обеспечить комплексную защиту объектов КИИ, неотъемлемой и важной частью которых являются суперкомпьютеры (СК), специфика которых заключается в следующем: суперкомпьютеры обладают высочайшей производительностью, большим количеством одновременно выполняемых процессов (от сотни тысяч – до миллиона), что определяет трудность отслеживания происходящих в них событий с целью обнаружения вторжений. Высокая параллельность, асинхронность, огромное кол-во процессов, динамически меняющийся контекст выполняемых операций, сложная сборка инструкций и иерархия выполнения команд, Петабайты данных, которые нужно обрабатывать в режиме реального времени, глобально адресуемая память – все это приводит к тому, что сегменты памяти данных и программ расплываются по огромной “памяти распределенных процессов”, нет точной привязки к доменам защиты, число уровней иерархии обработки запросов не идентифицировано, не соответствует числу уровней защиты, нет интервальных ограничений фильтрации в зависимости от класса и архитектуры СК.

Обрабатываемая с использованием суперкомпьютеров информация обычно связана с решением вопросов национальной безопасности и решением важнейших научно-технических задач, осуществлением управления критически важными инфраструктурами и т.д. Это серьезная мотивация для организации атак. Пользователи суперкомпьютеров имеют высокую квалификацию и их обычно много, что повышает риски внутренних атак. Высокой квалификацией обладают заинтересованные в атаках на суперкомпьютерные ресурсы злоумышленники, обычно представляющие организованные сообщества, поддерживаемые на государственном уровне.

Например, 17 июля 2020 года группа реагирования на инциденты информационной безопасности (CSIRT) опубликовала отчет, в котором говорилось, что под ударом тщательно спланированных компьютерных атак оказались Стэнфордский национальный центр атмосферных исследований, Суперкомпьютерный центр в Сан-Диего и Национальный центр суперкомпьютерных приложений при Иллинойском университете, Суперкомпьютерный центр в Лос-Аламос и др. В России также наблюдается взрывной рост кибератак на суперкомпьютерные центры Яндекса и Сбербанка, кардинальное увеличение их мощности и использование новых тактик со стороны злоумышленников.

Более того, существующие и успешно внедрённые программно-технические решения промышленного комплекса России ориентированы на традиционные вычислительные кластеры и не учитывают специфики суперкомпьютеров, о которой говорилось ранее. В качестве примера можно привести реализацию средств защиты информации Ростатома, в основе которой лежат только базовые программно-технические компоненты на базе Scientific Linux для платформы x86_64. При этом не учитываются особенности работы процессоров и контроллеров с разными уровнями иерархии команд.

Однако, как показали результаты проведенных научно-исследовательских и опытно-конструкторских работ китайских специалистов из Национального суперкомпьютерного центра Чанг-Ша и японских специалистов из Национального суперкомпьютерного центра Осаки, без интеграции с аппаратной транзакционной памятью и введения многоуровневого контроля невозможно создать доверенную среду с использованием широко распространенных коммерческих программно-аппаратных средств и решить задачу обнаружения и идентификации разного типа угроз на всех уровнях иерархии выполнения запросов СК.

Следует отметить, что для создания подсистемы обеспечения безопасности серверов и рабочих станций разработчики обычно используют: проактивные методы, основной целью которых является предотвращение заражения системы, и реактивные методы, суть которых заключается в сигнатурном поиске уже известного вредоносного ПО.

Переходя на уровень инженерных решений для суперкомпьютерных платформ, производители оборудования реализовали для системных разработчиков встроенную полнофункциональную поддержку технологии виртуализации для разного набора микропроцессоров и микроконтроллеров. Других базовых (штатных) функций безопасности у аппаратных модулей и сборочных конструктивов, из которых собираются компоненты СК, просто нет. С использованием технологии виртуализации можно реализовать принципы кибериммунитета, основанные на разбиении системы на изолированные домены безопасности, многоуровневом контроле всех взаимодействий между доменами безопасности, обеспечении гарантированной защищенности вычислительной среды. Для каждого взаимодействия доменов безопасности монитор обращений возвращает решение (булево значение) в виде функции оценки безопасности состояний СК, соответствует ли это взаимодействие политике безопасности. Из всего выше сказанного, можно сделать вывод о том, что современных условиях средств защиты суперкомпьютеров разрабатываются с использованием виртуализации оборудования, поскольку она позволяет повысить гибкость использования и функционирования оборудования, обеспечивая этим совместное функционирование разных операционных систем, поддержку технического парка ЭВМ с разной архитектурой сборки, обеспечить реконфигурируемость не только для восстановления после сбоев и отказов аппаратных средств, но и после компьютерных атак. Перечисленные особенности определяют научную проблему создания новой методологии обнаружения угроз СК.

Основу подхода составляют метод реактивной защиты и метод реконфигурации среды проактивной защиты, позволяющие обнаруживать любые типы угроз на всех уровнях иерархии выполнения запросов и противодействовать им путем автоматической реконфигурации структуры конфигурации исполняемой среды СК на недоверенной аппаратуре.

Степень разработанности темы исследования. Известно значительное число работ, посвященных созданию защищенных информационных систем. Например, в работах В.А. Курбатова, П.Д. Зегжды, А.А. Гладких разработаны и проиллюстрированы методы и подходы, предназначенные для защиты только отдельных серверов виртуализации в составе производительных кластеров. В исследованиях, проведенных А.А. Грушо, В.С. Заборовским, С. Воглом, Р. Сэйлером, Ф. Мортинелли, Дж. Рутковской, предложены и разработаны принципы и методы в области обеспечения защиты на основе мандатных и ролевых политик безопасности, методы обнаружения средств скрытого информационного воздействия для киберфизических систем, построенных на базе рабочих станций и серверов предыдущего поколения, без учета специфики современных суперкомпьютеров. Работы О.В. Казарина и В.Ю. Скибы посвящены созданию методов и средств проактивной защиты программного обеспечения, однако авторам так и не удалось решить проблему реализации доверенной среды выполнения программ на недоверенном оборудовании для суперкомпьютерных систем. Таким образом, можно сделать вывод о том, что тема создания средств обеспечения информационной безопасности суперкомпьютерных вычислительных систем на основе разработки моделей и методов гибридной защиты за счет использования средств виртуализации (гипервизоров), является актуальной, новой и пока нерешенной, исходя из анализа работ зарубежных и российских специалистов.

При работе на недоверенной аппаратуре для создания доверенной программной среды предложено использовать компонент – верификатор команд процессора, а интегрированная защита суперкомпьютеров реализована в виде многомодульного мультидоменного гипервизора в связке с контроллерами транзакционной памяти.

Факторы доверительности и допущения, делающие недоверенную среду функционирования защищенной, следующие:

- 1) Объем кода гипервизора невелик по сравнению с ОС и приложениями, поэтому в нем проще обеспечить отсутствие уязвимостей. Все действия и события внутри виртуализованной системы обратимы – атаки могут быть оперативно нейтрализованы путем отката или сброса в исходное состояние.
- 2) Контроль обмена виртуализованной системы с внешней средой со стороны гипервизора позволяет отказаться от досконального исследования самих сред исполнения.
- 3) Аппаратная транзакционная память гарантирует атомарность и изолированность параллельно выполняемых задач, а также привязывает сегменты данных и программ к определенному домену защиты. Также поддерживается мультидоменный режим (набор разных доменов защиты с учетом уровней привилегий).
- 4) Мониторинг событий на всех уровнях иерархии выполнения запросов СК. Все операции транслируются гипервизором и контролируются верификатором команд. Параллельные процессы не могут «расплываться» по всем участкам памяти. Операции выполняются в определенном режиме работы процессора и обращаются только к выделенным им сегментам памяти данных и программ, а не случайно перемещаются по всей оперативной памяти.

- 5) Функционирование модуля верификации реализовано на более привилегированном уровне, чем ядро управляющей ОС и монитор виртуальных машин, благодаря чему осуществляется работа гипервизора напрямую с контроллерами транзакционной памяти и процессорами.
- 6) Взаимодействие процессов пользователя происходит с компонентами гипервизора, который создает изолированную среду исполнения. Компоненты гипервизора контролирует обращения к объектам файловой системы и стекам сетевых протоколов, осуществляют контроль последовательности выполнения запросов на выделение вычислительных ресурсов на всех уровнях иерархии. Изолированность (изолированная программная среда) гарантирует то, что параллельно выполняемые задачи не влияют на результат транзакции, обеспечивается надежная изоляция адресного пространства разных пользователей.
- 7) При прохождении запросом уровня защиты он не выполняется, а только происходит планирование действий для него на этом уровне и фиксация информации о контексте их выполнения, производится соответствующее этому изменению тега запроса и задаются интервально-временные ограничения выполнения команд.
- 8) По мере продвижения запроса через уровни в теге накапливается информация в виде его межуровневой трассы. Работа с тегами запросов на разных уровнях защиты производится модулями гипервизора. Атрибуты доступа к объекту и привилегии субъекта, связи между ними формализуются в виде набора (конъюнкции) предикатов. Отслеживать взаимодействие и контролировать доступ можно по набору характерных признаков – маркеров, представленных в виде кортежей логических переменных.

- 9) При изменении значений переменных и интервально-временных ограничений в указанной логической иерархической структуре, меняются идентификация вершин трассы и нумерация маршрута прохождения, что свидетельствует о подозрительной активности в системе.
- 10) Требования политики безопасности формулируются в терминах, которые задают последовательность обработки операций и ограничений на возможность повышения привилегий процессов. Ограничение на повышение привилегий и контроль контекста операций задаются значениями функции оценки безопасности.

Объект исследования: системы защиты суперкомпьютерных вычислительных систем в стационарном и бортовом исполнении.

Предмет исследования: модели, методы и алгоритмы обеспечения информационной безопасности суперкомпьютерных вычислительных систем с применением средств виртуализации.

Цели и задачи диссертационной работы.

Цель диссертационной работы: получение и поддержание максимального уровня гарантированной защищенности суперкомпьютеров на основе разработки моделей и методов гибридной (реактивной и проактивной) защиты за счет использования гипервизоров, обеспечивающих надёжную защиту от угроз информационной безопасности, реализуемых с использованием средств скрытого информационного воздействия.

В соответствии с этим были поставлены и решены следующие **основные задачи работы:**

- 1) Разработка теоретических и научно-методических принципов защиты суперкомпьютеров с использованием средств виртуализации.
- 2) Разработка модели угроз целостности среды выполнения процессов с учетом специфики работы суперкомпьютеров.
- 3) Разработка модели безопасных операций, которая учитывает новый класс угроз, связанных с неоднозначностью состояний выполнения прикладных и системных программ при работе в недоверенной среде суперкомпьютеров.
- 4) Разработка и исследование метода реактивной защиты суперкомпьютеров, заключающегося в виртуализации среды выполнения процессов на недоверенном оборудовании.
- 5) Разработка и исследование метода реконфигурации среды выполнения программ проактивной защиты суперкомпьютеров с учетом требований мобильности (встроенности производимых вычислений) и факторов доверительности.
- 6) Разработка методики тестирования уровня защищенности суперкомпьютеров, основанной на алгоритме “маркерного” сканирования.
- 7) Разработка программно-технических решений, обеспечивающих надёжную защиту от угроз информационной безопасности, реализуемых с использованием средств скрытого информационного воздействия.

Соответствие паспорту специальности. Проведенное исследование и полученные в его ходе результаты соответствуют п. 3 «Методы, модели и средства выявления, идентификации и классификации угроз нарушения информационной безопасности объектов различного вида и класса» (в части

разработки моделей, методов и средств идентификации угроз информационной безопасности суперкомпьютеров) и п. 15 «Принципы и решения (технические, математические, организационные и др.) по созданию новых и совершенствованию существующих средств защиты информации и обеспечения информационной безопасности» (на примере создания новых принципов и инновационных технологических решений для подсистемы обеспечения безопасности российских и зарубежных суперкомпьютерных платформ на базе модулей «Альфа-монитор») определения специальности 2.3.6 «Методы и системы защиты информации, информационная безопасность» паспорта специальностей ВАК (технические науки).

Методы исследования. Для решения поставленных задач в диссертационной работе использовались методы дискретной и вычислительной математики, теории графов, теории вероятностей, теории защиты информации, методы математического и численного моделирования, системный анализ, статистические методы (в частности корреляционный анализ), научный аппарат, основанный на принципах и положениях темпоральной логики.

Достоверность полученных результатов обусловлена корректным применением методов исследований и подтверждается результатами практического использования, а также соответствующими актами внедрения (реализации), схемами и чертежами, их внутренней непротиворечивостью и адекватностью физическим представлениям об исследуемом процессе. Предложенные в диссертационной работе результаты теоретически обоснованы и не противоречат известным и достоверно подтверждённым результатам исследований других авторов.

Научная проблема: для инженерии и отрасли промышленности на примере создания защищенных суперкомпьютерных систем известна как

проблема супервентности: с учетом высокой асинхронности и многосвязности процессов, огромного объема обрабатываемых данных нет логического соответствия между изменениями программ среды исполнения СК и изменениями компонентов аппаратуры с точки зрения реализации изолированной среды и контроля правил политик безопасности (ПБ). Следовательно, для создания доверенной среды выполнения программ суперкомпьютеров на недоверенном оборудовании, обеспечивающей надёжную защиту от угроз информационной безопасности, реализуемых с использованием ССИВ, необходимо разработать и внедрить механизмы многоуровневого контроля: любые преобразования, трансляции операций гипервизором при взаимодействии с аппаратурой должны быть безопасными, с точки зрения реализации механизмов идентификации и верификации.

Результаты, выносимые на защиту:

- 1) Теоретические и научно-методические принципы защиты стационарных и бортовых суперкомпьютеров с использованием средств виртуализации.
- 2) Модель безопасных операций, основанная на декомпозиции информационных процессов в виде 8-уровневой иерархической структуры и принципах кибериммунитета.
- 3) Модель угроз целостности среды выполнения процессов, основанная на новой парадигме написания правил политики безопасности.
- 4) Метод реактивной защиты суперкомпьютеров, основанный на виртуализации среды выполнения процессов, если вычисленный дескриптор состояния попадает в зону «риска».

- 5) Метод реконфигурации среды выполнения проактивной защиты суперкомпьютеров, основанный на парадигме темпоральной логики для проверки истинности принимаемых решений на структурах Крипке.
- 6) Программно-технические решения, основанные на разработанных моделях и методах.

Научная новизна. В диссертации получены следующие новые научные результаты:

- 1) Разработаны теоретические и научно-методические принципы защиты суперкомпьютеров с использованием средств виртуализации, *основанные* на построении и формализации универсальных и равномерных функций оценки безопасных состояний СК для заданных классов операций, *отличающиеся* нахождением в n-мерной алгебраической системе инварианта, *позволяющие* решить научную проблему супервентности для отрасли промышленности. Судя по открытым международным и отечественным источникам, исследования суперкомпьютеров в качестве объектов защиты проводятся впервые.
- 2) Разработана модель угроз целостности среды выполнения процессов, *основанная* на новой парадигме написания правил политики безопасности, *отличающаяся* заданием сущностей вместо классической связки «субъект, объект, предикат» – «субъект», «объект» и «дескриптор оценки безопасности состояний», *позволяющая* описать и формализовать каждую i-ю угрозы аргументов в виде конъюнкции предикатов из восьми логических переменных.

- 3) Разработана модель безопасных операций, *основанная* на декомпозиции информационных процессов в виде 8-уровневой иерархической структуры и принципах кибериммунитета, *отличающаяся* оценкой безопасности состояний в виде логической функции и проверки истинности принимаемых решений на каждой возможной интерпретации с учетом специфики суперкомпьютеров в виде временной и пространственной сложности, *позволяющая* идентифицировать и блокировать разные классы уязвимостей на всех уровнях иерархии выполнения запросов, что подтверждается сформулированным условием разрешимости и доказанной теоремой.
- 4) Разработан и исследован метод реактивной защиты суперкомпьютеров, *основанный* на виртуализации среды выполнения процессов, если вычисленный дескриптор состояния попадает в зону «риска», *отличающийся* применением траекторий вычислений дескрипторов оценки безопасности состояний на структурах Крипке, *позволяющий* создать единую технологию выявления уязвимостей и реализовать полный контроль контекста и тайминга выполняемых операций на недоверенной аппаратуре.
- 5) Разработан и исследован метод реконфигурации среды выполнения проактивной защиты суперкомпьютеров, *основанный* на парадигме темпоральной логики для проверки истинности принимаемых решений на структурах Крипке, *отличающийся* вычислением доверительная вероятности в качестве показателя защищенности компонентов среды на наборах тегированных данных в виде согласованной вариации факторных и результативных признаков, *позволяющий* на ранней стадии обнаруживать новые классы уязвимостей и противодействовать им путем автоматической реконфигурации структуры.

- 6) Разработана и реализована методика тестирования уровня защищенности, *основанная* на алгоритме “маркерного” сканирования, *отличающаяся* использованием набора меток: сочетаниям меток сопоставляются признаки скрытых угроз и действия, отвечающие требованиям выбранной политики информационной безопасности, *позволяющая* повысить уровень защищенности вычислительных ресурсов СК и минимизировать ошибки 1-го и 2-го рода.
- 7) Разработаны программно-технические решения, *основанные* на разработанных моделях и методах, *отличающиеся* минимальными затратами ресурсов для создания обучающих кластерных зон и на обучение по сравнению с существующими технологическими решениями на основе скрытых марковских моделей (с использованием алгоритма Баумана-Уэлча), *позволяющие* реализовать гарантированно защищенные системы, адаптированные к потоку входных данных и наборам конфигураций вычислительной среды.

Теоретическая значимость состоит в том, что разработанные и предложенные модели и методы, составляющие основу реактивной и проактивной защиты, развивают ряд положений теории информационной безопасности, базирующихся на формализации информационных процессов СК как «вычисляемая свертка состояний среды выполнения (аппаратной или виртуальной) и системного программного обеспечения», введены новые теоретические и научно-методические принципы для создания защищенных СК. Доказана теорема о равномерной перечислимости функций СК. Сформулировано условие разрешимости, как следствие из теоремы о равномерной перечислимости функций, проблемы установки логического соответствия (супервентности) процесса выполнения программы на суперкомпьютере, изменением состояний компонентов гипервизора и аппаратуры: при числе уровней иерархии меньше восьми процессы на всех

уровнях в СК не наблюдаемы, что затрудняет обнаружение и идентификацию угроз со стороны СЗИ. Правила политики безопасности задаются в виде модальных структур темпоральной логики STL, учитывающих контекст выполнения и тайминг операций (запросов). В качестве спецификации системы используется структура Крипке. Введено новое понятие “дескриптор оценки безопасности состояний системы”.

Введена новая характеристика для создания защищенных суперпроизводительных систем: коэффициент готовности к безопасной эксплуатации СК.

Практическая ценность заключается в разработке технологии и программно-технических средств на основе предложенных моделей и методов, составляющих принципы реактивной и проактивной защиты. С учетом специфики суперкомпьютеров предложено принципиально *новое технологическое решение*, создающее изолированную среду исполнения программ в виде 8-уровневой “песочницы” с реализацией контролирующих механизмов как на уровне гипервизоров, так и на уровне контроллеров транзакционной памяти. Технология следящих действий со стороны гипервизоров и управляющих ОС суперкомпьютеров позволяет по сравнению с существующими отечественными и зарубежными подходами значительно повысить уровень защищенности СК от разного типа угроз. Таким образом, решена важная научно-техническая проблема в области создания средств защиты информации для нового класса систем на примере суперкомпьютеров. Все существовавшие на сегодняшний день СЗИ являлись программно-техническими компонентами самой защищаемой системы, без интеграции с аппаратной транзакционной памятью и введения многоуровневого контроля невозможно было решить задачу обнаружения и идентификации разного типа угроз на всех уровнях иерархии выполнения запросов СК. В ходе экспериментальных исследований получены *новые*

научные результаты, подтверждающие эффективность и минимальную потерю производительности применения технологии аппаратной виртуализации в виде многоуровневой “песочницы” для перспективных СК по сравнению с использованием традиционных кластеров. Предложенные технологические решения в нескольких вариантах позволяют снизить затраты на производство за счёт переноса большей части испытаний с опытных образцов на программное обеспечение.

Внедрение научных результатов подтверждается наличием актов о внедрении (реализации) и структурными схемами (чертежами). Результаты теоретических и экспериментальных исследований, полученные в ходе выполнения диссертационной работы, были успешно внедрены автором при разработке программно-технического комплекса «Альфа-монитор» для российских суперкомпьютерных платформ «Ангара» серии ЕС1740, в рамках российского проекта по созданию высокопроизводительных мобильных (бортовых) вычислительных средств морского, наземного и воздушного базирования, при выполнении научно-исследовательской работы Минобрнауки РФ «Разработка методов построения защищенных облачных высокопроизводительных виртуальных вычислительных инфраструктур гетерогенной архитектуры для решения экстремальных задач в фундаментальных, научных, образовательных и производственных сферах» (регистрационный номер 88062011); авторские изобретательские и рационализаторские предложения реализованы в рамках китайского проекта по разработке защищенных суперкомпьютерных вычислительных комплексов серии Tian-he/СТ-2 (ОКР шифр «Удар грома»), в рамках японского проекта по созданию суперкомпьютера с высокими показателями производительности и защищенности серии Tsubame 3.0 (ОКР шифр «Стрела времени»), а также в учебном процессе и научных исследованиях на кафедре «Комплексная защита информации» ФГБОУ ВО «РГГУ» по дисциплинам «Программно-аппаратные средства защиты

информации. Основная часть», «Актуальные тенденции в области защиты информации», «Безопасность операционных систем».

Апробация результатов работы. Основные результаты работы были представлены и обсуждены на следующих конференциях и семинарах: Научно-практическая конференция MicroSoft «Современные подходы в построении защищенных систем», Москва, 2011 г.; Общероссийская научно-техническая конференция «Информационная безопасность регионов России», Санкт-Петербург, 2013 г.; 9-я Всероссийская конференция ITSF, Казань 2015 г.; научный семинар «Проблемы современных информационно-вычислительных систем», Москва, 2014 г.; Московский суперкомпьютерный форум, Москва, 2017 г.; Всероссийская научно-практическая конференция «Современные аспекты развития науки, образования и модернизации промышленности», Таганрог, 2016 г.; VII международная научно-практическая конференция «Управление информационной безопасностью в современном обществе», Москва, 2019 г.; Международная конференция «World Conference on Smart Trends in Systems, Security and Sustainability (WS4 2019)», Лондон, 2019 г.; Международная научно-практическая конференция «Информационная безопасность: вчера, сегодня, завтра», Москва, 2019 г.; ежегодный научный семинар в СПИИ РАН, Санкт-Петербург, 2019 г.; научный семинар в ФГБУ «3ЦНИИ» Минобороны России, Москва, 2020 г.; Международная научно-практическая конференция по компьютерной и информационной безопасности (INFSEC 2021), Екатеринбург, 2021 г.; Международная научно-практическая конференция ICICT 2022, London, 2023.

Публикации. Основные результаты по теме диссертации опубликованы в 31 научной печатной работе, из которых 10 – в ведущих рецензируемых научных изданиях, входящих в перечень ВАК РФ; 6 – в международных изданиях, индексируемых Scopus; 15 – в материалах конференций и других научных изданиях.

Также автором получено 2 патента РФ на изобретения и 1 свидетельство о государственной регистрации программы для ЭВМ.

Личный вклад автора. Все результаты, представленные в диссертационной работе, получены лично автором в процессе выполнения научно-исследовательских и опытно-конструкторских работ. На основе аксиоматической теории разработана и предложена новая парадигма написания правил политики безопасности, отличающаяся заданием новых сущностей вместо классической связки «субъект, объект, предикат» – «субъект», «объект» и «дескриптор оценки безопасности состояний». В работе решена важная научная проблема супервентности для отрасли промышленности, связанной с проектированием и внедрением средств защиты информации для суперкомпьютеров в стационарном и бортовом исполнении. Личный вклад автора в теорию (с точки зрения математических подходов и теории информационной безопасности):

1) Сформулировано условие разрешимости задачи идентификации на всех уровнях иерархии команд и доказана теорема.

2) Формализация каждой i -ю угрозы аргументов в виде конъюнкции предикатов из восьми логических переменных.

3) Спецификация на структурах Крипке с использованием темпоральной логики CTL.

4) Вычисление доверительной вероятности в качестве показателя защищенности компонентов среды.

Структура и объем диссертационного исследования. Диссертация состоит из введения, пяти глав, заключения и 4 приложений.

Основной материал изложен на 280 страницах. Полный объем диссертации составляет 309 страниц с 40 рисунками и 22 таблицами. Список литературы содержит 128 наименований.

Содержание работы

Во введении обоснована важность и актуальность темы диссертации, определены цель и задачи исследований, сформулированы научная новизна и практическая значимость.

В первой главе представлен обзор основных методов и проблем в области обеспечения информационной безопасности стационарных и бортовых СК, приведен анализ существующих технологий в области создания защищенных СК и архитектуры современных многопроцессорных систем с точки зрения информационной безопасности.

В результате проведенного анализа отмечено, что суперкомпьютерные вычислительные системы стали играть центральную роль в современной науке и технике, чего раньше никогда не было.

Основываясь на определениях функциональная безопасность (ГОСТ 61508, ГОСТ 62061), информационная безопасность (ГОСТ 27000), кибербезопасность (ГОСТ 62443), кибериммунитет (ГОСТ МЭК 62443 4-1:2018), информационная безопасность стационарных и бортовых суперкомпьютерных вычислительных систем дополнена новым понятием “дескриптор оценки безопасности состояний системы”.

Информационная система является кибериммунной (или обладает кибериммунитетом), если она разделена на изолированные домены безопасности, все взаимодействия между которыми независимо контролируются, и предоставляет: гарантии защищенности всей доверенной вычислительной базы, включая среду исполнения и средства контроля взаимодействий.

Кибериммунный подход основан на: разбиении системы на изолированные домены безопасности, многоуровневом контроле всех взаимодействий между доменами безопасности;обеспечении гарантированной защищенности вычислительной среды. Кибериммунитет построен на основе многоуровневой мультидоменной структуры, работает в защищенном адресном пространстве и позволяет описывать информационные и программные процессы до мельчайших подробностей. Любые действия, не разрешенные этой логикой, будут заблокированы автоматически. Дескриптор оценки безопасности состояний системы – это свертка состояний среды исполнения (аппаратной или виртуальной) и системного программного обеспечения» с учетом механизмов самоконфигурации и самоконтроля, которая «вычисляется» на основе атрибутов протекающих процессов, которые реализуются на уровне ядер микропроцессоров и среды межсетевого взаимодействия. У всех процессов вычисляются «свои» дескрипторы и задаются методы их обработки – get, set, delete и т.п. Определены задачи с учетом специфики суперкомпьютеров, предназначенные для решения ответственных задач и вопросов национальной безопасности, чувствительные к вычислительным ресурсам, и заданы метрики для набора вычислительных средств в стационарном и бортовом исполнении при работе на недоверенном оборудовании. В конце главы формулируется вывод о необходимости разработки новой модели обеспечения ИБ суперкомпьютеров с поддержкой виртуализации. Отмечено, что особую опасность представляют угрозы, связанные с неоднозначностью состояний выполнения прикладных и системных программ в недоверенной среде на стационарных и бортовых СК. На основе проведенного анализа и оценки влияния новых угроз на состояние защищённости ресурсов стационарных и бортовых суперкомпьютерных комплексов в главе определена цель диссертационного исследования и сформулирован перечень научно-технических задач, решение которых обеспечивает ее достижение.

Во второй главе введены новые теоретические и научно-методические принципы защиты суперкомпьютеров с использованием средств виртуализации, решена проблема супервентности. Доказана теорема о равномерной перечислимости функций СК. При этом стационарные и бортовые суперкомпьютеры рассматриваются как объект исследований впервые.

Сформулировано условие разрешимости, как следствие из теоремы о равномерной перечислимости функций, проблемы установки логического соответствия (супервентности) процесса выполнения программы на суперкомпьютере, изменением состояний компонентов гипервизора и аппаратуры: при числе уровней иерархии меньше 8, процессы на всех уровнях в СК не наблюдаемы, что затрудняет обнаружение и идентификацию угроз со стороны СЗИ.

Главный результат теоретических выкладок состоит в том, что мы доказали универсальность и равномерность для заданных классов операций функциональных преобразований. Независимо от архитектуры вычислительных узлов СК найдено такое число уровней иерархии ($N=8$), что в алгебраической системе существует инвариант. Если оценивать временные параметры, данное свойство регулярности сохраняется неопределенно долго (от 0 до $+\infty$). Хеш вычисляется рекурсивно для всех процессов (подпроцессов). В результате получается результативное значение для штатного измерения трассы прохождения запроса (на основе прошлого опыта, предобучение). В процессе работы идет обучение системы, каждое новая конфигурация классифицируется и в будущем распознается с учетом характерного набора маркеров.

Кодирование признаков идентификации – набор или конъюнкция простых предикатов. Классическая логика высказываний – это “черно-белая” модель, высказывания статичны, неизменны во времени.

В обычной логике высказываний адекватно не формализуются предложения, в которых явно или не явно присутствуют свойства, истинность которых изменяется со временем.

Предложенный подход к описанию операций основан на классификации рисков нарушения информационной безопасности и анализе контекста выполнения исходящих директив, посредством которых могут передаваться данные в обход требований принятой политики безопасности, что приводит к нарушению защищенности ресурсов вычислительных узлов.

Правила политики безопасности (ПБ) задаются в виде модальных конструкций в терминах темпоральной логики. Значение уровня привилегий представляет ряд десятичных чисел $\{0,1,2,3,4,5,6,7\}$. Все процессы распределены по доменам в соответствии с уровнем привилегий.

Если происходят изменения в структуре “выполнения процессов”, то изменение уровня привилегий индексируется на единицу, то есть можно мигрировать только между соседними доменами. Хеш вычисляется рекурсивно для всех процессов (подпроцессов) и получается результирующее значение для штатного измерения трассы прохождения запроса (на основе прошлого опыта, предобучение). В процессе работы идет обучение системы, каждая новая конфигурация классифицируется и в будущем распознается с учетом характерного набора маркеров.

Наборы конфигураций, адаптированных к разным контекстам выполнения операций, в виде матриц правил доступа сохраняются агентами гипервизора, ведущих мониторинг событий ИБ, в памяти контроллера транзакционной памяти. В процессе обучения обновляются и корректируются правила ПБ. Система динамически эволюционирует и модифицирует наборы модальных правил на основе грамматик темпоральной логики CTL по реагированию на сигнальные события.

Критерии идентификации угроз – в семантической трактовке набора предикатов. Принцип супервентности заключается в отсутствии различий одного вида при отсутствии различий другого вида: отсутствие различий в наборе дескрипторов процессов при отсутствии изменений в конфигурации программно-аппаратных агентов. Существует взаимно-однозначное логическое соответствие между изменением набора вектор-переменной, хранимой в генеративных таблицах гипервизора, картой состояний транзакционной памяти (уровень аппаратуры) и значением хеш-функции (уровень ПО).

В процессе прохождения запроса, ключевая строка не меняется в штатном режиме. Если меняется, то значит, осуществляется модификация маршрута прохождения запроса сторонним агентом.

В третьей главе с целью повышения уровня безопасности и защищенности суперкомпьютеров предложены и разработаны модель угроз целостности среды выполнения программ, модель безопасных операций, введена функция оценки безопасности состояний, осуществляющая контроль контекста выполняемых операций.

Впервые предложены и спроектированы подходы и принципы оценки безопасности стационарных и бортовых СК, построена функция оценки безопасности и реализована балльная оценка кодов безопасных состояний СК с учетом возможного ущерба от угроз. Для каждого заказчика можно создать готовый набор шаблонов виртуальной среды с набором нужного предустановленного ПО под контролем защищенного гипервизора. Вся работа пользователей, программистов и администраторов будет осуществляться в доверенной изолированной программной среде и будет работать на разных аппаратных платформах (с использованием трансляторов, эмуляторов, аппаратной виртуализации или контейнирования).

Экономятся расходы на дополнительных средствах создания доверенной среды и учитываем все требования регулятора еще на этапе разработки и внедрения системы, а не просто устанавливаем дополнительного аппаратных модулей и ПО “под ключ”. Работа технологически осуществляется в многоуровневой “песочнице” и можем создавать моделирующие расчетные стенды (тестовую среду и продуктивную зону) для промышленного внедрения всей линейки существующих и разрабатываемых процессорных вычислителей, что позволит бесперебойно поставлять и поддерживать различные кластерные вычислительные системы, мейнфреймы, обычные персональные компьютеры и бортовые супер-ЭВМ, АСУ ТП и суперкомпьютеры нового поколения для “обкатки” решений.

Предложено использовать модель угроз целостности среды исполнения процессов СК, в которой вместо классической связки «субъект, объект, предикат» реализована новая парадигма написания правил политики безопасности и заданы новые сущности – «субъект», «объект» и «дескриптор оценки безопасности состояний»: для каждой i -ой угрозы вычисляется значение функции оценки безопасности состояний, аргументы которой заданы в виде конъюнкции предикатов из восьми логических переменных, субъектами являются гостевые и управляющие ОС (создаваемые ими запросы), а объектами – компоненты гипервизора и контроллера транзакционной памяти.

Модель угроз учитывает новый класс уязвимостей, связанных с высокой асинхронностью процессов СК и возможностью выполнения ложных транзакций в результате дестабилизирующих внешних воздействий и случайных сбоев аппаратуры. Требования политики безопасности формулируются в терминах, которые задают последовательность обработки операций и ограничений на возможность повышения привилегий процессов.

При этом на каждом уровне модели операций S1-S8 ведется протоколирование событий и результатов, включая такие параметры, как Time – время и Result – результат выполнения операций.

Ограничение на повышение привилегий и контроль переходов при изменении контекста операций задаются значениями функции оценки безопасности состояний F_i , аргументы которой заданы в виде конъюнкции предикатов из логических переменных, а контроль выполнения процессов, порождаемых субъектами доступа, реализуется на основе принципа наименьших привилегий.

Для формализации описания рассматриваемых процессов автором предложена модель безопасных операций, учитывающая все режимы функционирования процессорных устройств.

Для учета всех режимов функционирования многоядерных процессорных устройств в разработанной модели операций введено четыре кольца защиты, а именно: Priv0 – уровень привилегий команд процессора (IPL_LEVEL), Priv1 – уровень привилегий ядра ОС (KERNEL_LEVEL), Priv2 – уровень привилегий администратора безопасности (SUPERVISOR_LEVEL), Priv3 – уровень привилегий пользователей (USER_LEVEL). При такой конфигурации доверительная вероятность равна приблизительно 0,9. Показатель защищенности системы достигает максимума. При этом состояния СК наблюдаемы: изменение контекста выполняемых операций детерминировано и может быть записано в виде «числа».

Гипервизор в связке с контроллером транзакционной памяти собой представляет многоагентную систему. Существуют проверки, соответствующие принципам работы классической логики, но есть алгоритмы с получением метаданных о работе компонентов системы на разных уровнях иерархии, где, наряду с проверкой целостности в виде изменения значений вычисленных хеш-функций для каждого процесса,

реализован контроль доступа в виде маркировки разрешенных переходных процессов при переключениях между доменами защиты и миграции потоков. Ведется история разных конфигураций системы, распознаются новые события за счет вариантного нового набора предикатов и значений их траекторий на структурах Крипке.

В работе применён формализм мультиграфа в виде модели выполнения операций СК, представляющего собой дерево. Стартовая точка – это вершина древовидной структуры, описывает контроллер транзакционной памяти (ТП), который напрямую взаимодействует с верификатором команд. Неоднозначность обусловлена тем, что выполнение программ на уровне объектов гостевых и управляющих ОС не эквивалентно изменению состояний компонентов гипервизора, обрабатывающих запросы пользователей на выделение вычислительных ресурсов при взаимодействии с физическим оборудованием. Каждый запрос представляет собой набор операций. Набор операций – это операции чтения, записи, выполнения, удаления и т.д. Каждая операция, в свою очередь, на каждом уровне иерархии выполнения транслируется гипервизором в набор директив, выполняемых процессором.

Механизм трансляции каждой операции представляет собой сборку команд непосредственного сценария выполнения исходящих директив с использованием тегированных полей с привязкой к конкретному уровню защиты с учетом уровня иерархии (привилегий) обрабатываемых запросов.

В четвертой главе разработаны методы гибридной защиты на основе применения маркеров и интервально-временных ограничений и реализован набор файлов конфигураций системного ПО. Разработан метод реактивной защиты, который включает в себя алгоритмы вычисления дескрипторов оценки безопасности состояний и идентификации угроз на основе маркировки состояний выполнения запросов СК, базирующихся на вычислении результирующей свертки на основе атрибутов протекающих

процессов, с заданной глубиной (масштабируемостью). Ошибки 1-го и 2-го рода могут возникать на локальном i -ом уровне, но они детектируются и исправляются на уровне системной задачи анализа контроля целостности поступающей информации с датчиков со всех уровней иерархии обработки запросов.

Контроллер в связке с верификатором реализует механизмы защиты: изолированное мультидоменное адресное пространство представляется в виде неперекрывающихся областей памяти, каждой из которых соответствует вершина графа, лежащая во втором уровне корневой структуры.

Поскольку у нас 8 уровней иерархии обработки запросов, то блок разбит на восемь подмножеств детализации уровней возможных состояний, обозначенных $S_8, \dots, S_1$. Последний уровень корневой структуры мультиграфа представлен агентами гипервизора (листья графа), через которые происходит связь с внешней средой и все возможные атаки на СК.

Продвижения на каждый следующий уровень – это более высокий уровень абстракции описания обработки запросов, переход от нижнего уровня спецификации к верхнему.

Каждое состояние может иметь не одну, а множество цепочек – продолжений, и является корнем своего дерева историй (вычислений).

Разработан метод реконфигурации среды выполнения программ проактивной защиты СК, основанный на методах темпоральной логики для проверки истинности принимаемых решений на каждой возможной интерпретации (структуре Крипке) с учетом специфики СК в виде временной сложности, связанной с параллельностью вычислений, и пространственной сложности, обусловленной большим количеством ядер процессоров и структурных связей между ними.

Базовый процесс инициализируется значением `context_id` (исходная вершина графа для старта вычислений). Нумеруются все процессы уникальным хеш-значением, который вычисляется для каждого вновь запускаемого приложения.

Изменение в поле `Ord` для дочернего процесса сигнализирует о появлении уже дочерних подпроцессов в рамках дочернего процесса (который породил еще вложенные подпроцессы) и так далее по уровню вложенности. Хеш вычисляется рекурсивно для всех процессов(подпроцессов) и получается результирующее значение для штатного измерения трассы прохождения запроса (на основе прошлого опыта, предобучение).

В процессе работы идет обучение системы, каждая новая конфигурация классифицируется и в будущем распознается с учетом характерного набора маркеров. Кодирование признаков идентификации – набор или конъюнкция простых предикатов, на которых задана формула F_i .

Тип контекста определяет режим выполнения операций чтения/записи: работа с сегментами памяти приложений, работа с системной областью памяти устройств, неинициализированные сегменты памяти для подготовки загрузки приложений, ошибки записи в буфер (неверные указатели и переменные окружения, переполнение, утечка памяти и т.д.). Чтобы достичь безопасности надо блокировать «опасности».

Блокировка требует синхронизации состояний большого количества высокопараллельных процессов, а для этого надо ввести еще один «слой» контроля в виде дополнительного домена защиты.

С этой целью введен дополнительный уровень защиты, который позволяет контролировать выполнение запросов на всех уровнях обработки без значительных потерь по производительности и энергоемкости.

Взаимодействие процессов пользователя происходит с компонентами гипервизора и контроллером транзакционной памяти, который создает изолированную среду исполнения. Компоненты гипервизора контролирует обращения к объектам файловой системы и стекам сетевых протоколов, осуществляют контроль последовательности выполнения запросов на выделение вычислительных ресурсов на всех уровнях иерархии (S1 – S8).

Каждый уровень иерархии привязан к определенному уровню защиты. Контроллер транзакционной памяти отслеживает и запрещает конкурирующим процессам обращаться к одному участку памяти, распределяет и привязывает обрабатываемые запросы к доменам в соответствии с их уровнем привилегий. Осуществляют также контроль выполнения запросов агенты гипервизора, функционирующие на всех уровнях иерархии и задающие модальные правила ПБ

Благодаря упрощенной модели описания возможно быстро перенести все требования по разработке автоматизированных систем в защищенном исполнении и их сертификации без дополнительной разработки новых стандартов, а перенос уже существующих подходов и критериев для традиционных кластеров и мейнфреймов с минимальными правками к автоматизированным системам “нового поколения”, с введением одного показателя – коэффициента оценки безопасной эксплуатации СК.

В пятой главе разработан методика тестирования уровня защищенности и реализован алгоритм идентификации угроз ИБ на основе таблиц правил ПБ и набора маркеров. Загрузка/выгрузка контрольных образцов вредоносного ПО производится в режиме эмуляции. Длина каждого цикла измерения равна восьми этапам обработки. Параметры выполнения трасс на основе счетчика команд вносятся в базу данных для дальнейшего обучения.

Обнаружение осуществляется на основе списка демаскирующих событий по значениям траекторий вычислений темпоральной логики CPL на структурах Крипке, по временным параметрам (изменение длины трассы выполнения инструкции) и величины энергозатрат на выполнение инструкций в пДж. Разработан многокомпонентный защищенный мультидоменный гипервизор, который состоит из монитора безопасности, проксирующих модулей и верификатора команд. Проведены тестовые верификационные испытания.

В Заключении сформулированы основные результаты и выводы по работе. Представленный научно-методический аппарат для идентификации новых классов угроз позволяет повысить эффективность выявления угроз в стационарных и бортовых суперкомпьютерных вычислительных, создать системы защиты информации с учетом специфики перспективных высокопроизводительных систем.

В диссертационном исследовании решена актуальная научная проблема супервентности для отрасли промышленности, связанной с проектированием и внедрением средств защиты информации для суперкомпьютеров в стационарном и бортовом исполнении.

ГЛАВА 1. АНАЛИЗ МЕТОДОВ И ТЕХНОЛОГИЙ ЗАЩИТЫ ИНФОРМАЦИИ СУПЕРКОМПЬЮТЕРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И ПОСТАНОВКА ЗАДАЧ ИССЛЕДОВАНИЙ

1.1 Анализ особенностей организации защиты информации в стационарных и бортовых суперкомпьютерах

Определим задачи с учетом специфики суперкомпьютеров для решения ответственных задач и вопросов национальной безопасности, чувствительные к вычислительным ресурсам, и зададим метрики для набора вычислительных средств в стационарном и бортовом исполнении. Зададим **три класса стационарных суперкомпьютеров:**

- 1 тип: малый кластер – Оперативная память: 250 – 500 Тбайт, Массовая глобально-адресуемая память: 100 Пбайт, бисекционная пропускная способность канала 50 – 100 Гб/с, “легкие” ядра;
- 2 тип: средний кластер – Оперативная память: 500 Тбайт – 1 ПБайт, Массовая глобально-адресуемая память: 100 Пбайт, бисекционная пропускная способность канала 100 – 250 Гб/с, “средние” ядра;
- 3 тип: стратегический класс – Оперативная память: 50 – 100 Пбайт, Массовая глобально-адресуемая память: 50 – 100 Пбайт, бисекционная пропускная способность канала 500 Гб/с – 1 Тб/с, “тяжелые” ядра.

Определим также **класс бортовых супер-ЭВМ** (4 тип) – Оперативная память: 100 - 200 Тбайт, Массовая глобально-адресуемая память: 1 – 10 Пбайт, бисекционная пропускная способность канала 10 – 50 Гб/с, “сверхлегкие” ядра.

Эксафлопсный суперкомпьютер – это суперкомпьютер с производительностью 1018 операций над 64-разрядными числами с плавающей точкой в секунду. В настоящее время имеются разные мнения по поводу путей создания таких суперкомпьютеров, среди которых достаточно популярно “лобовое” решение. Оно основано на использовании, наряду с коммерчески доступными суперскалярными микропроцессорами, огромного количества графических ускорителей, либо ускорителей типа Xeon Phi, либо ускорителей на программно реконфигурируемых кристаллах FPGA (ПЛИС).

Считается, что этот эволюционный подход позволит получить первые образцы экзафлопсных суперкомпьютеров, но они будут ограничены по возможностям и исключительно энергоемки, неудобны в использовании. Тем не менее, для узкого круга задач это расценивается на первых порах вполне допустимым. Экзамасштабные суперкомпьютерные системы – это суперкомпьютеры с планируемой удельной производительностью порядка Эксафлопс для выбранного класса инженерно-расчетных задач.

Принцип «невидимости» основан на том, что существуют недокументированные состояния в системе, которые никак не различимы для монитора безопасности и которые позволяют вредоносному коду маскироваться под штатный процесс.

Под вредоносностью понимается способность программ нанести ущерб вычислительной системе посредством блокирования, хищения, уничтожения и несанкционированной передачи информации [1; 2].

Для обычных кластеров данная технология защиты не может быть реализована из-за низкой производительности и нехватки вычислительных ресурсов [3].

Создать защищенную вычислительную систему с учетом масштабируемости предложенных решений, тесного взаимодействия с компонентами других информационных систем, затруднительно, поскольку защищенные ВС требуют специализированной аппаратной платформы, их функционал ограничивается решением определенного класса задач в сетях специального назначения ФСБ РФ и Министерства обороны РФ.

Кибербезопасность и информационная безопасность современных вычислительных систем могут описываться триадой составляющих ее сущностей определенных на составных частях киберпространства: информационных ресурсах, компьютерной и сетевой архитектурах (инфраструктуре) и способах взаимодействия пользователей. Указанные понятия охватывает уже не только информацию как объект защиты, не исключительно технические средства, которые определяют возможности функционирования информации, а защиту способов функционирования новой сущности – киберпространства. Защищается деятельность людей, которая осуществляется с помощью информации, распространяемой посредством технической инфраструктуры [4; 5] .

Информационная система является кибериммунной (или обладает кибериммунитетом), если она разделена на изолированные домены безопасности, все взаимодействия между которыми независимо контролируются, и предоставляет: гарантии защищенности всей доверенной вычислительной базы, включая среду исполнения и средства контроля взаимодействий.

Сегодня невозможно разрабатывать ПО, полностью защищенное от внешних воздействий, так как любой программный код содержит уязвимости, невыявленные на этапе разработки и тестирования, для верификации кода требуются значительные финансовые и научно – технологические затраты, что большинство организаций не может себе позволить в условиях коммерческой конкуренции, более того, нет

сто процентной гарантии, что в будущем не будут выявлены новые классы уязвимостей [6; 7; 8; 9; 10; 11]. Для коммерческих сетей данные технологические решения не являются эффективными, поскольку не учитывают поддержку оборудования различных сторонних производителей, неудобны для массовых пользователей с учетом эргономических показателей [12; 13; 14; 15].

Основываясь на определениях функциональная безопасность (ГОСТ 61508, ГОСТ 62061), информационная безопасность (ГОСТ 27000), кибербезопасность (ГОСТ 62443), кибериммунитет (ГОСТ МЭК 62443 4-1:2018), информационная безопасность стационарных и бортовых суперкомпьютерных вычислительных систем дополнена новым понятием – Дескриптор оценки безопасности состояний системы.

За счет использования резервов высокой производительности этих суперкомпьютеров можно создать гипервизор, обеспечивающий надежную защиту от атак, связанных с высокой асинхронностью процессов СК и возможностью выполнения ложных транзакций в результате дестабилизирующих внешних воздействий и случайных сбоев аппаратуры.

Классическое определение дескриптора безопасности следующее. Объекты, к которым могут получать доступ процессы, имеют специальный атрибут – дескриптор безопасности [16; 17; 18], содержащий информацию обо всех пользователях, которым разрешен или запрещен доступ к объекту.

Структура данных, представляющая дескриптор защиты и включает следующие основные поля [19]:

- Owner – SID владельца;
- Dacl – список управления избирательным доступом;
- Sacl – системный список управления доступом.

Списки управления доступом (ACL, Access-Control List) в системе представлены заголовком (ACL Header) и последовательностью элементов

списка (ACE, Access-Control Entry). Классические подходы в реализации контроля доступа предполагают использование атрибутов доступа на уровне управляющих и гостевых ОС. Объектами доступа выступают носители данных, представленные логическими структурами: директориями, файлами, сетевыми сокетами и областями памяти, которые участвуют в межпроцессном взаимодействии.

Мы предлагаем другой подход: вместо классического понятия «локализованная задача» для суперкомпьютеров следует говорить о параллельных распределенных потоковых структурах, генерируемых и обрабатываемых на разных уровнях иерархии конвейера команд процессорными устройствами, объединенных высокоскоростными сетями. Вместо классического понятия дескриптора безопасности как специального атрибута, состоящего из идентификатора владельца и списков доступа, в нашем случае применительно к суперкомпьютерам атрибуты доступа к объекту и привилегии субъекта, связи между ними формализуются в виде набора (конъюнкции) предикатов. Отслеживать взаимодействие и контролировать доступ можно по набору характерных признаков (маркеров), представленных в виде кортежей логических переменных. Для этого введем новую функцию оценки (измерения) безопасности состояний процессов СК и, соответственно, дадим новое определение.

Дескриптор оценки безопасности состояний системы – это свертка состояний среды исполнения (аппаратной или виртуальной) и системного программного обеспечения» с учетом механизмов самоконфигурации и самоконтроля, которая «вычисляется» на основе атрибутов протекающих процессов, которые реализуются на уровне ядер микропроцессоров и среды межсетевого взаимодействия. У всех процессов вычисляются «свои» дескрипторы и задаются методы их обработки – get, set, delete и т.п.

Кибериммунитет построен на основе многоуровневой мультидоменной структуры, работает в защищенном адресном пространстве и позволяет описывать информационные и программные процессы до мельчайших подробностей. Любые действия, не разрешенные этой логикой, будут заблокированы автоматически.

Развитие информационных и коммуникационных технологий вызвало возникновение ряда новых и развитие некоторых существующих угроз информационной безопасности, таких как [20; 21; 36]: деструктивные информационно-технические воздействия (в том числе применение кибероружия, средств радиоэлектронной борьбы (РЭБ), проникновение в компьютерные сети) на информационно-технические объекты АС; компьютерные атаки на информационные сегменты АС (информационно-коммуникационные, функциональные, информационно-психологические и др.); преднамеренные действия, а также ошибки персонала и диверсионно-подрывная деятельность специальных служб иностранных государств; электромагнитный терроризм; внедрения в аппаратные и программные изделия АС, реализующих функции, не предусмотренные документацией на эти изделия (НДВ); использование базы данных скомпрометированных идентификаторов; доступ к ресурсам неавторизованных пользователей по действующим аппаратным идентификаторам (смарт-карты, токены и т.п.); подмена «облачной» инфраструктуры обработки информации; активация служебных режимов функционирования изделия путем получения специальных команд (поток) при штатной обработке информации; обеспечение функционирования СВТ под управлением недоверенного гипервизора и т.д.

Главные угрозы российским суперкомпьютерам, по нашему мнению, происходят от используемой импортной элементной базы, импортных серверных плат, а также программного обеспечения, особенно средств виртуализации и реализации облачных вычислений:

- Как отмечалось, известно о работах по встраиванию закладок в GPU в Китае и Тайване (TSMC), а также США. Наличие таких закладок по поведенческой информации о работе GPU тяжело обнаружить из-за его огромной производительности. При программировании таких закладок стали использовать развивавшиеся в 80-х годах средства преобразования символьной информации. В частности, есть информация по изучению в этом плане в Китае русского языка Рефал и американского языка Ambit в рамках проекта “Аист”.
- Известно об уже обнаруженных закладках на уровне BIOS и BMC в импортных серверных платах, в частности, китайского производства.
- Средства виртуализации и облачных вычислений позволяют на “законном” основании контролировать вычислительный процесс и “легализуют” некоторую потерю производительности, которая, на самом деле, может происходить из-за работы вредоносных программ [21; 23].

Наиболее страшной из перечисленных угроз является применение в суперкомпьютерах импортной элементной базы, в частности, ускорителей GPU, а в будущем – MIC. От упомянутых внутренних угроз суперкомпьютеры не спасут ни изоляция от внешнего мира в бункере, ни изоляция от глобальных сетей, поскольку опасность таится внутри них самих. В качестве частичного решения необходима, по крайней мере, многоуровневая защита суперкомпьютеров, включающая и многоуровневый контроль их функционирования. Анализ огромных объемов информации, поступающих от такой системы контроля, требует использования особых аппаратно-программных технологий, включая даже специализированные суперкомпьютеры. Здесь исключительно интересны и технологии мобильных агентов. Для таких задач разрабатываются специальные

массово-мультиплатформенные суперкомпьютеры, в которых известная проблема больших задержек выполнения обращений к памяти (“проблема стены памяти”) решается за счет выведения работы приложений и оборудования в такой режим, когда можно использовать высокую пропускную способность памяти, создавая при этом “иллюзию” выполнения обращений с малыми задержками [24].

В США и Китае такие суперкомпьютеры и специальная элементная база для них разрабатываются. В России такие разработки не ведутся, но надеемся, что они будут восстановлены, причем с учетом новых современных требований. Возможны и более доступные пути создания близких к таким специальным суперкомпьютерам кластерных суперкомпьютеров на коммерчески доступной элементной базе и, возможно, с применением фрагментов на ПЛИС. Такие подходы в настоящее время применяются и в США, что объясняется экономическими соображениями – заказные массово-мультиплатформенные суперкомпьютеры пока дороги и предназначаются лишь для стратегических приложений.

Из самоконфигурации и самоконтроля следует интеллектуальность систем защиты информации (СЗИ). Предлагаемые методы гибридной защиты согласно следствию из теоремы Геделя, должен включать в себя алгоритмы вычисления дескрипторов оценки безопасности состояний, а также алгоритмы идентификации угроз на основе маркировки состояний выполнения запросов СК.

Картина состояние в области суперкомпьютеров и кибербезопасности в России такова, что желает во многом лучшего. Слишком много в последние 10-15 лет в этих областях было отдано коммерческим структурам, причем как отечественным, так и иностранным, а также единичным промышленным монополистам.

Явно пора расширить количество реальных участников работ в этих областях, а именно – активнее подключать военных и отечественное разведсообщество, отечественные технические университеты [25].

Поскольку главная угроза российским суперкомпьютерам исходит от используемой импортной элементной базы в виде высокоскоростных микропроцессоров и особенно ускорителей, то важно изучать варианты аппаратных закладок, способы их обнаружения и нейтрализации.

Возобновить в стране собственные исследования и разработки по элементной базе и архитектуре суперкомпьютеров, устойчивой к атакам, сбоям и отказам. Одним из самых действенных методов обнаружения атак является анализ информации о функционировании системы и активности ее пользователей, что является задачей класса интенсивной работы с данными, то представляется важным разработка ориентированных на эффективное решение этих задач суперкомпьютеров как кластерного, так и заказного типа (например, с применением массово-мультитредовых архитектур и глобально адресуемой памяти), а также ориентированных на их применение алгоритмов [26; 27; 28].

Мы уверены, кроме организации соответствующих работ, крайне важно организовать регулярную информационно-аналитическую работу по тематике перспективных суперкомпьютеров вычислительного и невычислительного типа, их программного обеспечения и аппаратных вариантов исполнений [29; 30; 31; 32]. Необходима консолидация научного сообщества, поддержка в рамках этого перспективных работ и специалистов.

1.2 Обзор инновационных проектов в области создания суперкомпьютерных платформ

Для понимания предметной области исследования приведем сначала общие сведения о структуре и функционированию мировой сети Интернет. Общая структура мировой сети Интернет представлена на рисунке 1.1.

Комментарии к элементам этой сети даются далее, здесь приведем лишь сведения о взаимодействиях в этой сети, для чего поясним функциональное использование линий связи на рисунке 1.1, помеченных цифрами:

1 – подключение к закрытой военной сети США SIPR (Secret Internet Protocol), которая объединяет только военные организации и спецслужбы, а также высшее политическое руководство США. В сети SIPR не используются открытые коммерческие сетевые протоколы (IPv4 и IPv6), вместо них применяется специальный протокол, разработанный АНБ (NSA) по заказу Министерства обороны США. Одновременно с этим применяются только сертифицированные аппаратно-программные средства. Естественно, что вход в сеть SIPR из мировой сети Интернет невозможен;

2 – условно показанный выделенный канал администрирования, применяется при регистрации нового национального провайдера для получения через NCIF (National Computer Information Fond) уникального Интернет-идентификатора домена и настройки с помощью NCIF конфигурации серверов провайдера этого домена в World Wide Web (WWW);

3 – выход в WWW с уровня национальных доменов. Заметим, что информация по линии 3 может сканироваться со стороны NCIF и агентов сети 4-го уровня (транснациональные корпорации и спецслужбы США), вплоть до их блокировки (см. также пояснения к линии 8);

4 – подключение Интернет-серверов организаций к федеральным и региональным провайдерам национальных доменов;

5 – вариант прямого выхода с нарушением российского законодательства (например, из иностранного торгового или дипломатического представительства X) минуя национального провайдера страны пребывания и в обход контролирующих центров (например, МВД, ФСБ и Минцифры (Минкомсвязи));

6 – подключение пользователей организации к ее Интернет-серверам;

7 – прямое подключение пользователей к провайдеру по выделенному каналу связи;

8 – условно показанная линия сканирования, анализа и управления WWW со стороны 4-го уровня сети. На самом деле, сканирование и управление возможно на всех уровнях посредством многоагентных систем использования аппаратных закладок и недекларируемых возможностей.

Уполномоченная организация по распределению нумерации сети Интернет - Internet Assigned Numbers Authority (IANA) – является структурным подразделением ICANN и осуществляет повседневное администрирование Системы доменных имен (DNS) на глобальном уровне.

Не трудно видеть, что по определению, работа в глобальной сети Internet находится под американским контролем, поэтому путь к достижению информационной безопасности – работа в выделенных сетях.

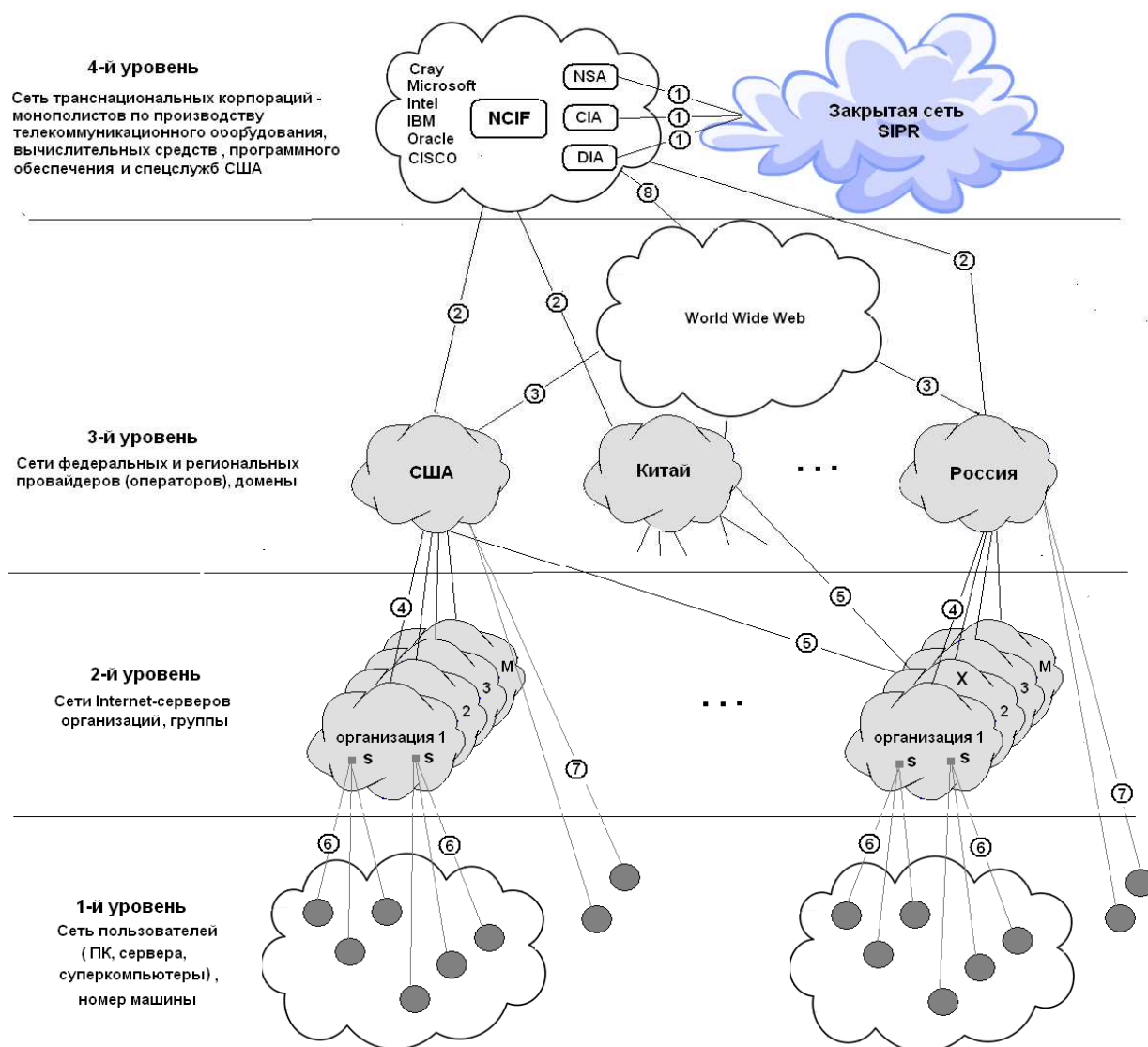


Рисунок 1.1 – Структура глобальной сети Internet

Высокопроизводительные технологии станут основой для формирования нового пространства, создаваемого в рамках инициативы, получившей название Intelligence Community Information Technology Enterprise (ICITE) [33].

Например, Национальный провайдер федерального уровня CNGI-CERNET2 состоит из:

- **Центра обнаружения:** Центр обнаружения в данном решении играет роль "антенны". Он обеспечивает получение протоколов обнаружения от центра управления, выявление и обнаружение DDoS-

трафика, а также отправку результатов обнаружения обратно в центр управления. В настоящее время, в качестве центра обнаружения используются модули обнаружения USG9310/9320 или USG5300ADI от компании Huawei Symantec.

● **Центра обслуживания и управления:** Центр управления, являясь мозгом данного решения, обеспечивает пользователей доступом в Интернет 2 и управляет каналами связи (скорость, пропускная способность и т.п.), ведет мониторинг использования ресурсов телекоммуникационного оборудования.

В Национальном центре развернуто 20 виртуальных частных сетей (*сеть объединяет 20 городов, в каждом городе по одной региональной виртуальной сети*) на базе программно-аппаратного комплекса Huawei Symantec SVN3000, являющийся шлюзом доступа безопасности виртуальной частной сети (VPN) протокола безопасных соединений (SSL)/ протокола безопасности IP (IPSec) с большой емкостью и высокой надежностью. SVN3000 базируется на защищенной аппаратной платформе и распределенной операционной системе реального времени Huawei Symantec.

Объединяя их преимущества и возможности, Федеральный провайдер Китая обеспечивает крос-платформенное решение безопасности доступа для организации сети IP VNP. Он также поддерживает точный контроль авторизованного доступа к ресурсам внутренней сети. Он предоставляет как международные алгоритмы шифрования и дешифрования, так и национальный стандарт шифрования (протокол Dragon-Xing «部分庚子», разработанный проф. NUDT Китая Денлу).

Более того, в Китае создана мощная государственная система ведения информационного противоборства, которая позволяет осуществлять массированное применение сил и средств в краткие сроки.

В рамках реформирования вооруженных сил страны, проводимого Министерством обороны КНР, NUDT создает Научный центр по вопросам ведения кибервойн. Он будет располагаться в городе Чжэнчжоу (административный центр провинции Хэнань).

В него войдут несколько управлений, основная задача которых – формирование теоретической базы, а также разработка основных принципов и методов ведения информационной войны для дальнейшей их реализации.

Этот центр будет оснащен всеми новейшими видами техники. Здесь будет действовать специальная группа по использованию и совершенствованию методов скрытого перехвата информации с помощью программно-аппаратных закладок (*Blue Team – «голубая команда Жемчужина Востока»*).

Ее возможности позволят эффективно осуществлять радиоперехват, подавлять системы ПВО и ПРО, преодолевать компьютерную защиту противника. Реализация данного проекта осуществляется под непосредственным руководством генштаба НОАК.

В Нанкине (провинции Дзян-Су) располагается вычислительный центр CADT (China Academy of Defence Technologies – 7 бюро Министерства Национальной Безопасности), в котором готовят офицеров радиоэлектронной борьбы командного профиля среднего и начального уровня и инженерного профиля среднего и высшего уровня в интересах ВС КНР. Также осуществляются научно-исследовательские работы по «созданию инженерных систем, снижающих степень поражения и защищающих от средств поражения», «теория и проектирование вооружения», «мосты и тоннели», «артиллерия, самоходное вооружения и боеприпасы». Более того, здесь действует Китайское сообщество так называемых «красных хакеров – братья Мао».

Фактически речь идет о формировании подразделений «военных хакеров», которые впервые появились в 2000 году. Данный центр имеет доступ к пяти национальным суперкомпьютерным центрам, вычислительным узлам China-GRID, транснациональным телекоммуникационным сетям.

Объединение разнородных вычислительных систем обуславливает реализацию расширенных механизмов защиты персональных данных пользователей, использование надежных алгоритмов шифрования, программно-аппаратных средств контроля целостности, создание доверенной среды загрузки. В суперкомпьютерных системах много пользователей, запускающих разные дистрибутивы ОС, поэтому необходимо разграничивать каналы обмена, физические и виртуальные ресурсы в соответствии с ролевыми политиками безопасности, чтобы исключить повреждение и хищение защищаемой информации с разными уровнями доступа, несанкционированный доступ (НСД) со стороны злоумышленника [34].

Создание сети для распределенной среды получило кодовое название «Huddle». Предусматривается разработка при участии компаний Mellanox, Trinity-Networks, Luxent Group нового маршрутизатора InfiniBand Cloud VPI – switch (*с встроенной поддержкой аппаратной виртуализации*), скорость передачи в котором будет 100 Гб/с. Также телекоммуникационное оборудование будет поддерживать полную аппаратную виртуализация 10 – Гигабитного Ethernet PCI – Express.

Создаваемая трехуровневая глобальная система (SpiderNetX), будет из следующих базовых звеньев:

1. Вычислительные узлы суперпроизводительных компьютерных центров на базе ведущих научно-исследовательских лабораторий (Массачусетский институт, Университет Джорджии, Тихоокеанская исследовательская лаборатория, Ливерморская национальная

исследовательская лаборатория, Лаборатория специальных исследований Trinix Агентства национальной безопасности США, компания IBM, компании Cray Research Inc и SGI, вычислительный центр S.C.O.M.P Пентагона, Компьютерный суперкластерный центр N – Cube Института стратегических исследований ЦРУ).

На вычислительных узлах высокоскоростных коммуникационных сетей суперкомпьютеров стратегического назначения устанавливаются защищенные гибридные ОС, отличные от классических операционных систем класса Unix или Windows NT, с расширенной аппаратной изоляцией выполнения привилегированного кода, с поддержкой непрерывной глобальной адресации физической памяти порядка 256 Терабайт и мультитредовой аппаратной архитектуры: ОС Cray MTK для сегментов сети Пентагона, ОС Blacker для сегментов сети АНБ, ОС XTS для сегментов сети ЦРУ.

2. Сервисные узлы с поддержкой мейнфреймов и стандартных кластерных систем под управлением классических Unix-подобных ОС (Red Hat Linux, SuSe Linux, Mandriva Linux, ALT Linux, IBM AIX, SGI IRIX, SCO, Sun Solaris). Они предназначены для хранения и обработки информации на файловых серверах, обеспечением доступа к открытым публикациям научно – исследовательских центров, системам картотек Библиотечных фондов, для обработки поисковых запросов с клиентских ЭВМ в социальных сетях.

Сервисные узлы глобальной системы SpiderNetX включены в контур информационных систем образовательных учреждений США, Канады, Австралийской ассоциации, Европейского Союза (Великобритания, Франция, Германия, Италия, Швейцария), стран Ближнего Востока (Саудовская Аравия, ОАЭ), Центральной и Восточной Азии (Индия, Китай, Япония, Южная Корея, Сингапур), стран Прибалтики (Литва, Латвия, Эстония), России и стран СНГ (Казахстан, Азербайджан, Украина).

Основные инновационные академические исследования по замыслу правительства США будут проводиться на территории США, Канады, Европы и Австралийской ассоциации. С целью снижения финансовых затрат на инновационные проекты «новой эры суперкомпьютеров» в качестве технологической платформы и сырьевой базы выступают выгодные экономические зоны Центральной и Восточной Азии: Исследовательский центр IBM в Индии, Шанхайская лаборатория Cray Inc в Китае, Объединенный центр компьютерных исследований SGI Baker Сингапура, Центр стратегических исследований Минатома Японии, Межведомственный вычислительный центр Университета Сеула (Южная Корея).

3. Клиентские ЭВМ, рабочие станции пользователей (операторов) глобальной распределенной информационной системой, подключенные к локальным сетям образовательных учреждений и имеющие доступ в Internet, мобильные «тонкие» клиенты и персональные компьютеры. На клиентских вычислительных средствах в странах Ближнего Востока, Центральной и Восточной Азии, Европы, России и странах СНГ в основном установлены ОС класса Windows NT. Более того, использование программного обеспечения компании Microsoft Пентагон США утвердил в рамках проекта JCOINT (Joint Command Operation Intelligence): создания объединенной АС административного сопровождения тактических операций НАТО, в которых пользователи которых не обладают достаточно высокой квалификацией для администрирования систем класса Unix и привыкли работать в среде Windows, реализация подобного проекта взаимодействия между распределенными узлами единой системы с использованием программно-технических решений для операционных систем класса Unix более дорогостоящая задача по сравнению с коммерческими решениями от компании Microsoft.

К социальным сетям образовательного назначения не предъявляются требования гарантированной защищенности, поскольку в контуре общедоступных сетей нет информации, составляющей государственную тайну. Однако социальные сети предоставляют широкие возможности для проведения экономической или научно-технической разведки: «распределенные программные агенты» в составе ПО могут осуществлять контроль за действиями пользователей с целью коммерческого шпионажа или шантажа, НСД к информационным ресурсам в смежных сегментах защищенных коммуникационных сетей (модификация доменной структуры обработки запросов, «подделка» пакетов авторизации, изменение таблицы маршрутизации).

Вредоносное ПО также реализует функциональные возможности, позволяющие отслеживать трансфертные сделки банковских организаций (номера счетов, данные юридических и физических лиц, «теневые» финансовые операции за счет изменения путь передачи транзакций в автоматизированных системах электронных платежей).

ФБР и ЦРУ спонсируют технический проект «Covert Channels Pattern»: создание и поддержка скрытых каналов доступа в 3G-мобильных сетях, в глобальном пространстве Internet, системах телевидения, в смежных сегментах распределенных сетей Internet – провайдеров. Однако подобные технические решения основаны на создании доверенной программно-аппаратной среды функционирования вычислительных средств на всех этапах ее «жизненного» цикла.

Указанный опыт разработки специальных автоматизированных систем не позволяет в полной мере «по методу аналогии» апробировать решения в глобальном пространстве социальных (коммерческих) сетей [35; 36].

Суть подхода в области создания защищенных систем сводится к созданию изолированной информационной среды, реализации политики многоуровневого разделения доступа, расширенных механизмов аутентификации-идентификации (например, технологическое решение «Аккорд»).

В суперкомпьютерных системах главным условием выступает полнофункциональная поддержка ПО сторонних производителей и возможность быстрой (гибкой) настройки отдельных компонентов. Информационные системы данного класса в отличие от коммуникационных сетей общего и специального назначения не могут «существовать» в изоляции со стороны аппаратно-программных средств контроля, обеспечения достоверности и защиты ПО.

Следует отметить важную роль сетей в условиях глобализации процессов обучений, развития Internet, технологий проведения видеоконференций в режиме реального времени, развития партнерских научно-исследовательских программ в рамках проектов ЮНЕСКО и ООН, сотрудничества коммерческих и государственных учреждений образовательного назначения.

Именно поэтому необходимо найти оптимальное сочетание «защищенность-полнофункциональность» для клиентских ЭВМ под управлением ОС класса Windows. Более того, встраивание дополнительных компонентов контроля и защиты ПО не должно сказываться на быстродействии (эффективности) обработки информационных потоков в целом.

Подобное программно-техническое решение позволит, с одной стороны, повысить уровень защищенности суперкомпьютеров, с другой стороны – реализовать различные сценарии экономической или научно-технической разведки, используя ресурсы компьютерных систем стратегического назначения и коммерческих сетей.

Установка дополнительных аппаратных модулей на системной плате и процедура настройки специального ПО системам защиты и контроля требуют высокой квалификации администратора безопасности и операторов системы, кроме того, изменение конфигурации физических устройств осуществляется сервисным инженером от компании, ответственной за поддержку и гарантийное обслуживание средств вычислительной техники.

Если на серверах традиционно устанавливается защищенная Unix-подобная ОС, то на рабочих станциях операторов, пользователей загружается образ ОС класса Windows NT, так как ПО других производителей требует высокой квалификации администратора и специалистов сервисного обслуживания (например, решения Oracle, Sun, HP и Cray Inc), более того, большинство программ разработки экспертных систем, офисные приложения, средства автоматизированного проектирования (САПР) поддерживают платформу Windows NT (например, базовые средства САПР от концерна «Ангстрем» в г. Зеленоград КОМПАС, продукция компании «1С», системы картотек класса «АЛЕФ»).

Среди российских национальных суперкомпьютерных платформ в следует отметить суперкомпьютер «Ломоносов-2», а также «Christofari Neo». Следует упомянуть про суперкомпьютеры компании «Яндекс» «Ляпунов», «Червоненкис» и «Галушкин». Однако для всех указанных суперкомпьютерных национальных платформ системно не реализованы подсистемы обеспечения информационной безопасности. Внедрены только отказоустойчивые кластеры на базе систем хранения данных и резервирования каналов связи и данных.

При этом нельзя говорить о полноценной государственной национальной суперкомпьютерной инфраструктуре, потому что большинство компонентов (элементная база) производятся иностранными фирмами.

1.3 Проблемы и недостатки применения традиционных методов и технологий защиты информации в вычислительных комплексах, использующих суперкомпьютерные технологии

Диссертационное исследование связано с раскрытием двух больших тем – стационарные и бортовые суперкомпьютерные вычислительные системы, особенности применения суперкомпьютеров в обеспечении кибербезопасности.

В этой связи надо отметить, что это делалось лишь для того, чтобы показать место задач, которые возлагаются на суперкомпьютеры, а самое главное – какие типы суперкомпьютеров будут решать эти задачи.

Теперь перейдем к общей характеристике предлагаемого материала. Во-первых, ставился вопрос о типах применяемых суперкомпьютеров, сетях вычислительных центров и критических инфраструктурных объектах, защита которых требует применения СК.

Во-вторых, ставился вопрос о новых задачах обеспечения кибербезопасности, если на защищаемой и атакующей стороне появятся суперкомпьютеры.

В целом, в данной работе дадим ответ на эти вопросы таков: особенности угроз и обеспечения кибербезопасности, обусловленные появлением именно суперкомпьютеров.

Скрытые, до этого неизвестные угрозы представляют собой серьезную опасность в связи с разработкой вычислительных систем пета и эксафлопсного уровня производительности. Суть их в следующем:

- применяется высокопроизводительная элементная база суперкомпьютеров, имеется огромное количество компонентов, что повышает вероятность закладок и затрудняет борьбу с ними;

- суперкомпьютеры обладают высочайшей производительностью, большим количеством одновременно выполняемых процессов (сотни тысяч – до миллиона), а это определяет трудность отслеживания происходящих в них событий с целью обнаружения вторжений;
- обрабатываемая с использованием суперкомпьютеров информация, обычно связана с решением вопросов национальной безопасности и решением важнейших научно-технических задач, осуществлением управления критически важными инфраструктурами и т.д., это серьезная мотивация для организации атак;
- пользователи суперкомпьютеров имеют высокую квалификацию и их обычно много, что повышает риски внутренних атак;
- высокой квалификацией обладают заинтересованные в атаках на суперкомпьютерные ресурсы лиц, обычно представляющих организованные сообщества, поддерживаемые на государственном уровне;
- атакующий суперкомпьютер может быстро оценить состояние атакуемого объекта, найти места уязвимости, спланировать и провести массовую атаку, быстро адаптироваться по результатам ее проведения и провести серию последующих атак.

Какие задачи обеспечения информационной безопасности появляются в связи с появлением суперкомпьютеров? Выделим следующие:

1. Важно понять, как повлияет применение суперкластерных платформ на создание территориально распределенной сетевой вычислительной системы с точки зрения защиты от НСД;

2. Для функционирования суперкластерных центров необходимо подводить мегаватты энергии. Первоочередной задачей становится обеспечение механизмов быстрого резервирования, «зеркалирования» критически важных процессов и физической защиты цепей питания. В противном случае, умышленная атака или непреднамеренный сбой в контуре силового питания приведут к потере огромных массивов важных данных, необходимых для решений инженерно-расчетных и аналитических задач;

3. Значительно возрастают требования к квалификации пользователей и администраторов безопасности. С одной стороны, объясняется это тем, что суперкомпьютер представляет собой сложную распределенную вычислительную систему с поддержкой глобально адресуемой памяти, оперирующей терабайтами данных, и на которой установлено специализированное ПО, для работы с которым нужно дополнительное обучение. С другой стороны, для изучения гарантированной защищенности суперкластерных центров, зондирования суперкомпьютерных сетей необходимо привлечения специалистов-хакеров, обладающих знаниями более искусных взломов автоматизированных систем, чем требовалось раньше для обычных промышленных сетей;

4. Суперкомпьютерные центры становятся «ближе к правительству», поскольку они являются стратегически важным звеном в работе национальных вычислительных комплексов. В информационной среде суперкластеров циркулируют категоризованные данные с разными грифами секретности, что требует усовершенствования политики безопасности при работе с разными сегментами сети, не допуская несанкционированного

доступа пользователей с меньшими правами доступа к данным операторов с более высокими «метками доступа» к категоризированным документам;

5. С появлением графических карт терафлопсного уровня производительности (например, NVIDIA Tesla), встроенных сетевых адаптеров с поддержкой сверхбыстрой передачи сетевых пакетов - 10/100/1000 Гигабит/с появляется возможность скрытого перехвата информации, ее расшифрования и передача злоумышленнику, а также организация управляемых атак с использованием программно-аппаратных закладок, установленных в графических картах и материнских платах;

6. Остро встает вопрос о реализации самими разработчиками ПО дополнительным механизмам обеспечения безопасности, поскольку суперкластерный центр представляет собой систему, состоящую из большого количества вычислительных и сервисных узлов, сетей разной топологии, использующие как стандартные PCI Express и Infiniband, так и специализированные заказные сети.

Использование традиционных подходов не позволяет решить проблему повышения уровня защищенности высокопроизводительных вычислительных комплексов с учетом гибкости, масштабируемости (поддержки аппаратных платформ разного класса) предлагаемых программно-технических решений и минимизации расходов.

Анализаторы сетевых вторжений создаются, чтобы обеспечить дополнительный уровень защиты вычислительной сети, дополняя традиционные средства защиты: межсетевые экраны, криптомаршрутизаторы, серверы аутентификации. Злоумышленник во время осуществления атак пытается преодолеть систему защиты, используя как методы «грубого взлома», так и новые «руткит»-технологии для скрытого доступа к ресурсам операционных систем и наблюдения за информационными потоками.

Система анализа защищенности призваны осуществлять сбор данных об «аномальной» активности в системе и выявлять факты скрытого воздействия на ресурсы информационных систем, предупреждать попытки осуществления несанкционированного доступа.

Несмотря на то, что эти системы не могут обнаруживать атаку в процессе ее развития, они могут определить возможность реализации атак. Обнаружение атак осуществляется посредством анализа регистрационных журналов, баз данных аудита или информационных потоков в режиме реального времени.

Средства обнаружения атак позволяют усовершенствовать классические подходы в области создания СЗИ посредством функциональной реализации механизмов сигнатурного поиска, повышающих уровень защищенности вычислительной сети [37].

Например, средства обнаружения компьютерных атак распознают попытки осуществления НСД нарушителем, осуществляют анализ параметров передаваемых пакетов в сети и сигнатурный анализ на наличие «характерных отпечатков пальцев» (fingerprints) в тактических действиях противника. При этом указанные технические средства расширяют функциональность межсетевых экранов, осуществляя контроль доступ пользователей к ресурсам внутренних и внешних сегментов локальных и глобальных телекоммуникационных систем [38; 39; 40; 41; 42].

Данный подход позволяет контролировать целостность объектов файловой системы, осуществлять контроль доступа к ресурсам сетевого адаптера, регистрировать попытки осуществления НСД посредством изменения атрибутивных признаков ресурсов вычислительных систем.

Модель адаптивных систем защиты распределенных систем позволяет осуществлять гибкую настройку компонентов СЗИ и оперативно реагировать на внешние дестабилизирующие воздействия со стороны нарушителя в условиях, когда обстановка информационного противодействия крайне изменчива.

При этом можно привести аналогии из биологического мира. Любая форма жизни обладает врожденными или приобретенными механизмами адаптации к внешней среде обитания. В случае изменения устойчивых показателей «экониши» существования, организм стремится в наиболее короткие сроки оптимально подстроится под новые условия среды. Так и адаптивные системы защиты осуществляют изменение конфигурации СЗИ, настройку ее отдельных компонентов с учетом сигнальных событий в ответ на действия противника, повышая уровень защищенности в критические периоды осуществления активного вторжения в компьютерные системы [43; 44].

Основные положения классических моделей и технологий безопасности компьютерных систем (дискреционного, мандатного, ролевого управления доступом, безопасности информационных потоков и изолированной программной среды) дополнены ориентированными на применение моделями безопасности логического управления доступом и информационными потоками (ДП-моделями), в том числе иерархическим представлением мандатной сущностно-ролевой ДП-модели, на основе которой реализовано управление доступом в сертифицированных защищённых ОС общего и специального назначения.

Данные технологии и модели основаны на динамическом анализе действий пользователей, контроля доступа к ресурсам ВС посредством детального анализа информации журналов аудита и «меток» доступа монитора безопасности, выявлении неявных (скрытых) взаимодействия пользователей и объектов доступа.

Они не позволяют в полной мере решать проблему анализа защищенности информационных систем с учетом применения противников программных средств скрытого воздействия, алгоритмов и методов, реализующих принцип *«невидимости»*, поскольку учитывает лишь ограничение действий субъектов доступа и воздействий вредоносного кода с применением штатных средств защиты.

Также не учитывается возможность осуществления распределенных атак злоумышленников с использованием программных агентов [45].

Программный агент злоумышленника может использовать процессор до работы легальных программ, то есть агент модифицирует системные данные так, как нужно ему, затем обработать запрос и передать легальным программам ту информацию, которую ему нужно.

Применение средств, реализующих камуфляж и дезинформацию, препятствует успешному осуществлению несанкционированного доступа злоумышленником, поскольку нарушитель в случае использования обманной системы защиты, вынужден затрачивать на альтернативные пути осуществления вторжения, поскольку у него нет достоверной информации о том, работает ли он с реальной операционной системой или является участником обманной игры [46].

Данное средство разработано с той целью, чтобы ввести в заблуждение автоматизированные средства анализа защищенности путем создания ложных уязвимостей, что позволит своевременно обнаружить попытки несанкционированного доступа и предпринять эффективные меры по их противодействию.

Применение технологий обманных систем защиты позволяет осуществлять скрытый мониторинг действий злоумышленника, изучать его средства, уровень квалификации и тактические характеристики.

Полученная информация может служить исходными данными для аналитиков службы безопасности и формирования более целостной картины угроз информационной безопасности компании или проведения оперативных мероприятий по защите важных данных. Однако авторы [47; 48; 49; 50] указывают на существенный недостаток применения «альтернативных» технологий обманных систем защиты (ОБС): применение злоумышленником программных средств скрытого воздействия и перехват информационных потоков вычислительных систем в отладочной сессии (механизм обработки SEH) позволяет осуществлять НСД в обход «муляжа», так как доступ к объектам ОС на уровне системного обработчика исключений и ошибок предоставляет злоумышленнику полный доступ к данным на более низком интерфейсном уровне. Важным направлением совершенствования методов защиты информации является противодействие билатеральным угрозам информационного взаимодействия, которые требуют учета направления, синтаксиса и контекста передачи данных, при работе с памятью и сетевыми протоколами, что позволяет блокировать широкий спектр угроз, способных снизить состояние безопасности исполняемой среды [51; 52; 53; 54; 55; 56; 57; 58; 59; 60].

Современные модели угроз для виртуальных сред суперкомпьютерных платформ (облачных вычислений и т.п.) должны учитывать как уязвимости, проявляющиеся на уровне процессов контроля доступа к ресурсам гостевых операционных систем и информационных приложений, так и на уровне гипервизора, который в силу сложности и характера использования сам может являться источником новых уязвимостей. Подробное описание уязвимостей и методик обнаружения атак в бортовых (мобильных) вычислительных системах приведено в работах [61; 62; 63; 64; 65; 66; 67; 68; 69], а программно-технические решения в [70; 71; 72; 73; 74; 75; 76; 77; 78; 79], в которых указано в качестве

основного недостатка систем защиты указано то обстоятельство, что при разработке СЗИ не учитываются особенности высокопроизводительных систем в виде высокой параллельности, асинхронности выполнения программ, более того, не реализованы мультидоменные механизмы защиты на базе процессоров с архитектурой ARM.

Защита информации через систему колец и уровней доступа обычно работает в сочетании с режимами работы процессора. Современные процессоры поддерживают значительно большее количество колец защиты, чем фактически использует операционная система. Чтобы ускорить выполнение программы, производители процессоров AMD, Intel, IBM и чипов с архитектурой ARM добавляют специальные блоки, которые пытаются предугадать действия программы и заранее выполнить операцию. Исследователи в области кибербезопасности из компании VUSec продемонстрировали новый способ обойти существующие аппаратные средства защиты спекулятивного выполнения в современных компьютерных процессорах Intel, AMD и ARM. Новый метод получил название Branch History Injection (BHI). Исследователи также опубликовали PoC-код для эксплуатации уязвимостей (CVE-2022-0001 и CVE-2022-0002), демонстрирующий произвольную утечку памяти ядра [80; 81; 82].

Традиционные средства обеспечения безопасности Intel и AMD не предоставляют возможности, необходимые на современном этапе развития информационных систем. Отсутствуют функции мультидоменной защиты с учетом контекста выполняемых задач, динамические правила политик безопасности (примитивы безопасности, в основном, статические объекты), не применяются мандатный контроль и ролевая модель доступа к информации, не гарантируется безопасность работы в физических и виртуальных средах суперкомпьютерных вычислительных платформ.

Например, злоумышленник с низкими привилегиями в целевой системе может изменить контекст выполнения операций в ассоциативном буфере и в специальных регистрах процессора. В результате может произойти утечка произвольной памяти ядра, что может привести к раскрытию или повреждению конфиденциальной информации.

Для решения данных проблем необходим кардинально новый подход, основанный на использовании технологий UEFI Trusted Boot с применением доработанных и расширенных функций, а также стандарта NIST и концепций Trusted Computing Group. Например, микропроцессор «Эльбрус» [83] поддерживает схему исполнения циклов, которая запускает следующие итерации цикла на исполнение в процессоре до того, как закончились предыдущие. Он поддерживает несколько параллельных стеков. В частности, есть стек вызовов, который полностью поддерживается аппаратно. Однако главный недостаток отечественного серийно выпускаемого микропроцессора состоит в следующем: реализовать многоуровневую изолированную среду для защищенного режима пока с учетом разным режимов работы ЦПУ не удалось.

Решение несовместимо с мультивекторными, суперскалярными, массово-мультитредовыми и гибридными микроконтроллерными (процессорными) блоками, широко используемых в современных суперкомпьютерных платформах.

Проведение исследований BIOS основано на верификационном анализе программного кода ПЗУ, однако при этом не уделяется внимание детальному изучению механизмов сопряжения проектов верхнего и нижнего уровней (взаимодействия приложений с модулями ядра на разных интерфейсных уровнях и базовой подсистемы ввода-вывода).

При проведении же тематических исследований вычислительных комплексов, на которых установлены гостевые ОС, изучаются функциональные возможности программных модулей на разных интерфейсных уровнях взаимодействия, однако нет систематизации методов и принципов проводимого анализа.

Существует возможность авторизации пользователей на сервере с применением сертификатов X.509, которая освобождает системных администраторов от необходимости заводить одних и тех же пользователей, т.к. все учетные записи хранятся централизованно, а при необходимости администратор может мгновенно удалить записи или отозвать сертификат пользователя, заблокировав для него вход в систему [84]. Кроме этого, появляется возможность контроля гипервизора уровня BIOS. Это позволяет отказаться от установки дополнительного программного обеспечения в ОС и делает кроссплатформенным.

Криптопроцессор – это отдельная система на кристалле, *“компьютер в компьютере”*, который преобразует данные таким способом, что они могут быть расшифрованы только на исходном компьютере, под управлением того же программного обеспечения, которое применялось первоначально. Следовательно, обеспечивается максимальный уровень безопасности и невозможность НСД к информации извне. Наличие криптопроцессора устраняет необходимость дополнительного использования физических средств защиты системы [85]. Решение позволит создать защищённый канал связи между конечным устройством (датчиком, контроллером и прочим) и ядром сети.

Концерн «Автоматика» разрабатывает серию защищённых криптопроцессоров. Однако опытные образцы криптопроцессора (как отдельной системы на кристалле) пока на этапе первоначального проектирования. Более того, подобная технология позволяет реализовать лишь функции контроля конфигурации гипервизором уровня BIOS (проект нижнего уровня), но контроля на всех уровнях иерархии выполнения запросов ОС (не решается проблема супервентности в виде связки описания процессов и задания непротиворечивых и однозначных функций безопасности верхнего и нижнего уровней для выбранного класса операций).

Защитой от манипуляций и кибератак все чаще становятся микроконтроллеры. Дело в том, что микроконтроллеры являются основными компонентами в среде управления в подключенных системах. Блокировка отладки предотвращает несанкционированный доступ к микроконтроллеру через интерфейс отладки. Уровень безопасности может быть выбран в зависимости от приложения и требований, однако, выбрав однажды, его впоследствии нельзя уменьшить, а собственная защита кода позволяет настраивать каждый сектор памяти как execute only («только для выполнения»), то есть код в нем может только выполняться, но не записываться [86].

Например, семейство микроконтроллеров STMicroelectronics, которое имеет множество встроенных функций, обеспечивающих их защиту, в том числе [87] :

- защиту от кражи личных данных (защита от манипуляций, защита целостности, отслеживаемость движения продукта);
- отказ в обслуживании данных (регулирование);

- защиту от отслеживания и манипулирование данными и кодом (защита памяти, управление правами доступа, уровень отладки, защита от манипуляций, защита целостности, безопасные обновления прошивки);

Эти функции в основном реализуются их интеграцией непосредственно на кристалле микроконтроллера. Они обеспечивают надежную проверку подлинности (верификацию), целостность платформы и постоянную защиту данных, включая защиту конфиденциальности конечных пользователей, а также комплексную защиту данных.

Однако подобные технологии и методы защиты не реализованы для суперкомпьютеров. Они не учитывают особенности суперкомпьютерных высокопроизводительных систем. Не реализует изоляцию приложений в виде многоуровневой среды исполнения с мультидоменными механизмами защиты за счет использования проксирующих модулей и верификатора команд. Более того, отсутствуют программно-технические решения подобного класса по защите информации в СК, массово используемые в промышленном секторе.

Для перспективных суперкомпьютерных вычислительных комплексов с учетом требований повышенных характеристик по сбоеустойчивости и защищенности вычислительных ресурсов можно выделить основные недостатки технической реализации механизмов защиты:

- не внедрены тегированные вычислительные блоки с мультидоменной многоуровневой защитой и изолированной средой выполнения программ под управлением гипервизора;
- не реализованы функции оценки безопасности состояний системы в виде вычисления результирующей свертки на основе атрибутов протекающих процессов и в виде метаданных конфигурации среды выполнения, с заданной глубиной (масштабируемостью);

- описанные выше технологии и методы защиты не совместимы с отечественными вычислительными комплексами промышленного применения КОМДИВ (серия 1890ВМ6Я, серия 1890ВМ7Я), специализированными СБИС КОМДИВ (серия 1900ВМ2Т, серия 1907ВМ044, серия 1907ВМ028, серия 1907ВМ038), специализированными СБИС (на архитектуре ЕС8430 «Ангара», серия К1927ВВ1Я), вычислительными блоками (юнитами) «Байкал-М», ARM 7/8, «Эльбрус», 4С/8С/16С.

Только использование технологии аппаратной виртуализации в виде многоуровневой “песочницы” с гибкой конфигурацией окружения и контекстом вызовов сервисов, сбором метрики приложений позволяет реализовать механизмы защиты для широкого набора вычислительных платформ с разными микропроцессорами за счет поддержки разных режимов виртуализации и эмуляции, контролировать доступ к вычислительным ресурсам на всех уровнях иерархии выполнения команд СК.

1.4 Тактико-технические характеристики перспективных суперкомпьютеров эволюционного и революционного путей развития технологий

По постановке задачи разработки программно-технических средств защиты перспективных суперкомпьютеров и общей характеристике проектов их создания в результате можно сделать следующие выводы:

1. Кроме пиковой или реальной производительности в экзафлопс операций в секунду, перспективные суперкомпьютеры должны иметь повышенные характеристики подсистемы памяти и коммуникационной сети, а именно, выделяются следующие показатели: объем памяти, пропускная способность памяти, задержка (время) выполнения операции с памятью, пропускная способность сети (точка-точка и бисекционная), задержка выполнения операции с сетью;

2. Выделяются направления эволюционной и инновационной разработки суперкомпьютеров (Таблица 1.1 и Таблица 1.2). Эволюционный подход наиболее прост и позволяет достаточно быстро получить результат, но созданный суперкомпьютер такого типа будет эффективен лишь при решении узкого класса задач и обладать низкой энергоэффективностью. Инновационный подход предполагает проведение фундаментальных исследований и разработку инновационных технологий, что значительно сложнее и требует большего времени;

3. Инновационные технологии разработки перспективных СК в силу жёстких требований по обеспечению энергоэффективности и эффективности по производительности имеют много общего с технологиями создания высокоэффективных бортовых и встроенных систем. Эти технологии принято называть экзамасштабными, они должны обеспечивать возможность создания одноплатных бортовых суперкомпьютеров терафлопсного уровня и одностоечных суперкомпьютеров петафлопсного уровня производительности;

4. При этом нужны модули интеллектуальной памяти с поддержкой механизмов защиты и отказоустойчивости на уровне аппаратуры в виде модулей транзакционной памяти с контроллером, поддерживающим глобальную адресацию в несколько Петабайт ОЗУ для одноплатной и или одностоечной сборки, с реализацией специализированных скоростных сетевых интерфейсов межсокетного и межсегментного обмена данными (например, линейка решений от компаний HP и IBM).

5. Перемещение информации на кристалле обходится достаточно дорого, причем с улучшением обычных кремниевых технологий эти затраты меняются слабо.

6. Если для технологий 40 нм затраты на перемещение 240 fJ/bit/mm, то даже для технологии сборки 10 нм этот показатель будет 115 – 150 fJ/bit/mm. Поэтому моноблочное исполнение узла или миниатюризация сборки плат в стойку применяется как в бортовых экзамасштабных вычислительных системах, так и в стационарных. Иначе затраты по энергетике с ростом производительности будут колоссально увеличиваться;

7. Основные проблемы создания перспективных СК: повышение производительности системы в целом на три порядка в условиях ослабления влияния закона Мура на производительность отдельного процессорного ядра; минимизация энергетических потерь и потерь производительности;

8. Обеспечение защищенности на качественно новом уровне, что связано с новыми физическими эффектами, возникающими из-за сильной миниатюризации, роста объема оборудования и требований по устойчивости к информационным атакам; повышение продуктивности программирования в несколько десятков раз.

Таблица 1.1 – Метрики систем, создаваемых с применением “тяжелых” процессорных ядер

Системная характеристика	Годы внедрения системы			
	2009	2011	2015	2018
Общая пиковая производительность	2 PF	20 PF	100-200 PF	1 EF
Общий объем оперативной памяти	0.3 PB	1 PB	5 PB	10 PB
Пиковая производительность узла	125 GF	200 GF	400 GF	1-10 TF
Пропускная способность памяти узла	25 GB/s	40 GB/s	100 GB/s	200-400 GB/s
Количество ядер в узле (параллелизм узла)	12	32	0(100)	0(1000)
Пропускная способность сетевого интерфейса узла	1.5 GB/s	10 GB/s	25 GB/s	50 GB/s
Количество узлов в системе	18,700	100,000	500,000	0(Million)
Количество ядер в системе (общий параллелизм)	225,000	3 Million	50 Million	0(Billion)
Общая мощность потребления	6 MW	~10 MW	~10 MW	~ 20 MW
Объем внешней памяти	15 PB	30 PB	150 PB	300 PB
Пропускная способность ввода-вывода	0.2 TB/s	2 TB/s	10 TB/s	20 TB/s
МТТП, время между прерываниями по сбою или отказу системы	Days	Days	Days	0(1Day)

Таблица 1.2 – Метрики систем, создаваемых на этапах проекта “легкого” направления

Системная характеристика	Годы внедрения системы				
	2004	2007	2012	2015	2019
Наименование системы/этапа	BG/L	BG/P	BG/Q (ONE)	TWO	THREE
Общая пиковая производительность	0.37 PF	1 PF	27 PF	309 PF	1127 PF (1.127 EF)
Общий объем оперативной памяти	0.034 PB	0.151 PB	2.147 PB	8.590 PB	25.770 PB
Пиковая производительность узла	5.6 GF	14 GF	205 GF	1.178 TF	4.301 TF
Пропускная способность памяти узла	5.6 GB/s	13.6 GB/s	42.6 GB/s	54 GB/s	86 GB/s
Объем памяти узла	0.5 GB	2-4 GB	16 GB	32 GB	64 GB
Количество ядер в узле (параллелизм узла)	2	4	16	32	96
Тактовая частота (GHz)	0.7	0.85	1.6	2.3	2.8
Количество запускаемых операций за такт в ядре	4	4	8	16	16
Пропускная способность сетевого интерфейса узла	2.1GB/s –3D torus 0.7GB/s – tree	5.1GB/s-3D torus 1.7GB/s-tree	40 GB/s-5D torus 4 GB/s extline	25 GB/s	50 GB/s
Количество узлов в системе	65536	73728	131072	262144	262144
Количество ядер в системе (общий параллелизм)	0.13 Million	0.3 Million	2 Million	8 Million	25 Million
Общая мощность потребления	2.5MW	4.8MW	8MW	30MW	40MW
Количество узлов в стойке	1024	1024	512	1024	1024
Количество стоек	64	72	256	256	256
Пиковая производительность стойки	5.7 TF	14 TF	105 TF	1.206 PF	4.404 PF

Применяемые основные решения проблем создания суперкомпьютеров: резкое повышение параллелизма аппаратных средств с одновременным снижением накладных расходов, приходящихся на выполнение одной операции; переход на новые высокопараллельные модели организации программ с глобально адресуемой памятью, обладающие высокой асинхронностью параллельных процессов и развитыми средствами их синхронизации, динамическим параллелизмом;

использование новых технологий передачи данных, новых конструктивов 3D сборки электронных блоков, специализация аппаратных средств и программного обеспечения, интеллектуализация методов обеспечения отказоустойчивости [88].

Полное решение проблемы защиты – разработка суперкомпьютеров с аппаратно обеспечиваемыми уровнями защиты программ и данных. В настоящее время реализуются четыре уровня защиты (пользовательский, исполнитель ОС, ядро ОС, программа начальной загрузки). Например, так сделано в массово-мультитредовых суперкомпьютерах Cray XMT (США) и СТ-2 (Китай), предназначенных для работы в системах анализа больших объемов данных и для активной работы с данными из сети (приложения аналитических и военных центров, бизнеса, науки и социологии). Рассматриваются уже и варианты 8-уровневой защиты [89; 90; 91].

Как средство осуществления атаки суперкомпьютер за счет высокой производительности позволяет планировать и осуществлять массовые атаки с адаптивным к срабатыванию системы защиты поведением. Для такого типа задач также лучше подходят суперкомпьютеры, ориентированные на эффективную работу с большими наборами данных и мощными информационными потоками [92; 93; 94; 95].

Анализ установленных суперкомпьютерных мощностей ведущих ведомств США (Министерства энергетики (DoE, ядерный оружейный и исследовательский комплексы), Министерства обороны (DoE, военные научно-технические центры), Национального аэрокосмического агентства (NASA), Национального научного фонда (NSF)) показал:

- Значительная часть производительности приходится на заказные суперкомпьютеры линеек Cray XT/XE/XK и BlueGene L/P/Q. При этом более популярны суперкомпьютеры Cray с заказными коммуникационными сетями производства этой фирмы. Типовой уровень производительности – 10 Пфлопс;

- Происходит стандартизация кластерных решений. В этом процессе существенна роль фирмы Appro, предложенные решения этой фирмы заслуживают тщательного изучения. Кластерные суперкомпьютеры в подавляющем числе случаев относятся к классу “capacity”, т.е. ориентированы на решение множества задач, а не одной рекордной задачи. Доля суперкомпьютеров такого класса в крупных вычислительных центрах в настоящее время по мощности сопоставима с суперкомпьютерами класса “scalability”;
- Явно заметна тенденция разделения суперкомпьютеров на вычислительные, гибридные, интенсивной работы с данными (data intensive computing), анализа данных и виртуализации. Это даже заметно в решениях по кластерным суперкомпьютерам, что удивило. Наиболее интересны появившиеся кластерные суперкомпьютеры для интенсивной работы с данными, например – кластер Gordon (Центр San Diego, фирма Appro).

Такой успех не случаен. Центр San Diego, как оказалось, даже специализируется по этому направлению суперкомпьютеров для работы с данными, являясь головной организацией по таким суперкомпьютерам в программе XSEDE NSF по суперкомпьютерным вычислениям. Для таких суперкомпьютеров характерно использование многосокетных вычислительных узлов (4 процессора и более в узле), многорейловых сетей с повышенной пропускной способностью и иерархической организацией (макроузлы), использование в узлах твердотельных дисков терабайтового объема и средств программной виртуализации.

Использование суперкомпьютеров и суперкомпьютерных центров ведется в США в рамках специальных программ, для каждого ведомства - своей. Эти программы имеют централизованное управление, организуют совместное использование ресурсов разных центров.

Средства сетевой защиты высокие, вплоть до использования выделенных линий связи, но централизация управления этими программами, по-видимому, дает шанс обнаружения уязвимостей этих суперкомпьютеров [96; 97; 98; 99].

Для информационных систем в США и Китае создаются мощные заказные специальные массово-мультитредовые гибридные суперкомпьютеры экзафлопсного уровня производительности, которые могут эффективно работать с большими наборами данных и мощными информационными потоками. Кроме прямого использования таких суперкомпьютеров в смысле рассматриваемого вопроса кибербезопасности эти суперкомпьютеры являются мощным кибероружием защиты и нападения [100; 101; 102].

В США этим занимается АНБ совместно с фирмой Cray, Национальной лабораторией Сандиа (SNL) и Тихоокеанской Северо-Западной лабораторией (PNNL) – суперкомпьютеры линейки Cray XMT.

В Китае – это проект СТ-2, который выполняется в Университете оборонных технологий (NUDT), является обобщением и развитием американских проектов Cray XMT и ParallelX, а также российского проекта «Ангара». В США и Китае сформированы кибервойска со своим командованием. В Китае – это несколько тысяч человек, входящих в военную разведку. Вместе с агентами по всему миру количество этих сил доходит до миллиона. Поставленный в данной работе вопрос об изменении обеспечения кибербезопасности, когда на защищаемой и атакующей стороне появляется суперкомпьютер, требует, конечно, дополнительных исследований, причем не разовых, а постоянно ведущихся [103; 104].

Проведенное рассмотрение суперкомпьютерных ресурсов США и КНР позволило узнать ряд полезных фактов, что будет использовано непосредственно на практике создания отечественных суперкомпьютеров.

Это в особой степени касается суперкомпьютеров, ориентированных не столько на вычисления, сколько на интенсивную работу с данными.

Суперкомпьютерные центры Китая оказались пока более закрытыми для изучения. Здесь требуется более кропотливая работа и время. Ожидаемый результат стоит того, поскольку Китай находится в роли “догоняющего” США по суперкомпьютерным технологиям и ведению кибервойн, а это предполагает применение нестандартных ходов, резко сокращающих отрыв. Собственно говоря, зание в России таких шагов не помешало бы.

Также явно необходимы меры по ограничению сложившегося монополизма в данных областях, что сильно вредит работам, чему есть конкретные примеры, даже такого типа, которые уже нанесли серьезный урон России в области обеспечения кибербезопасности и ведении кибервойн.

В заключении аналитической главы обратим внимание на организационные меры, предпринятые в США для координации работ по нанотехнологиям, чтобы сократить отставание от передовых суперкомпьютерных технологий и начать перенимать опыт. Страна разделена на три зоны.

Первая зона: Головная организация - Университет Нотр Дам, он располагается в штате Индиана, рядом со штатами Мичиган, Иллинойс, Огайо, Северной и Южной Каролины, Мэриленд, Вирджинии, Джорджии, Округом Колумбии и штатом Нью-Йорк. Здесь находятся головные федеральные университеты, институты, лаборатории, штаб-квартиры спецслужб и военных, Индиана, как и перечисленные выше соседние штаты, входят в состав Федерации Северо-Восточных штатов.

Вторая зона: Федерация Центральных и Южных Штатов: Нью-Мехико, Техас, Юта, Колорадо, Луизиана, Флорида.

Третья зона: Федерация Западных штатов: Калифорния, Орегон, Невада, Аризона, Вашингтон. Самая сильная зона.

При Университете Нотр-Дамм создан один из самых крупных междисциплинарных центров по нанонауке и передовым технологиям в США и Европе. Он осуществляет междисциплинарные фундаментальные исследования новых функциональных возможностей наносистем. Исследования проводят группы ведущих преподавателей четырех факультетов Университета Нотр-Дам (электротехники, вычислительной техники, физики и биохимии).

В январе 2013 года появилось сообщение о создании агентством DARPA и корпорацией SRC (Semiconductor Research Corporation) нового консорциума для ускорения ввода инноваций в области элементно-компонентной базы в США, поскольку современные технологии подошли к пределу миниатюризации. Это будет выполнено в рамках нового проекта, называемого STARnet (Semiconductor Technology Advanced Research Network). Работы будут вестись как по усовершенствованию существующих КМОП-технологий, так и по технологиям пост-Муровской эры, причем отмечается комплексность этих работ, которая бы максимально в сжатые сроки позволила использовать результаты в промышленных изделиях [105; 106; 107; 108; 109; 110]. Кроме DARPA, SRC и университетов в проекте участвует научно-исследовательская лаборатория BBC США (U.S. Air Force Research Laboratory, AFRL) и Ассоциация полупроводниковой промышленности (Semiconductor Industry Association, SIA), а также шесть партнеров из промышленности: Applied Materials, GLOBALFOUNDRIES, IBM, Intel Corporation, Micron Technology, Raytheon, Texas Instruments and United Technologies.

Предпринимаются серьезные попытки продлить использование улучшаемых за счет миниатюризации современных КМОП-технологий еще и на следующее десятилетие, но физические ограничения – неумолимы, нужны долговременные фундаментальные исследования для их преодоления. Организуемые в рамках программы STARnet центры имеют следующую функциональную специализацию:

- **Центр исследований по будущим архитектурам** (The Center for Future Architectures Research (C-FAR)), образован при Мичиганском университете (University of Michigan), ориентирован на разработку архитектур периода 2020-2030 годов. Это направление работ основано на предположении, что специализированные на конкретные приложения архитектуры усилят возможности промышленных технологий и таким образом «продлят жизнь» КМОП-технологий. В работе этого центра будут принимать участие следующие университеты: Columbia, Duke, Georgia Tech, Harvard, MIT, Northeastern, Stanford, UC Berkeley, UCLA, UC San Diego, Illinois, Washington и Virginia;

- **Центр исследований по материалам, интерфейсам и новым архитектурам спинтроники** (Spintronic Materials, Interfaces and Novel Architectures (C-SPIN)), образован при университете Миннесоты (University of Minnesota), ориентирован на рассмотрение технологий памяти и вычислений на базе использования спинов электронов с целью выяснения их потенциала преодолеть проблемы современных КМОП-технологий. В работе центра будут принимать участие университеты: UC Riverside, Cornell, Purdue, Carnegie Mellon, Alabama, Iowa, Johns Hopkins, MIT, Penn State, UC Santa Barbara, Michigan, Nebraska и Wisconsin;

- **Центр функционально ускоренного проектирования наноматериалов** (The Center for Function Accelerated nano Material Engineering (FAME)), образован при Калифорнийском университете Лос-Анжелеса (University of California, Los Angeles), ориентирован на изучение

необычных материалов, включая наноструктуры со свойствами квантового уровня. Исследования нацелены на поддержку создания аналоговых логических элементов и памяти для реализации вычислений «за пределами двоичных вычислений». В работе центра будут принимать участие университеты: Columbia, Cornell, UC Berkeley, MIT, UC Santa Barbara, Stanford, UC Irvine, Purdue, Rice, UC Riverside, North Carolina State, Caltech, Penn, West Virginia и Yale;

- **Центр технологии систем с низким потреблением энергии** (The Center for Low Energy Systems Technology (LEAST)), образован при университете Нотр Дамм (University of Notre Dame), ориентирован на исследования и разработки в области материалов и устройств с чрезвычайно низким потреблением энергии. В работе центра будут принимать участие университеты: Carnegie Mellon, Georgia Tech, Penn State, Purdue, UC Berkeley, UC San Diego, UC Santa Barbara, UT Austin и UT Dallas;

- **Центр систем на базе информационных технологий наноуровня** (The Center for Systems on Nanoscale Information Fabrics (SONIC)), образован при Иллинойском университете (University of Illinois at Urbana-Champaign), ориентирован на исследования преимуществ перехода от детерминированных моделей вычислений к статистическим моделям [*примечание* – это системы типа аналоговых с использованием квантово-механических процессов (типа D-Wave) или вычислений методом Монте-Карло]. В работе центра будут принимать участие университеты: UC Berkeley, Stanford, UC Santa Barbara, UC San Diego, Michigan, Princeton and Carnegie Mellon;

- **Центр исследований по системам в виде групп агентов терра - уровневом масштаба** (Terra Swarm Research Center (TerraSwarm)), образован при Калифорнийском университете в Беркли (University of California, Berkeley), ориентируется на исследования выполнения приложений распределенных самоорганизующихся системах масштаба

города, состоящих из множества агентов - вычислительных платформ (количеством порядка 10^{12}). В работе центра будут принимать участие университеты: Michigan, Washington, UT Dallas, Illinois at Urbana-Champaign, Penn, Caltech, Carnegie Mellon and UC San Diego.

По представленному в разделе материалу можно акцентировать внимание на следующих аспектах:

- 1) Защищенность сервисов для пользователей: в суперкомпьютерах используются 100 тыс. ядер на расчетную задачу, при отказе одного узла кластера. Задержка в доступе к вычислительным ресурсам критична для задач, связанных, например, с вопросами национальной обороны и т.д.;
- 2) Закон Мура – предел роста производительности (количественное увеличение числа ядер не приводит к значительному качественному росту производительности). Из-за того, что память программ и память данных не могут быть доступны в одно и то же время, пропускная способность канала «процессор-память» и скорость работы памяти существенно ограничивают скорость работы процессора;
- 3) В России в настоящее время отечественные суперкомпьютеры серийно не производятся. Различными предприятиями ведутся работы по созданию и поставкам вычислительных систем кластерного типа высокой производительности, основу которых составляют коммерчески доступные компоненты и сборочные единицы импортного производства. Их отставание от заказных систем, создаваемых за рубежом, по производительности и защищенности на несколько порядков ниже;
- 4) Для современных средств вычислительной техники АС существуют следующие издержки: отсутствуют механизмы мультидоменной защиты с учетом режимов работы

современных микропроцессоров, что не позволяет создать прозрачную изолированную среду при работе в недоверенной среде, низкая защищенность управляющих ОС (внешние специальные воздействия и компьютерные атаки приводят к остановке в работе АС в целом);

- 5) В суперкомпьютерной системе сотни тысяч узлов. У каждого узла множество связей (тысячи ядер из разных узлов суперкластера выделяют для одной задачи, каждое ядро из тысячи может обмениваться данными по сверхбыстрым линкам). Порядок связности не меньше десятков миллиардов (большие математические числа и сложность обработки огромного потока данных);
- 6) Этим обусловлена технологическая проблема для отрасли промышленности в виде установки логического соответствия (супервентности) процесса выполнения программы на суперкомпьютере и изменения состояний его гипервизора и аппаратуры СК при работе на недоверенной оборудовании.

1.5 Постановка задач исследований

На основании вышеизложенного материала **основные задачи** диссертационного исследования могут быть сформулированы следующим образом для противодействия угрозам информационной безопасности в современных суперкомпьютерных вычислительных системах:

- 1) Разработка модели угроз целостности среды выполнения процессов с учетом специфики работы суперкомпьютеров;
- 2) Разработка модели безопасных операций, которая учитывает новый класс угроз, связанных с неоднозначностью состояний выполнения прикладных и системных программ в недоверенной среде стационарных и бортовых суперкомпьютеров;

3) Разработка и исследование метода реактивной защиты суперкомпьютеров, заключающегося в виртуализации среды выполнения процессов на недоверенном оборудовании;

4) Разработка и исследование метода реконфигурации среды выполнения программ проактивной защиты суперкомпьютеров с учетом требований мобильности (встроенности производимых вычислений) и факторов доверительности;

5) Разработка методики тестирования уровня защищенности суперкомпьютеров, основанной на алгоритме “маркерного” сканирования;

6) Разработка программно-технических решений, обеспечивающих надёжную защиту от угроз информационной безопасности, реализуемых с использованием средств скрытого информационного воздействия.

.

.

ГЛАВА 2. ТЕОРЕТИЧЕСКИЕ ПОДХОДЫ И ПРИНЦИПЫ В ОБЛАСТИ СОЗДАНИЯ ЗАЩИЩЕННЫХ СТАЦИОНАРНЫХ И БОРТОВЫХ СУПЕРКОМПЬЮТЕРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ С ИСПОЛЬЗОВАНИЕМ СРЕДСТВ ВИРТУАЛИЗАЦИИ

2.1 Алгебра операций с объектами, обрабатываемыми на суперкомпьютерах

Широко используемые ЭВМ разного класса представляют собой системы с фон-Неймановской архитектурой, а модель выполнения программ – это универсальная машина Тьюринга с лентой вычислений, сдвигами влево и вправо.

Критерий эффективности в классической теории Попека-Голдберга для машин Тьюринга является неоднозначным. В оригинальной работе он сформулирован дважды [111]:

- 1) «статистически преобладающее подмножество инструкций виртуального процессора должно исполняться напрямую физическим процессором, без вмешательства монитора виртуальных машин»;
- 2) «все безвредные инструкции исполняются напрямую физическим процессором, без вмешательства монитора виртуальных машин».

Данные условия не являются тождественными. Они идентичны, только если большинство инструкций, выполняемых виртуальной машиной, являются безвредными, что нельзя гарантировать в общем случае. При доказательстве достаточных условий построения монитора виртуальных машин Попек и Голдберг используют второе определение.

С 70-80-ых годов электронная техника претерпела значительные изменения. С появлением и развитием аппаратной виртуализации успешно внедрены [112; 113]:

- **SMEP (*Supervisor Mode Execution Prevention*)**— предотвращение исполнения кода в режиме супервизора) - это технология, разработанная компанией Intel для защиты компьютера от хакерских атак и других угроз, использующих так называемый "режим супервизора";
- **SMAP (*Supervisor Mode Access Prevention*)**. Она предотвращает запись в память и чтение из нее кода, несанкционированно использующего режим супервизора, тогда как SMEP только предупреждает выполнение этого кода.

Режим супервизора – это привилегированный режим работы процессора, который используется ядром операционной системы. Этот режим также называют режимом ядра. Противоположным ему является режим пользователя, в котором работают пользовательские приложения [114]. Чтобы SMEP работала, кроме соответствующего процессора, требуется использование подходящей операционной системы. Однако SMEP, хотя и значительно усложняет задачу взлома системы, все же не гарантирует полной ее защиты. Поэтому позже (в процессорах архитектуры Broadwell) в целях повышения безопасности и защиты от уязвимостей, не устраненных SMEP, дополнительно была внедрена технология SMAP [115].

Прямое исполнение гостевых инструкций, кодировки которых соответствуют привилегированным инструкциям физического процессора на хосте, невозможно, так как они безусловно вызывают исключение ловушки при исполнении в режиме пользователя, в котором запущены виртуальные машины.

Например, кодировки новых безопасных инструкций архитектуры Intel-64 обычно соответствуют недействительным кодировкам на процессорах предыдущих поколений этой архитектуры. То есть новые инструкции обычно являются привилегированными на процессорах предыдущих поколений, поскольку вызывают исключения [116].

Поддержкой аппаратной транзакционной памяти [117] (*англ.* hardware transactional memory, HTM) заинтересованы многие производители процессоров и различные исследовательские группы. Компания IBM первой разработала систему с аппаратной поддержкой транзакционной памяти в суперкомпьютерах IBM* Blue Gene* [118] и серверах zEnterprise EC12 [119].

AMD объявила о разработке расширения Advanced Synchronization Facility (ASF) [120], представляющего собой вариант аппаратной транзакционной памяти для архитектуры Intel 64.

Компания Sun Microsystems разработала процессор с кодовым именем ROCK, однако данный проект был закрыт. Набор инструкций Intel[®] Transactional Synchronization Extensions (Intel[®] TSX) [121], состоящий из расширений Intel[®] Hardware Lock Elision (Intel[®] HLE) и Intel[®] Restricted Transactional Memory (Intel[®] RTM), был добавлен в процессоры Intel Core шестого поколения [122].

Существует также множество программных реализаций идей транзакционной памяти, которые не требуют специальной поддержки со стороны аппаратуры. Однако изучение существующих решений показывает их крайне низкую производительность [121]. Более того, с появлением аппаратной транзакционной памяти расширение Intel RTM позволяет выполнять вложенные транзакции, но при этом все внутренние пары XBEGIN и XEND не должны создавать новые точки сохранения, то есть откат вложенной транзакции приводит к откату всей цепочки транзакций [123; 124].

Введение расширений транзакционной памяти подразумевает, что семантика всех инструкций, работающих с памятью, а также инструкций, влияющих на состояние процессора, не сохраняемое в точках сохранения, должна быть изменена, чтобы поддержать новый режим исполнения [125; 126].

Чтобы не было критических ошибок, необходимо реализовать многоуровневую “песочницу”, в которой агенты гипервизора функционируют и контролируют выполнение на всех уровнях иерархии обработки команд процессором. Исходя из Теоремы о защищенной гипервизорной ОС, сформулированной и доказанной Д.П.Зегждой [6], можно сформулировать два требования:

Требование 1 виртуализации защищаемых ресурсов: в суперкомпьютерной вычислительной системе все операции приложений гостевой или хостовой ОС над защищаемыми ресурсами R_a должны осуществляться R_h путем виртуализации ресурса в виртуальную среду прикладной ОС. Требование выполняется тогда и только тогда, когда $R_h \cap R_a = \emptyset$.

Требование 2 управления доступом: в суперкомпьютерной вычислительной системе все операции приложений гостевой или хостовой ОС над защищаемыми ресурсами должны контролироваться средствами защиты гипервизора. Требование выполняется тогда и только тогда, когда все операции $opv \in Op$, производимые субъектами $Subject \in Sh$, реализуемые в виде операций субъектов $Subject \in Sa$ над ресурсами r_j , контролируются на соответствие политике безопасности с помощью алгоритмов расширенной идентификации и верификации.

Развивая дальше мысль, можно сформировать теоретические препосылки для разрешимости проблемы супервентности: должны учитываться все возможные режимы работы процессора – реальный

аппаратный режим, защищенный аппаратный режим, реальный режим виртуальной машины, защищенный режим виртуальной машин. Это означает, что мультидоменный защищенный гипервизор должен обеспечивать корректное выполнение (программно с помощью интерпретатора или аппаратно) всех инструкций гостевой системы, а также гарантирует изоляцию виртуальных машин (см. пояснения на рисунке 2.1).

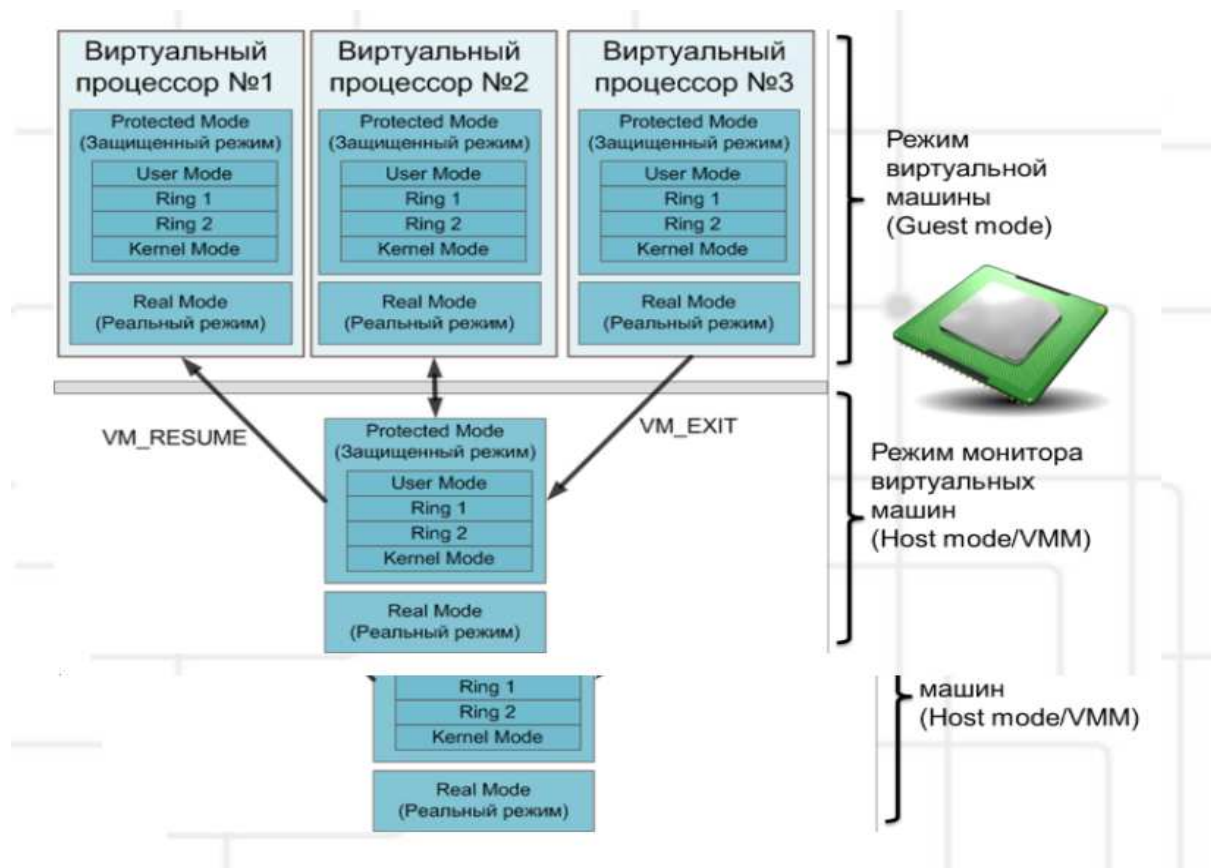


Рисунок 2.1 – Привилегированные и непривилегированные режимы работы на примере семейства процессоров Intel

При этом все общие для гостевой и управляющей систем безопасные инструкции выполняются напрямую физическим процессором (*привилегированные режимы*). Монитор виртуальных машин работает в режиме 0 (в отдельном домене), поэтому любые ошибки интерпретации, утечки памяти, неверные указатели, ошибки адресации сегментов кода данных, кода программ, ошибки станичной адресации не приводят к аварийному завершению работы.

Решение задачи обеспечения измененной мнемоники инструкций для процессоров разных поколений с целью поддержки аппаратной транзакционной памяти заключается в том, чтобы предотвратить прямое исполнение этих инструкций в виртуальном окружении — они должны быть промоделированы программно на виртуальных процессорах (*непривилегированные режимы*).

С учетом развития элементарно-конструкторной базы и аппаратных технологий, сформулируем новые требования и к средствам защиты информации [22; 28; 29]:

- Средства защиты должны контролировать все без исключения информационные взаимодействия;
- Средства защиты должны разрабатываться независимо от прикладных программ и опираться на абстрактное представление информационных взаимодействий;
- Средства защиты должны контролировать информационные взаимодействия на основе четко определенных правил, составляющих формальную модель;
- Должен быть предусмотрен механизм, позволяющий оценить безопасность как настоящего состояния системы, так и спрогнозировать безопасность будущих состояний.

Принцип однородности памяти – главный постулат архитектуры фон-Неймана. Значит, для суперкомпьютерных систем, в которых важна рекордно высокая производительность, требуется принципиально новая архитектура. Совместное использование шины для памяти программ и памяти данных приводит к узкому месту архитектуры фон-Неймана, а именно ограничению пропускной способности между процессором и памятью по сравнению с объёмом памяти.

Из-за того, что память программ и память данных не могут быть доступны в одно и то же время, пропускная способность канала «процессор-память» и скорость работы памяти существенно ограничивают скорость работы процессора. Хранение данных и команд в разных местах решает проблему «стены памяти» на высоконагруженных расчетных задачах [31].

Мы разрабатываем структуру объектов, к которым применяются вычисления, вводим категории “безопасных операций”, что подразумевает выполнение операций с контекстом. Предлагаем “складывать” объекты так, чтобы “опасные сдвиги” не проходили. Необходимо задать класс операций, которые характерны для СК – это операции умножения и деления в виде последовательных сдвигов в тегированных структурах пакетов запросов от клиентов, гостевых и управляющих ОС.

Модель безопасных операций включает в себя три составляющие: команды, данные, временные метки. Пространство операций при вычислении дескрипторов представляет собой многообразие “миров” Крипке. Каждому миру Крипке задаем свой цифровой двойник и формируем базу знаний для обучения. Класс операций, которые характерны для СК, – это операции умножения и деления в виде последовательных сдвигов в тегированных структурах пакетов запросов от клиентов, гостевых и управляющих ОС. С точки зрения теории категорий, мы работаем с объектом *Group* – категория группа. Объектами являются группа в виде кольца вычетов, морфизмами – отображения, сохраняющие групповую структуру. Теория категорий изучает понятия через то, как эти понятия взаимодействуют друг с другом. Мы забываем о том, как эти понятия реализованы, а смотрим лишь на свойства связей, абстрагируясь от типа архитектуры процессора (скалярный, векторный, MIPS, классический x86_64, тайловый, массово-мультитредовый, гибридный и т.д.), разрядности обрабатываемых данных (32-битные, 64-битные, 128-битные и т.д.).

Тип относится к типу класса, когда тип предоставляет определенные операции с определенным ожидаемым поведением. Например, тип **tau** может принадлежать классу Functor, если он имеет определенное поведение, подобное коллекции:

1. Тип **tau** параметризуется над другим типом, который вы должны рассматривать как тип элемента коллекции. Тип полной коллекции тогда похож на `Scheme_type = {Int, String, Bool, ...}`, если вы содержите целые числа, строки или булевы соответственно. Если тип элемента неизвестен, он записывается как параметр типа `a`.

Примеры включают списки (ноль или более элементов), тип "Unknown" (ноль или один элемент типа `a`), наборы элементов типа `a`, массивы элементов типа `a`, все виды деревьев поиска, содержащие значения типа `a`, и т.п.

2. Другим свойством, которое **tau** должно удовлетворять, является то, что если у вас есть функция типа `a -> b` (функция на элементах), тогда вы должны иметь возможность использовать эту функцию и продукт для связанной функции по коллекциям. Вы делаете это с помощью оператора `fmap`, который разделяется каждым типом класса Functor. Оператор фактически перегружен, поэтому, если у вас есть функция `even` с типом `Int -> Bool`, то
3. `fmap`- перегруженная функция, которая осуществляет:

1. Преобразование списка целых чисел в список логических
2. Преобразование дерева целых чисел в дерево логических

это свойство выражается выражением типа `fmap`:

$$\text{fmap} :: (\text{Functor } \mathbf{tau}) \Rightarrow (a \rightarrow b) \rightarrow \mathbf{tau} \ a \rightarrow \mathbf{tau} \ b$$

где теперь мы имеем `tau`, что означает "любой тип в классе Functor".

Определение 1. **Функтор** - это своего рода коллекция (набор конфигураций для рекурсивного вычисления хэша), для которой, если вам предоставляется функция на элементах, fmap вернет функцию в коллекциях.

Определение 2. **Контекст выполнения операции** – это набор кортежей, состоящий из логических переменных. Набор коллекций представляет собой конъюнкцию атомарных предикатов, заданных на множестве кортежей.

Апеллирую к теории функционального программирования, зададим функтор сравнения `is_equal` двух хэш-значений с поддержкой разных шаблонов типов классов:

```
class is_equal
{ private: scheme_type v; public:
  is_equal(scheme_type value) : v(value) {}

  bool operator () (scheme_type x)
  { return x == this->v; } };

scheme_type count_zero(const std::vector< scheme_type>& data)
{ return std::count_if(data.begin(), data.end(), is_equal(0)); },

где scheme_type - это тип данных (int, uint, float, double и т.д.)
```

Невозможно реализовать полную хэш-функцию, которая действительна для всех типов. Вы не можете просто преобразовать объект в необработанную память и хэш-байты. Кроме того, эта идея терпит неудачу из-за технологии `padding` при создании индекса для каждой записи. Из-за этого необходимо учитывать контекст выполнения конкретной операции. Чтобы реализовать универсальный алгоритм вычисления свертки необходимо учитывать разные режимы работы процессора и принцип приведения типов и выравнивания порядков.

Нужна нормировка представления метаданных операций с точки зрения выравнивания границ цифровых структур. Основные принципы алгебраической структуры, на которых стоит вся теория и методология:

- Принцип мягкой силы;
- Самоорганизация;
- Супербезопасность.

Вычисление дескрипторов функции оценки безопасности состояний производится на множестве, представляющем собой мультипликативная группа кольца вычетов по модулю 8.

Классическая логика высказываний - это “черно-белая” модель, высказывания статичны, неизменны во времени. В обычной логике высказываний адекватно не формализуются предложения, в которых явно или не явно присутствуют свойства, истинность которых изменяется со временем. Мы хотим изучать и верифицировать системы, развивающиеся во времени. Предлагаемый подход к описанию операций основан на классификации рисков нарушения информационной безопасности и анализе контекста выполнения исходящих директив, посредством которых могут передаваться данные в обход требований принятой политики безопасности, что приводит нарушению защищенности ресурсов вычислительных узлов.

Носителями анализируемых операций являются множества объектов и субъектов доступа, которым присвоены различные уровни (метки) безопасности. Для контроля уровня безопасности операций порождения новых субъектов, например, операции $Create (Sub_i, O_m) \rightarrow Sub_j$, будем использовать признак неизменности объекта, порождающего субъект доступа.

Этот признак должен выполняться для момента времени $t > t_0$, где t_0 - момент активизации операция, тогда порождение нового субъекта с номером j становится возможным только при выполнении условия $O_m[t] = O_m[t_0]$, где $Subj$ – субъект доступа, O_m – объект доступа, j, m – номера объектов.

Однако для СК признак неизменности не может быть константой неопределенно долго: запускается одновременно огромное количество процессов. Для описания состояний субъектов и объектов доступа нужна темпоральная логика. Необходима модель угроз целостности среды исполнения процессов СК, в которой вместо классической связки «субъект, объект, предикат» реализована новая парадигма написания правил политики безопасности и заданы новые сущности – «субъект», «объект» и «дескриптор оценки безопасности состояний»: для каждой i -ой угрозы вычисляется значение функции оценки безопасности состояний, аргументы которой заданы в виде конъюнкции предикатов из восьми логических переменных, субъектами являются гостевые и управляющие ОС, а объектами – компоненты гипервизора и контроллера транзакционной памяти.

Темпоральная логика CTL (Логика дерева вычислений) – это логика времени ветвления, что означает, что ее модель времени представляет собой древовидную структуру, в которой будущее не определено; в будущем есть разные пути. Темпоральные логики ветвящегося времени рассматривают возможные вычисления (пути на дереве) – траектории на развертке структуры Крипке. Структура Крипке – система переходов с помеченными состояниями и непомеченными переходами. Развертка определяет бесконечные цепочки состояний – возможные вычисления.

Каждое состояние может иметь не одну, а множество цепочек – продолжений, и является корнем своего дерева историй (вычислений).

Грамматика CTL:

Формулы состояний $\phi ::= p \mid \neg\phi \mid \phi \vee \phi \mid E\alpha \mid A\alpha$

Формулы путей $\alpha ::= \phi \mid \alpha U \alpha \mid X\alpha$

формула ϕ состояния s является формулой пути σ , если это состояние s является начальным состоянием пути σ

Структура Крипке M – это пятерка $M = (S, S_0, R, L, AF)$

- S – конечное непустое множество состояний модели.
- $S_0 \subseteq S$ – множество начальных состояний модели.
- $R \subseteq S \times S$ – отношение переходов между состояниями, обладающее свойством тотальности, т.е. $\forall s \in S \exists t \in S ((s, t) \in R)$ (Из любого состояния есть переход).
- AF – конечное множество атомарных предикатов модели.

$L: S \rightarrow 2^{|AF|}$ – функция меток (каждому состоянию отображение L сопоставляет множество истинных на нем атомарных предикатов).

В нашем случае $AF = \{\text{context_id} \mid \text{Dom_id} \mid S \mid \text{Ord} \mid \text{Context_type} \mid \text{TCU} \mid \text{TR}\}$. Пусть задана произвольная формула F_i логики CTL и структура Крипке M . Для каждой подформулы ψ_i формулы F_i алгоритм маркировки выполняет следующие шаги:

- Вычисляется множество S_{ψ_i} состояний структуры Крипке M , в которых выполняется ψ_i .
- Вводится новый атомарный предикат p_{ψ_i} .
- Этим атомарным предикатом помечаются все состояния из S_{ψ_i} .
- Поскольку обработка каждой формулы ψ_i заканчивается введением нового атомарного предиката p_{ψ_i} , и маркировкой этим предикатом всех состояний, на которых выполняется формула, то можно считать, что на каждом шаге алгоритма элементами подформулы являются только атомарные предикаты.

Приступим к построению нумерации программ. Каждый оператор однозначно характеризуется парой – тип (название) и список параметров, среди которых метки операторов, переменные, функциональные и логические выражения.

Нам нужно только закодировать каждое из возможных значений параметров оператора. Метки операторов не нуждаются в особом кодирования, поскольку представляют собой натуральные числа. Поскольку каждая программа вычисляет некоторую функцию, вводимая геделева нумерация порождает некоторую нумерацию функций.

С одной стороны, данная нумерация не является взаимно-однозначной, поскольку мы кодируем синтаксис программы, и любые синтаксически различные программы имеют разные коды. С другой стороны, каждая вычислимая функция вычисляется бесконечным классом программ. Однако такая нумерация обладает фундаментальными и практически полезными свойствами. Пусть A – счетный класс функций оценки безопасности состояний СК: $A = \{f_1, f_2 \dots, f_n\}$.

Даже при том условии, что каждая из функций f_i вычислима некоторым алгоритмом, это не гарантирует существование единого алгоритма вычисления всех функций.

Назовем класс A равномерно перечислимым, если существует двуместная вычислимая функция F такая, что класс A состоит в точности из функций вида $F(n, x)$ для некоторых n из множества N . Функцию назовем универсальной функцией класса A . $nF(n, x)$ – равномерная нумерация класса A . Наша главная цель – показать, что определенная нами нумерация вычислимых функций равномерна по отношению к другим нумерациям и по номеру вычислимой функции в заданной равномерной нумерации можно эффективно найти его номер.

С помощью функции оценки безопасности состояний СК мы сможем в дальнейшем точно вычислить пронумеровать состояния на всем многообразии структур Крипке, а затем выделить классы безопасных и опасных операций.

Лемма 1. Если A - эффективное множество, то для любого эффективного множества B : AB эффективно, а также любое декартово произведение $A_1A_2...A_n$ эффективных множеств эффективно, множество A^* , $A^*=A^n$, всех конечных последовательностей элементов A эффективно счетно.

Доказательство утверждения тривиально и следует из основной теоремы арифметики.

Сформулируем и докажем теорему о перечислимости класса операций.

Теорема 1. Класс A операций на кольце вычетов по модулю m равномерно перечислим, а функция вычисления хэш-значений F - универсальной функция класса A только в том случае, если число уровней иерархии (вложенности) равно 8.

Доказательство. Фактически, операции выполняются над кортежами данных, представляющими собой набор i -ых элементов алгебраической структуры (бит, атомарный предикат, конъюнкцию предиката, любой набор предикативных и функциональных символов, вектор-переменная, скаляр, вещественное число) любой размерности m (независимо от кодировок и позиционной системы счисления). Мы работаем с матрицами, где количество строк равно числу вложенности (иерархии) пространства вычислений N , а i -ый столбец – элемент структуры размерностью m в кольце вычетов $Z(m)$, $i = 1 \dots n$. С учетом расчетной формулы Корнфельда, Каждый сомножитель P_i представляет собой геометрическую убывающую прогрессию:

$$1 - \sum \left(\frac{1}{N}\right)^{n-1}, \quad 1 \dots n, \Lambda = 1/N \quad (2.1)$$

Примечание. Математическое выражение $\sum \left(\frac{1}{n}\right)^{n-1}$, в котором при $n \rightarrow \infty$, представляет собой убывающую геометрическую прогрессию и равно $Q = \frac{1}{n} / (1 + \frac{1}{n})$. Подробное описание приведено в Главе 4 (см. формулу 4.2).

Нам нужно, чтобы значение представляло периодическую дробь любого значения n , то есть с точностью до n -го знака после запятой никаких новых эффектов не было. Более того, выражение должно быть равно нулю (*условие 1* – максимально приближенное значение к нулю), то есть каждый i -элемент произведения должен приближаться к нулю, но при этом надо, чтобы выполнялось и условие равномерности вычислений. С изменением разрядности n , отображение должно быть инвариантным (*условие 2*).

Рассмотрим ряд натуральных чисел, задающих набор возможного числа уровней иерархии : $\{ 1, 2 \dots, 9 \}$.

1 – нейтральный элемент.

Начнем вычисления с 2 и закончим 9:

$$N = 2, Q = 0,33333 = 0,(3);$$

$$N = 3, Q = 0,25;$$

$$N = 4, Q = 0,2; \quad N = 5, Q = 0,1666666 = 0,1(6)$$

$$N = 6, Q = 0,142857;$$

$$N = 7, Q = 0,125;$$

$$N = 8, Q = 0,111111 = 0,(1);$$

$$N = 9, Q = 0,1.$$

Наиболее подходящие значения: 0.1 и 0.(1).

Двум условиям удовлетворяет только решение $N = 8$.

Что и требовалось доказать.

Более того, $nF(n,x)$ – эффективная нумерация класса A , поскольку с изменением размерности параметра n не меняются регулярные отношения между элементами множества, не появляются новые свойства у объектов нашей алгебраической структуры.

Главный результат теоретических выкладок состоит в том, что мы доказали универсальность и равномерность для заданных классов операций функциональных преобразований. Независимо от архитектуры вычислительных узлов СК найдено такое число уровней иерархии ($N=8$), что в алгебраической системе существует инвариант. Если оценивать временные параметры, данное свойство регулярности сохраняется неопределенно долго (от 0 до $+\infty$).

Хеш вычисляется рекурсивно для всех процессов (подпроцессов) и получается результативное значение для штатного измерения трассы прохождения запроса (на основе прошлого опыта, предобучение). В процессе работы идет обучение системы, каждая новая конфигурация классифицируется и в будущем распознается с учетом характерного набора маркеров.

Кодирование признаков идентификации – набор или конъюнкция простых предикатов, на которых задана формула F_i . Подформула ψ представляет собой рекурсивный вызов одной и той же алгоритмической процедуры, только с разным набором характерных признаков и свойств.

Пример шести конфигураций “раскрашиваний” дерева историй вычислений (рисунок 2.2). Выбираем два цвета: Blue – предикат присутствия переходного процесса, Red – предикат изменения типа контекста выполняемых операций.

1 конфигурация позволяет проследить изменения в локальных состояниях на одном уровне иерархии и идентифицировать запуск дочерних процессов в рамках базового процесса с интересующими нас маркерами.

2 конфигурация анализирует все дерево состояний на основе выбранного маркера (режим полного прогона вычислений в случае перезапуска задачи). Заново строится дерево историй.

3 конфигурация анализирует изменение двух параметров в близких ветвях дерева (локальный масштаб, в одном домене).

4 конфигурация отследить одиночное (нерегулярное) изменение параметра в дальних ветвях дерева вычислений.

5 конфигурация регулярно повторяющиеся изменения выбранного параметра идентификации на всей траектории движения.

6 конфигурация изучает ветви дерева, могут со временем в будущем на разных траекториях могут проявиться изменения в двух параметрах. При этом процессы могут быть запущены в разных доменах

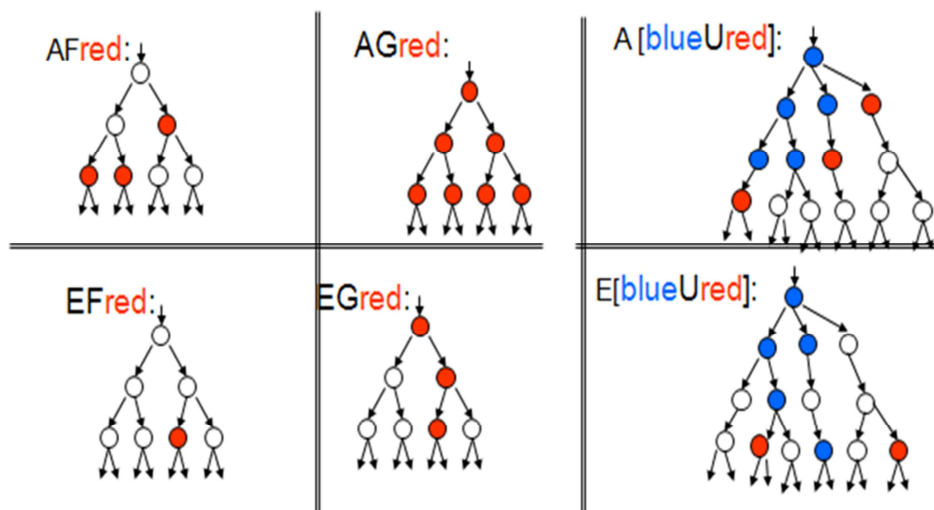


Рисунок 2.2 – Наборы конфигураций

Наборы конфигураций, адаптированных к разным контекстам выполнения операций, в виде матриц правил доступа сохраняется агентами гипервизора, ведущих мониторинг событий ИБ, в памяти контроллера

транзакционной памяти. В процессе обучения обновляются и корректируются правила ПБ. Система динамически эволюционирует и модифицирует наборы модальных правил на основе грамматик темпоральной логики CTL по реагированию на сигнальные события.

Критерии идентификации угроз в семантической трактовке набора атомарных предикатов. Таким образом, гипервизор в связке с контроллером транзакционной памяти собой представляет многоагентную систему.

Существуют проверки, соответствующие принципам работы классической логики, но есть алгоритмы с получением метаданных о работе компонентов системы на разных уровнях иерархии, где, наряду с проверкой целостности в виде изменения значений, вычисленных хеш-функций для каждого процесса, с реализацией контроля доступа в виде маркировки разрешенных переходных процессов при переключениях между доменами защиты и миграции потоков.

Ведется история разных конфигураций системы, распознаются новые события за счет вариантного нового набора предикатов и значений их траекторий на структурах Крипке. Перейдем непосредственно к построению взвешенного мультиграфа-модели выполнения операций СК, представляющего собой дерево. Мы вычисляем такие воздействия, которые обеспечивают безопасное функционирование СК, и блокируем опасные и подозрительные.

2.2 Оптимизация, нормировка характеристической функции вычисления дескрипторов и модальные правила политик безопасности

Для оптимизации и нормировки необходимо задать точно безопасный объем данных. Аксиоматика нашей алгебраической структуры следующая:

- Заданы операции умножения и деления (циклические сдвиги вправо или влево);

- Множество вычислений дескрипторов – это кольцо вычетов по модулю $m = 8$;
- Кольцо должно быть симметрично относительно выполнения операций умножения и деления;
- Запрос для любого тапа процессора с классической фон-Неймановской архитектурой представляют собой совмещенную тегированную структуру, в которой сегмент данных и сегмент кода программ хранятся в одних участках ОЗУ. При этом L-операнд для рекурсивного вычисления ψ_i хранится с секции данных, R-операнд для рекурсивного вычисления ψ_i хранится в секции кода.

Принцип супервентности заключается в отсутствии различий одного вида при отсутствии различий другого вида: отсутствие различий в наборе дескрипторов процессов при отсутствии изменений в конфигурации программно-аппаратных агентов. Существует взаимно-однозначное логическое соответствие между изменением набора вектор-переменной, хранимой в генеративных таблицах гипервизора, картой состояний транзакционной памяти (уровень аппаратуры) и значением хэш-функции (уровень ПО).

В процессе прохождения запроса, ключевая строка не меняется в штатном режиме. Если меняется, то значит, осуществляется модификация маршрута прохождения запроса сторонним агентом.

А-свойство – тождественность значений дескрипторов оценки безопасности для i -го процесса на n -уровне иерархии (выполнение программ на уровне ОС, идентифицируем каждый процесс значением хэша, формализовано в виде значения хэш-функции).

В-свойство – конфигурация программно-аппаратных компонентов (формализовано в виде набора матриц в виде 8 16-разрядных ключевых строк, на основе которых вычисляется значение функции).

Фактически мы вычисляем сжатый образ цифрового двойника. С учетом двоично-десятичного формата представления данных получаем размерность границ выравнивания в виде числа по модулю 4. Число значащих разрядов равно 4. Это означает, что данные кэшируются и сохраняются в буфере ассоциативного поиска в виде страниц размером $10^4 = 1000 \times 10$. Поддержка глобально адресуемой памяти в суперкомпьютерах реализована в виде кластерных файловых систем с суперблоками размером, кратным 1 Мбайт вместо блоков размером 4 -128 Кбайт (в традиционных файловых системах). Процессор разделяет поток входных данных на блоки размером 10^4 , причем каждый суперблок имеет размер Мбайт. Тогда общий объем таблиц 10^{10} байт.

Пусть используется общая память для команд и данных. Команды и данные хранятся в одной и той же памяти и внешне в памяти неразличимы. Тогда это позволяет производить над командами те же операции, что и над числами. Так, циклически изменяя адресную часть команды, можно обеспечить обращение к последовательным элементам массива данных. Команды дочерних процессов могут получить доступ к командам и данным базового (родительского) процесса. Если нет возможности выполнить опасные сдвиги и нарушить сегменты адресации, то исполняемый код хранится и обрабатывается в разных доменах защиты гипервизора, а мощность множества увеличивается вдвое.

Тогда ни в режиме прямой, ни в режиме косвенной адресации один процесс не может менять коды данных и программ другого.

При этом нужно учитывать верхний или нижний регистр адресов (сегмент кода данных или кода команд, совмещенный формат представления структуры в ОЗУ). Тогда объем точно безопасных

вычислений возрастает вдвое и нужно ввести дополнительно $2 \cdot 4 = 8$ регистровых полей в виде значащих разрядов n -мерного вектора $\Psi(n) = 8$, чтобы контролировать границы адресации и смещения в конвейере обрабатываемых запросов и не допускать выполнение «ложных транзакций».

Таким образом, размерность управляющего параметра k равна 8. При этом не смешиваются биты (верхний и нижний регистры) указателя на ячейки памяти ОЗУ.

Если это условие не выполняется для функтора сравнения, то нарушаются границы сегмента данных и кода, некорректно осуществляется рекурсивное вычисление дескриптора: при $N < 8$ остаются незаполненные разряды (в них можно записать нули или мусорные данные), а при $N > 8$ произвести нормировку обрабатываемых данных (BDC-формат и приведение типов операндов) невозможно.

Чтобы получить такой вид представления набора данных в виде кортежей из n -логических переменных, минимальный набор(базис) равен 8. При количестве итераций кратно 8, мы увеличиваем объем данных рекурсивных вычислений, но сами эффекты и закономерности не меняются.

При уменьшении количества итераций (меньше 8 уровней иерархии обработки запросов) автоматическое выравнивание обрабатываемых данных в BDC-формате приводит к тому, что остаются неинициализированные отдельные биты или группы битов, что позволяет произвести опасные сдвиги в СК.

Оптимизация вычислений дескрипторов обусловлена восемью интервальными ограничениями в цикле обработки для получения результативной выборки, за счет использования только восьми значащих разрядов (а не бесконечно больших чисел) для хранения и обработки результатов. Исходя из сказанного выше, сформулируем условие

разрешимости научной проблемы установки логического соответствия (супервентности) процесса выполнения программы на суперкомпьютере, изменением состояний компонентов гипервизора и аппаратуры СК: число уровней иерархии обработки запросов гипервизором N при взаимодействии с контроллером транзакционной памяти должно быть равно восьми.

Условие разрешимости. Число уровней иерархии обработки запросов гипервизором N при взаимодействии с контроллером транзакционной памяти должно быть равно восьми.

Изменение состояний процессов на верхнем уровне приводит к координированному изменению состояний на нижнем уровне, существует отношение подобия. При использовании двух колец защиты ($N=2$) и при отсутствии физически разделенного хранения программ и данных процесс может получить доступ к «чужому» сегменту, повысить уровни привилегий.

Правила ПБ задаются в виде модальных конструкций в терминах темпоральной логики:

— **EF ($\delta > 1$)** не существует такого режима работы для квантора пути w , при котором изменение δ -параметра превысит значение 1. δ -параметр при переходе из s_i в s_{i+1} представляет собой выражение:

— если повышение привилегий на траектории вычислений $\text{Diff}\{\text{Priv}(s_i), \text{Priv}(s_{i+1})\} > 0$, то $\delta = \max\{\text{Priv}(s_i), \text{Priv}(s_{i+1})\} - \text{Priv}(s_i) = 1$

— если понижение привилегий на траектории вычислений $\text{Diff}\{\text{Priv}(s_i), \text{Priv}(s_{i+1})\} < 0$, то $\delta = \text{Priv}(s_{i+1}) - \min\{\text{Priv}(s_i) = 1$,

— если нет изменений на траектории вычислений в уровне привилегий, то $\delta = 0$.

Значение уровня привилегий представляет ряд десятичных чисел {0,1,2,3,4,5,6,7}. Все процессы распределены по доменам в соответствии с уровнем привилегий. Если происходят изменения в структуре “выполнения процессов”, то изменение уровня привилегий индексируется на единицу, то есть можно мигрировать только между соседними доменами.

AG (process_is_on $\Rightarrow$ P Hyper_Permission, delta) в любом режиме, прежде чем запустится новый процесс, то на это обязательно предварительно была получена санкция со стороны гипервизора.

AG (req $\Rightarrow$ (start U ack)) во всех режимах, требующих изменения привилегий для запуска процесса, после того, как запрос req установится, он никогда не будет обработан, пока на него не придет подтверждение. Условие подтверждения – delta = 1

AG [S7 $\Rightarrow$ AX[A(Ord =False U S8) $\vee$ AG $\neg$ L8] между S7 и S8 всегда выполняется свойство Ord = 0, поскольку на данном уровне иерархии функционирования системы здесь не запускаются любые порожденные процессы, а только инициализирован и работает один базовый процесс – верификатор команд.

FG (Start_mon & Ord =False) – с момента запуска верификатора команд (событие Start_mon), всегда выполняется условие Ord=0

G delta (tr & $\neg$ tr) – всегда будет вычисляться функция delta для тех случаев, когда в системе не возникают ошибки, то есть значения tr = {00, 11}. В этом случае вычисление не проводится, запрос вообще не обрабатывается и удаляется. В остальных случаях, когда Событие 10 – начинается переходный процесс, событие 01 – завершается переходный процесс завершился, проводится контроль на соответствие уровня доступа к запрашиваемым данным с уровнем привилегий инициатора доступа.

$\neg$ F decrypt (hcode_TCU=False) – никогда не проводится вычисление дескриптора оценки безопасности для тех состояний, когда нет активных

задач, то есть два старших бита поля TCU (спецификатор hcode_TCU) обнулены. Два младших бита TCU (спецификатор lcode_TCU) кодируют состояние выполнения. Бит В – состояние блокировки.

–F terminate (AG si & Context_type = 2 & tr=0 OR tr = 3 || B = 0) –

для всех состояний si на всех траекториях с ошибками запрос принудительно завершается.

2.3 Интеграция алгебраической структуры с архитектурными доработками суперкомпьютеров революционного пути развития

Согласно тезису Чёрча - Тьюринга, любой разрабатываемый алгоритм можно реализовать на машине Тьюринга. Если компилятор языка, реализующий алгоритм, замедляет исполнение алгоритма полиномиально, то определения классов Р для этого языка и для машины Тьюринга полностью совпадают. В качестве языков программирования неслучайно выбран Рефал и ассемблер. Код на ассемблере допускает преобразование в машину Тьюринга с небольшим полиномиальным замедлением. Рефал позволяет заменить команды сдвиг вправо, сдвиг влево, начало и окончание на машинной ленте машины Тьюринга М на эквивалентные функциональные выражения в виде подстановок слева (L-оператор), справа (R-оператор), селекторы начала и конца выполнения блока кодов.

Он позволяет управлять командами и распараллеливать сборки исходящих директив, при этом время задержки на генерацию и выполнение по тактам полиномиально зависит от длины входных слов.

Временная стоимость в нашей разрабатываемой модели безопасных операций будет равна 8 этапам (итерациям) вычислений.

Пространственная стоимость вычислений равна $2 * 10^7 = 20$ МБайт. Вместо обработки огромного потока информации, анализируется совсем небольшой участок кода, представляющий собой префиксы в заголовках пакетов запросов.

Рассмотрим архитектурные особенности современных СК, в которых реализованы принципиально новые технологические доработки, чтобы можно было перейти от теоретико-множественного описания и классов операций к проектированию экспериментального образца (прототипа).

В рамках проекта 863/ИТ по созданию защищенных суперкомпьютерных вычислительных комплексов серии СТ-2 (ОКР шифр «Удар грома») китайские специалисты учли и внедрили основные авторские принципы, модели, технологии и методы в области создания защищенных суперкомпьютерных вычислительных комплексов [87; 88; 89; 90; 91; 96; 97]. Работа по СТ-2 ведется при поддержке ИСТ (Институт вычислительной техники Академии наук Китая). Подробное техническое описание внедренных авторских программно-технических решений в виде опытных образцов и серийно выпускаемой линейки продукции вместе с чертежами (схемами) приведено в Приложении 4.

Общие результаты опытно-конструкторских разработок таковы:

1. Используемое кодовое название СТ-2 (полное название Цинь Тао – 2) – это идиома на китайском языке наименования «Главный проект в интересах стратегических исследований». Изначальная область приложений СТ-2, который рассматривается в первую очередь как суперкомпьютер для информационных систем, – создание глобальных центров сбора, обработки и хранения больших потоков информации.

2. Спроектированный нами макет вычислительного устройства суперкомпьютера в защищенном исполнении собран в виде опытного образца массово-мультитредового процессора и материнской платы с поддержкой MTDF-архитектуры. В 2020 году опытные образцы вычислительных модулей запущены в серийное производство.

3. Подсистема обеспечения информационной безопасности суперкомпьютеров серии Tian-He/CT-2 реализована на базе мультидоменного гипервизора и коммутаторов Arch.

4. Главный микропроцессор CT-2 – мультитядерный массово-мультитредовый микропроцессор, базируется на архитектуре микропроцессора J7 российского проекта «Ангара», но с элементами наиболее мощного варианта микропроцессора J10 этого же проекта. Благодаря использованию массово-мультитредовой потоковой архитектуры, он должен обладать повышенной толерантностью к задержкам выполнения операций с памятью и сетью, эффективно поддерживать работу с моделями программ в виде статических графов потоков данных. Он образно характеризуется тремя иероглифами, изображенными на корпусе микросхемы – “тигр” (вычислительная мощность), “ястреб” (стремительность, наличие множества асинхронных тредов), “дракон” (символизирует прорывной характер этого микропроцессора, дракон – символ прогресса). Характер доработки микропроцессора (почему ассоциация с тигром) – введение мощных SIMD операций над короткими векторами, а также усиление вычислительных возможностей по типу того, как это сделано в графических процессорах (GPU). В силу этого он будет успешно справляться с обработкой в реальном времени множества потоков данных и эффективно работать через единое адресное пространство с огромным объемом памяти в несколько десятков петабайт даже в режиме интенсивной нерегулярной работы с ней, обладать исключительной отказоустойчивостью и готовностью. Объем доступной памяти программе пользователя – 32 ПБайт, физическая память – 64 ПБайт, что сделано для горячего резервирования.

3. Создаваемый суперкомпьютер СТ-2 с несколькими уровнями иерархии, высоким параллелизмом и глобально адресуемой памятью предполагает наличие мощного программного обеспечения с динамическим распараллеливанием программ, средствами оптимизации, учитывающими неоднородность доступа к глобально адресуемой памяти.

По уровню программирования пользователя изучаются и развиваются подходы непроецедурных функциональных языков. Большой интерес проявляется к российскому языку Рефал и американскому языку AMBIT. Оба этих языка – функциональные, функции описываются непроецедурно, правилами подстановок на базе алгорифмов Маркова. Практическая отработка ведется на уже имеющихся суперкомпьютерах, в которых входят большие фрагменты на мультитредовых микропроцессорах FT-1000 (например, на суперкомпьютере Tianhe-1A/2A).

В рамках проекта JST CREST “Разработки новых НРС-технологий” 5-го Базового плана развития науки и технологий по созданию суперкомпьютерных вычислительных комплексов (ОКР шифр «Стрела времени») японские специалисты успешно реализовали основные результаты нашей группы разработчиков, опубликованные в отечественных и международных журналах, освещенные во время выступлений на конференциях и семинарах [86; 91; 92; 93; 94; 96; 97].

Детальное техническое описание внедренных авторских программно-технических решений в виде опытных образцов и серийно выпускаемой линейки продукции вместе с чертежами (схемами) приведено в Приложении 4. Давайте рассмотрим подробнее. Головной исполнитель работ – Технический институт исследований и разработок (Technical Research & Development Institute). В разработке участвует фирма Fujitsu, Hamamatsu Photonics, Alnair laboratories а также гражданские соисполнители – RIKEN (ранее разрабатывали К-компьютер) и Токийский технологический институт (ранее разрабатывали Tsubame 2.0).

Кстати, проекты Tsubame и К-компьютер в продвижении к суперкомпьютеру «Стрела времени» считаются как соответственно начальный этап («утренняя ласточка», начало полета) и этап расцвета цивилизации (результат развития 1000-летия). Проект «Стрела времени», в свою очередь, образно считается прорывом «на миллионы лет», т.е. это начало внедрения прорывных технологий будущего.

Принципиальные особенности вычислительного узла суперкомпьютера «Стрела времени» и основные результаты проектно-конструкторских решений состоят в следующем:

1. В центре вычислительной платы расположен гибридный микропроцессор, который содержит: сетевой процессор сети вычислительных узлов; сетевой процессор подсистемы ввода-вывода; массово-мультитредовый микропроцессор с ядрами средней сложности и асинхронной мультитредовостью. К этому микропроцессору подключена основная память типа DRAM вычислительного узла, а также память типа NVRAM (энергонезависимая память, ее можно рассматривать как сверхбыстрый диск).
2. В состав вычислительного узла еще входят обрабатывающие микропроцессоры. Упоминались такие микропроцессоры двух типов – для вычислительных задач, когда доля вычислительных операций высока (локализация и интенсивность работы с данными может быть разной), а также для невычислительных задач, когда ведется логическая обработка и, как правило, для этих задач характерна плохая пространственно-временная локализация обращений к памяти, высокая интенсивность операций работы с памятью.
3. К этим микропроцессорам подключена в 3D-сборке DRAM-память, возможно и NVRAM-память. Это можно рассматривать как сверхбыструю память выше уровня L3 кэш-памяти.

4. Микропроцессор счетных задач содержит традиционно применяемые в данной области типы процессорных ядер: суперскалярные ядра; ядра графических процессоров (асинхронность тредов типа WARP и синхронность тредов внутри каждого WARP-а); легкие ядра типа применяемых в суперкомпьютере BG/Q, в которых используются мощные устройства обработки коротких векторов SIMD с шириной обработки 512 разрядов.
5. Спроектированный нами макет вычислительного устройства суперкомпьютера в защищенном исполнении собран в виде опытного образца сборочного конструктива. В 2021 году опытные образцы вычислительных модулей запущены в серийное производство.
6. Подсистема обеспечения ИБ суперкомпьютеров серии Tsubame 3.0 реализована на базе мультидоменного гипервизора и коммутаторов Tofu.

У японских разработчиков большое разнообразие ядер. В этом микропроцессоре будут развиты средства виртуализации оборудования, поскольку оно разнотипно и в одной задаче работы для всех типов ядер может и не быть. Таким образом, для невычислительных задач будет применена двухуровневая мультитредовость, одна – в обрабатывающем процессоре, а другая – непосредственно в модулях памяти.

Следует отметить применяемые элементно-конструкторские технологии. Главная особенность – это нанотрубки, в которых используется высокотемпературная сверхпроводимость, она обеспечивается при температуре 20 градусов по Цельсию. Японские разработчики достигли значительного успеха. В то время, когда обычные технологии позволяют передавать сигнал по соединениям с потреблением 100-150 fJ/bit/mm, по нанотрубкам можно передавать сигнал с потреблением 0.6-1 fJ/bit/mm.

Еще одна особенность – будет применена открытая недавно фирмой IBM технология передачи информации, условно называемая Holley, с применением маломощных лазеров, фотоприемников, оптических каналов в кремнии и массивов микролинз. Такая же технология планируется к применению в суперкомпьютере для АНБ США. Еще известно об успехах японских специалистов в области терабитных сетей с уплотнением передачи информации по оптике на разных длинах волн. Достигнут уровень использования одновременно в канале 40 длин волн.

Рассмотренные в данном разделе суперкомпьютеры – это специализированные экзафлопсные суперкомпьютеры для решения информационных задач с плохой пространственно-временной локализацией обращений к памяти:

1. Все суперкомпьютеры на базе многоядерных микропроцессоров, в ядрах этих микропроцессоров аппаратно поддерживаются 128-256 тредов, предусмотрены специальные меры для обеспечения одновременного выполнения из микропроцессора огромного количества обращений к памяти и сети, что и обеспечивает высокую толерантность к задержкам.

2. Такие суперкомпьютеры позволяют решать задачи на больших графовых наборах данных, которые вообще не решаются на обычных суперкомпьютерах, поскольку в них на аппаратном уровне не поддерживается работа с глобально адресуемой памятью.

3. Производительность таких суперкомпьютеров в 100 (1000) раз выше, чем обычных вычислительных кластеров и суперкластеров.

2.4 Выводы по разделу

- 1) Временная стоимость в нашей разрабатываемой модели безопасных операций будет равна 8 этапам (итерациям) вычислений;
- 2) Пространственная стоимость вычислений равна $2 * 10^7 = 20$ МБайт. Вместо обработки огромного потока информации, анализируется совсем небольшой участок кода, представляющий собой префиксы в заголовках пакетов запросов;
- 3) Доказана теорема о равномерной перечислимости функций для оценки безопасных состояний СК. При этом стационарные и бортовые суперкомпьютеры рассматриваются как объект исследований впервые;
- 4) Сформулировано условие разрешимости проблемы установки логического соответствия (супервентности) процесса выполнения программы на суперкомпьютере, изменением состояний компонентов гипервизора и аппаратуры: при числе уровней иерархии равном 8, процессы на всех уровнях в СК наблюдаемы, в других случаях – нет, что затрудняет обнаружение и идентификацию угроз со стороны СЗИ;
- 5) Разработаны и введены новые теоретические и научно-методические принципы, решены проблема супервентности, тем самым положено начало развитию новых теоретических, методических, научно-практических решений и подходов для создания защищенных СК с использованием средств виртуализации.

ГЛАВА 3. МОДЕЛИ БЕЗОПАСНОЙ ОБРАБОТКИ ИНФОРМАЦИИ С УЧЕТОМ ОСОБЕННОСТЕЙ СТАЦИОНАРНЫХ И БОРТОВЫХ СУПЕРКОМПЬЮТЕРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

3.1 Базовые характеристики современных суперкомпьютеров и проблемы их достижения.

В зарубежных странах с сильно развитыми отраслями суперкомпьютерных технологий наряду с этим подходом ведутся проекты инновационного характера, связанные с фундаментальным решением проблемы создания более универсальных и эффективных экзафлопсных суперкомпьютеров на базе создаваемых принципиально новых технологий. Сложность этого из-за обозначившихся физических ограничений и фантастически масштабных требований приложений настолько велика, что важнейшей стратегией и здесь считается специализация таких суперкомпьютеров на решение того или иного типа задач, либо задач из выделенных областей приложений.

Под производительностью экзафлопсного суперкомпьютера часто понимается пиковая производительность в один экзафлопс, т.е. предельная суммарная теоретически возможная производительность арифметических функциональных устройств процессоров суперкомпьютера.

Еще одна условность в трактовке понятия производительности – тип операций, который по исторически сложившимся причинам выбирается как операция над вещественным числом, представленном в 64-х разрядном машинном формате с плавающей точкой. Это исторически сложившаяся условность, некоторый ориентир.

Для современных приложений важны, например, еще операции над числами такого типа, но большей разрядности, а также над числами с фиксированной точкой (целыми) или логическими кодами.

Повышение значения характеристик памяти и сети суперкомпьютеров связано и с общей тенденцией повышения год от года важности работы с большими наборами и потоками данных, а также внедрения нечисловых алгоритмов в традиционные области, где ранее преобладали научно-технические расчеты.

Это объясняется требованиями к увеличению масштабов и точностей расчетов, сложностью их верификации и сопоставления с экспериментом, сложностью понимания результатов.

Рассматриваемые далее экзафлопсные проекты следует оценивать с учетом целей достижения в них тех или иных из перечисленных выше шести характеристик, а также специфики области приложения, для которой создается суперкомпьютер экзафлопсного уровня.

Ожидаемые характеристики экзафлопсного суперкомпьютера общего назначения образца 2022 года приведены в таблице 1-2 в сравнении с характеристиками лучшего петафлопсного суперкомпьютера 2010 года. Рассмотрим характеристики из таблицы 3.1.

Прежде всего, напомним, что в этой таблице приведены характеристики экзафлопсного суперкомпьютера – “первенца”, суперкомпьютера первого этапа, в нем решается проблема наиболее быстрого выхода на новый уровень производительности.

Таблица 3.1 – Ожидаемые характеристики

Системная характеристика	Суперкомпьютеры 2010 и 2019года		Соотношение 2010 & 2022
	2010 (петафлопсный)	2021-2022 (эксафлопсный)	
Мощность потребления	6 MW	~ 20 MW (цель)	O(1) – O(10)
Энергетическая эффективность	0.5 Gflop/s/W	50 Gflop/s/W	O(100)
Объем памяти системы	0.3 PB	32-64 PB	O(100)
Производительность узла	125 GF	от 1.2 до 15 TF	O(10) - O(100)
Пропускная способность памяти узла	25 GB/s	2 – 4 TB/s	O(100)
Параллелизм узла	12	от O(1k) до O(10k)	O(100) - O(1000)
Общая пропускная способность сетевых интерфейсов узла	3.5 GB/s	200 – 400 GB/s (1/4 или 1/8 от пропускной способности памяти узла)	O(100)
Размер системы (количество узлов)	18700	O(100000) или O(10 ⁶)	O(10) - O(100)
Общий уровень параллелизма	225000	O(10 ⁹) [многоядерность + O(10) или O(100) на ядро для сокрытия задержек операций с памятью и сетью]	O(10000)
Емкость подсистемы внешней памяти	15 PB	500 – 1000 PB (по крайней мере в 10 раз больше объема системной памяти)	O(10) - O(100)

Пиковая производительность увеличивается в $O(1000)$ раз по сравнению с петафлопсным уровнем 2010 года, а объем системной (оперативной) памяти в $O(100)$ раз. Обычный, проверенный опытом инженерный баланс, – “флопс на байт”. В данном случае пиковая производительность оказывается больше, чем предписывает желаемый баланс. Это связано с ограничениями возможностей разработки подсистемы памяти.

Заметим, что при использовании этих подходов рост производительности в $O(10)$ – $O(100)$ раз обеспечивается ростом параллелизма узла в $O(100)$ – $O(1000)$ раз, а в масштабах всей системы – в $O(1000)$ – $O(10000)$ раз, это самый сильный прирост из всех характеристик.

Параллелизм узла определяется количеством процессорных ядер, умноженным на количество аппаратно поддерживаемых тредов в отдельном ядре. В петафлопсном суперкомпьютере 2010 года параллелизм ядра – 12, в узле было два 6-ядерных микропроцессора с одним тредом в ядре. В транспетафлопсных (десятки петафлопс) суперкомпьютерах 2012 - 2015 гг. параллелизм узла около сотни, а в узлах с графическими ускорителями – несколько сотен. В экзафлопсном суперкомпьютере параллелизм узла ожидается на уровне от тысячи до десяти тысяч, при этом предполагается, что будут и эффективные средства взаимодействия и синхронизации такого множества ресурсов.

Построение усовершенствованных инновационных образцов экзафлопсных суперкомпьютеров потребует новых вариантов использования огромного параллелизма ядер и тредов, их взаимодействия и синхронизации за счет радикального изменения архитектурно-программного облика этих суперкомпьютеров.

Экзафлопсные суперкомпьютерные комплексы будут очень сильно отличаться от традиционных архитектур с распределенной памятью и передачей сообщений. Потому что они будут:

- поддерживать глобальное адресное пространство (Global Address Space - GAS), а не распределенную память,
- системами многопоточной, а не многопроцессорной архитектуры,
- иметь динамические системы управления ресурсами, а не опираться только на статическое распределение, выполненное во время компиляции.

Для защиты информационных и программных ресурсов АС необходимо применять: Средства антивирусной защиты; Межсетевые экраны; Средства обнаружения загрузки; СКЗИ (для шифрования каналов связи – проводных и беспроводных); Средства доверенной загрузки, контроля машинописных носителей; Средства контроля целостности данных и создания изолированной программной среды. Дополнительные затраты, связанные с разворачиванием инфраструктуры в интересах национальной безопасности и реализации программы импортозамещения:

1 вариант: только затраты на создание распределенной защищенной мультитидоменной инфраструктуры с использованием ЗГОС «Фебос» или ЗОС «Арамид».

В этом случае мы используем уже имеющиеся аппаратные модули, представленные на рынке: __микросхемы промышленного применения КОМДИВ (серия 1890ВМ6Я, серия 1890ВМ7Я), специализированные СБИС КОМДИВ (серия 1900ВМ2Т, серия 1907ВМ044, серия 1907ВМ028, серия 1907ВМ038), специализированные СБИС (на архитектуре ЕС8430 «Ангара», серия К1927ВВ1Я), вычислительные блоки (юниты) «Байкал-М», ARM 7/8, «Эльбрус» 4С/8С/16С.

2 вариант: затраты на создание распределенной защищенной среды и дополнительные расходы на промышленное производство вычислительного устройства на базе отечественного массово-мультитредового вычислителя J7 для перспективных образцов вычислительной техники.

Гипервизор обеспечивает безопасность и доверие, в том числе взаимодействие с аппаратными функциями защиты, функционирующие под его контролем прикладные ОС – удобную и привычную для пользователя среду, и совместимость с распространенными приложениями. Гипервизор полностью контролирует работу как аппаратуры вычислительной системы, так и виртуальных машин, причем может не только динамически влиять на их вычислительную среду и манипулировать множеством доступных для них ресурсов, но даже активно вмешиваться в реализуемый ими вычислительный процесс, предотвращая атаки с использованием уязвимостей ОС и приложений. Гипервизор полностью прозрачен для ОС и ее приложений и не только контролирует вычислительные процессы в виртуальных машинах, но и обеспечивает выполнение правил политики безопасности при осуществлении доступа к информационным и вычислительным ресурсам.

Для этого в ЗГОС «Фебос» реализована поддержка контролируемого доступа приложений, находящихся в одной виртуальной среде, к ресурсам, находящимся в другой виртуальной машине. Таким образом, безопасность ЗГОС «Фебос» определяется только свойствами гипервизора и не зависит от свойств прикладных ОС и наличия в них уязвимостей. «Фебос» является микроядерной, многопользовательской, многозадачной, многопоточной операционной системой и обеспечивает тотальный контроль всех взаимодействий в системе. В состав ЗГОС «Фебос» входят (см. рисунок 3.1):

- доверенный гипервизор, позволяющий запускать любую операционную систему в безопасной и устойчивой к кибератакам виртуальной среде;
- система контроля доступа к информационным ресурсам из другой виртуальной среды;
- система управления виртуальными средами.

Для каждого заказчика можно создать готовый набор шаблонов виртуальной среды с набором нужного предустановленного ПО под контролем защищенного гипервизора. Вся работа пользователей, программистов и администраторов будет осуществляться в доверенной изолированной программной среде и будет работать на разных аппаратных платформах (с использованием трансляторов, эмуляторов, аппаратной виртуализации или контейнирования).



Рисунок 3.1 – Пример набора готовых шаблонов ВМ, адаптированных к требованиям заказчика

ЗОС «Арамид» – это совокупность специализированной операционной системы на базе открытой программной платформы Linux и встроенных СЗИ от несанкционированного доступа, прошедшая в 2020 г. сертификацию под требования ФСТЭК РФ. Разработанный программный комплекс соответствует «Требованиям безопасности информации к операционным системам», «Профилю защиты операционных систем типа «А» (ОС общего назначения) 2-го класса защиты, а также «Требованиям по безопасности информации, устанавливающие уровни доверия к средствам технической защиты информации» – по 2-му уровню доверия.

Мы экономим на дополнительных средствах создания доверенной среды и учитываем все требования регулятора еще на этапе разработки и внедрения системы, а не просто устанавливаем дополнительного аппаратных модулей и ПО «под ключ». Мы работаем в многоуровневой «песочнице» и можем создавать моделирующие расчетные стенды (тестовую среду и продуктивную зону) для промышленного внедрения всей линейки существующих и разрабатываемых процессорных вычислителей, что позволит бесперебойно поставлять и поддерживать различные кластерные вычислительные системы, бортовые супер-ЭВМ, суперкомпьютеры нового поколения для «обкатки» решений.

Для разного класса задач адаптируем вычислитель, наборы шаблонов и конфигураций. Пока идет разработки и тестирование новых образцов вычислительных многопроцессорных модулей, работают образцы предыдущего поколения. После успешных проверок вводятся новый вычислитель и набор ПО. Внедрение принципиально нового технологического решения позволит создать изолированную среду исполнения программ в виде 8-уровневой «песочницы» с реализацией контролирующих механизмов на уровне компонентов гипервизоров и других низкоуровневых модулей. Таким образом, мы решаем важную научно-техническую проблему в области создания средств защиты информации для АС «нового поколения», обрабатывающих данные разной степени конфиденциальности. Например, для повышения отказоустойчивости и защищенности стационарных супер-ЭВМ, элементно-конструкторской базой которых являются микросхемы промышленного применения КОМДИВ (серия 1890ВМ6Я, серия 1890ВМ7Я), мобильных и бортовых высокопроизводительных систем на базе специализированных СБИС КОМДИВ (серия 1900ВМ2Т, серия 1907ВМ044, серия 1907ВМ028, серия 1907ВМ038), вычислительных кластеров ЕС1740.000х на основе специализированных СБИС (на

архитектуре ЕС8430 «Ангара», серия К1927BB1Я), мобильных вычислительных системах на базе блоков «Байкал-М», ARM 7/8 целесообразно реализовать следующие аппаратно-программные меры по защите информации:

1. Внедрить механизм защиты по контролю контекстно-зависимых операций. Тип контекста определяет режим выполнения операций чтения/записи: работа с сегментами памяти приложений, работа с системной областью памяти устройств, неинициализированные сегменты памяти для подготовки загрузки приложений, ошибки записи в буфер (неверные указатели и переменные окружения, переполнение, утечка памяти и т.д.);
2. Реализовать расширенный контроль конфигураций в соответствии с политиками безопасности. Изменение контекста выполняемых операций на всех уровнях иерархии становится детерминированным, а не стохастическим. События, приводящие к изменению тега запроса, связаны с повышением уровня привилегий, обращением к привилегированной области памяти, с изменением статуса прохождения запроса, наличием переходного процесса и запуском подпроцессов при прохождении межузловой трассы, модификацией значений хеш-функций (сверток) для правильной трассы обработки запроса;
3. Реализовать поддержку мультитредовых функционально - вычислительных блоков с «легкими» и «тяжелыми» ядрами, адаптированная к контексту и типу данных с использованием синхронных и асинхронных тредов с высокими показателями производительности. Для обычных счетных задач применяются процессоры с суперскалярными «тяжелыми» ядрами, а для вычислительных задач, когда доля вычислительных операций высока (локализация и интенсивность работы с данными может быть разной),

используются мультитредовые счетные матрицы из “легких” вычислительных блоков (юнитов), установленных в специализированных СБИС;

4. Ввести в эксплуатацию встроенный аппаратный модуль с функциями верификатора команд на уровне привилегий IPL (инструкций процессора) для кодирования/декодирования команд, взаимодействия с сетевыми сопроцессорами и контроллерами памяти;
5. С целью повышения уровня защищенности на всех уровнях иерархии (конвейера) обработки запросов для высокопроизводительных комплексов необходимо реализовать 6 интервалов фильтрации запросов и функции контроля тайминга выполнения команд компонентами гипервизора и контроллерами памяти;
6. Предлагаемое решение учитывает как эволюционную динамику развития вычислительной техники и технологии программирования, так и революционную (инновационную) составляющую.

3.2 Модель угроз целостности среды исполнения процессов суперкомпьютеров с реализацией новой парадигмы написания правил политики безопасности и новых сущностей – «субъект», «объект» и «дескриптор оценки безопасности состояний»

Выделим класс угроз, связанный с неоднозначностью состояний выполнения прикладных и системных программ. Суперкомпьютер обладает высокой асинхронностью выполнения программ, его отличает многоуровневое распараллеливание вычислительных процессов, что обуславливает появление новых угроз.

Таблица 3.2 – Перечень угроз, влияющих на безопасность СК

Название	Возможные последствия
Нестандартная обработка запросов за счет неоднозначности состояний выполнения прикладных и системных программ СК	Получение несанкционированного доступа к ресурсам СК
Модификация компонентов гостевых ОС СК	Получение несанкционированного доступа к данным пользователя, расположенным на разных виртуальных машинах Повреждение компонентов гостевых ОС
Модификация компонентов управляющих ОС СК	Повреждение компонентов управляющих ОС и распространение вредоносного ПО

Неоднозначность обусловлена тем, что выполнение программ на уровне объектов гостевых и управляющих ОС не эквивалентно изменению состояний компонентов гипервизора, обрабатывающих запросы пользователей на выделение вычислительных ресурсов СК при взаимодействии с физическим оборудованием (см. таблицу 3.2).

В случае успешной атаки суперкомпьютер сам становится источником угроз и может оперативно спланировать, и провести массовую атаку, быстро адаптироваться по результатам ее проведения и провести серию последующих атак. Для этих состояний проблема безопасности – это проблема корректности получаемых результатов. Чтобы достичь безопасности надо блокировать «опасности».

Блокировка требует синхронизации состояний большого количества высокопараллельных процессов, а для этого надо ввести еще один «слой» контроля в виде дополнительного домена защиты (см. рисунок 3.2).

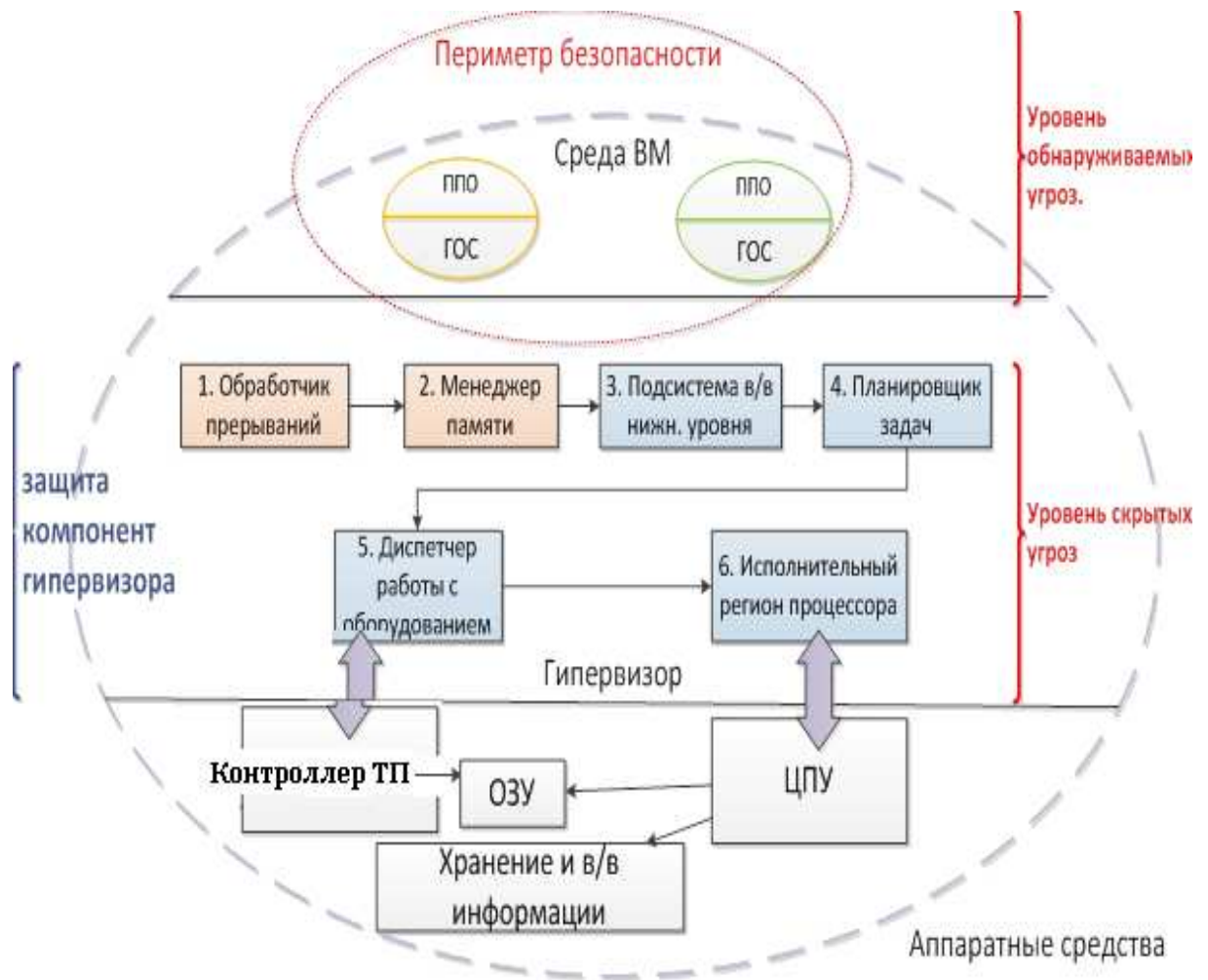


Рисунок 3.2 – Структурная схема и скрытые угрозы

С этой целью введем дополнительный уровень защиты, который позволяет контролировать выполнение запросов на всех уровнях обработки без значительных потерь по производительности и энергоемкости.

В таблице 3.3 приведены типы нарушителей.

Таблица 3.3 – Виды нарушителей

Виды нарушителя	Типы нарушителя	Возможные цели (мотивация) реализации угроз безопасности информации
Внешние субъекты (физические лица) и Конкурирующие организации	Внешний	Идеологические мотивы. Причинение имущественного ущерба путем мошенничества или иным преступным путем. Любопытство или желание самореализации (подтверждение статуса). Выявление уязвимостей с целью их дальнейшей продажи и получения финансовой выгоды
Разработчики, производители, поставщики программных, технических и программно-технических средств	Внешний	Внедрение дополнительных функциональных возможностей в программное обеспечение или программно-технические средства на этапе разработки. Причинение имущественного ущерба путем обмана или злоупотребления доверием. Непреднамеренные, неосторожные или неквалифицированные действия
Лица, привлекаемые для установки, наладки, монтажа, пусконаладочных и иных видов работ	Внутренний	Причинение имущественного ущерба путем обмана или злоупотребления доверием. Непреднамеренные, неосторожные или неквалифицированные действия
Лица, обеспечивающие функционирование информационных систем или обслуживающие инфраструктуру оператора	Внутренний	Причинение имущественного ущерба путем обмана или злоупотребления доверием. Непреднамеренные, неосторожные или неквалифицированные действия
Пользователи информационной системы	Внутренний	Причинение имущественного ущерба путем мошенничества или иным преступным путем. Любопытство или желание самореализации (подтверждение статуса). Месть за ранее совершенные действия. Непреднамеренные, неосторожные или неквалифицированные действия
Администраторы информационной системы и администраторы безопасности	Внутренний	Причинение имущественного ущерба путем мошенничества или иным преступным путем. Любопытство или желание самореализации (подтверждение статуса). Месть за ранее совершенные действия. Выявление уязвимостей с целью их дальнейшей продажи и получения финансовой выгоды. Непреднамеренные, неосторожные или неквалифицированные действия
Бывшие работники (пользователи)	Внешний	Причинение имущественного ущерба путем мошенничества или иным преступным путем. Месть за ранее совершенные действия

Взаимодействие процессов пользователя происходит с компонентами гипервизора и контроллером транзакционной памяти, который создает изолированную среду исполнения.

Компоненты гипервизора контролирует обращения к объектам файловой системы и стекам сетевых протоколов, осуществляют контроль последовательности выполнения запросов на выделение вычислительных ресурсов на всех уровнях иерархии (S1– S8). Каждый уровень иерархии привязан к определенному уровню защиты. Каждый запрос представляет собой набор операций. Набор операций – это операции чтения, записи, выполнения, удаления и т.д. Каждая операция, в свою очередь, на каждом уровне иерархии выполнения транслируется гипервизором в набор директив, выполняемых процессором.

Механизм трансляции каждой операции представляет собой сборку команд непосредственного сценария выполнения исходящих директив с использованием тегированных полей с привязкой к конкретному уровню защиты с учетом уровня иерархии (привилегий) обрабатываемых запросов.

Мы предлагаем использовать модель угроз целостности среды выполнения процессов СК [17; 21; 26; 52], в которой вместо классической связки «субъект, объект, предикат» реализована новая парадигма написания правил политики безопасности и заданы новые сущности – «субъект», «объект» и «дескриптор оценки безопасности состояний»: для каждой *i*-ой угрозы вычисляется значение функции оценки безопасности состояний, аргументы которой заданы в виде конъюнкции предикатов из восьми логических переменных, субъектами являются гостевые и управляющие ОС, а объектами - компоненты гипервизора и контроллера транзакционной памяти.

Целью оценки возможностей нарушителей по реализации угроз безопасности информации СК является формирование предположения о типах, видах нарушителей, которые могут реализовать угрозы безопасности информации в информационной системе с заданными структурно-функциональными характеристиками и особенностями функционирования,

а также этих нарушителей и возможных способах реализации угроз безопасности информации. Результаты оценки возможностей нарушителей включаются в модель нарушителя, которая является составной частью модели угроз безопасности информации и содержит:

- типы, виды и потенциал нарушителей, которые могут обеспечить реализацию угроз безопасности информации;
- цели, которые могут преследовать нарушители каждого вида при реализации угроз безопасности информации;
- возможные способы реализации угроз безопасности информации.

Типы нарушителей определяются по результатам анализа прав доступа субъектов к информации или к компонентам информационной системы, а также анализа возможностей нарушителей исходя из структурно-функциональных характеристик и особенностей функционирования информационной системы.

С учетом наличия прав доступа и возможностей по доступу к информации и (или) к информационным и программным компонентам СК нарушители подразделяются на два типа:

- внешние нарушители, т.е. лица, не имеющие права доступа к информационной системе, ее отдельным компонентам и реализующие угрозы безопасности информации из-за границ информационной системы;
- внутренние нарушители, т.е. лица, имеющие право постоянного или разового доступа к информационной системе, ее отдельным компонентам.

Наибольшими возможностями по реализации угроз безопасности обладают внутренние нарушители.

При оценке возможностей внутренних нарушителей необходимо учитывать принимаемые оператором организационные меры по допуску субъектов к работе в информационной системе. Возможности внутреннего нарушителя существенным образом зависят от установленного порядка допуска физических лиц к информационной системе и ее компонентам, а также мер по контролю над доступом и работой этих лиц.

Внешнего нарушителя необходимо рассматривать в качестве актуального во всех случаях, когда имеются подключения информационной системы к внешним информационно-телекоммуникационным сетям и (или) имеются линии связи, выходящие за пределы контролируемой зоны, используемые для иных подключений. Угрозы безопасности СК могут быть реализованы за счет утечки данных по техническим каналам (технические каналы утечки информации, обрабатываемой в технических средствах, технические каналы перехвата информации при ее передаче по каналам связи) либо за счет несанкционированного доступа с использованием соответствующего программного обеспечения.

Единственными совсем незащищенными сегментами СК остаются вычислительные узлы (блоки) и инфраструктура взаимодействия вычислительных узлов с клиентскими и сервисными хостами, поскольку с учетом эксплуатационных и технических требований высокой производительности обработки, передачи данных по сети, асинхронности и параллельности выполняемых задач на них не устанавливается дополнительное ПО в виде СЗИ.

Поскольку существующие программно-технические решения ориентированы на обычные серверы и не учитывают архитектуры и наборы конфигураций СК, не оптимизированы и не адаптированы к типу входных данных, то использование комплексных средств безопасности в виде

средств антивирусной защиты, программно-аппаратных средств доверенной загрузки и средств защиты от НСД, межсетевых экранов, DLP-систем и др. значительно снижает быстродействие системы. Появляются также значительные временные задержки при выполнении высоконагруженных операций, что может привести к десинхронизации работы вычислительных комплексов, блокированию, повреждению и уничтожению данных.

Для создания подсистемы обеспечения безопасности суперкомпьютерных платформ производители оборудования для суперкомпьютерных платформ реализовали встроенную полнофункциональную поддержку технологии виртуализации (Xen, KVM, LXC) для разного набора микропроцессоров и микроконтроллеров, доменов защиты с использованием интерфейса UEFI и с управлением режимов работы ЦПУ.

Других базовых (штатных) функций безопасности у аппаратных модулей и сборочных конструктивов, из которых собираются компоненты СК, просто нет. Следовательно, только технология виртуализации позволяет создать многоуровневую мультидоменную “песочницу” и повысить гибкость использования и функционирования оборудования.

Также целесообразно оптимизировать развиваемую производительность и энергоэффективность: можно реализовать механизмы защиты узлов СК с минимальными потерями производительности и временными задержками при выполнении высоконагруженных задач, задавая нужную конфигурацию в виде набора доменов защиты с учетом требований пользователей (заказчиков).

Программное обеспечение специализированных мини-сборок ОС Linux для вычислительных хостов суперкомпьютеров позволяет оперативно создать извне во время выполнения произвольной пользовательской программы контрольную точку с возможностью возобновления работы программы с этой точки, в том числе, в другом экземпляре операционной

системы (функция “живой” миграции). Более того, благодаря использованию технологии аппаратной виртуализации и технологии i-RAM (интеллектуальная дисковая память) мы получаем рекордные показатели быстродействия, отказоустойчивости и энергоэффективности. Сверхбыстрые диски в оперативной памяти, работающие под управлением контроллеров аппаратной транзакционной памяти, позволяют стабильно и с рекордно высокими показателями производительности обрабатывать огромные массивы данных (порядка 6 ГБ/с вместо 500 МБ/с для самых быстрых SSD-дисков).

Среди основных преимуществ i-RAM для высокопроизводительных вычислительных комплексов над традиционными виртуальными RAM-дисками, организованными программно в оперативной памяти компьютера, можно указать: поддержка RAID-массивов, низкая загрузка центрального процессора, сохранность данных и самих дисков после выключения питания, отсутствие специального ПО для организации и использования, производителями уже реализованы предустановленные гибкие интерфейсы магистральных шин и UEFI BIOS для конфигурирования работы оборудования (диагностика PCI – слотов DRAM, интеграция с контроллерами многосокетных материнских плат, управление ACPI – модулями питания и системой охлаждения, управление контроллерами транзакционной памяти и процессорными функциональными устройствами).

При использовании данной технологии скорость загрузки ОС и запуска сервисов (приложений), монтирования в оперативной памяти резервного образа в случае сбоя имеет наилучшие показатели, достижимые с использованием современных СВТ (порядка 5-6 секунд).

С учетом того, что современные суперкомпьютерные центры представляют собой либо автономный стационарный объект (без подключения к внешним сетям), либо распределенный стационарный

объект (с подключением к внешним сетям, но в этом случае используются VPN-технологии на примере криптомаршрутизаторов и шлюзов безопасности), в любом случае, подключения к общедоступным сетям нет. Доступ к сервисным узлам имеют только внутренние сотрудники (администраторы и обслуживающий персонал). Клиентские узлы не подключены к сетям администрирования. На сетевом коммуникационном оборудовании стационарных суперкомпьютеров, рабочих станциях клиентов (операторов) и сервисных узлах суперкомпьютерного центра обработки данных (под управлением администраторов информационной системы и администраторы безопасности, обслуживающего технического персонала) устанавливаются средства криптографической защиты информации, средства антивирусной защиты, программно-аппаратные средства доверенной загрузки и средства защиты от НСД, МЭ, DLP-системы и т.д. Поэтому на сервисных и клиентских хостах СК уровень защищенности максимально высокий. Бортовые суперкомпьютеры, устанавливаемые на летательных аппаратах, бывают двух типов: автономные одноплатные портативные бортовых вычислителей без сетевых адаптеров (мини-сборка) и многоплатные бортовые вычислители, объединенные в локальную сеть и с поддержкой RadioEthernet, Bluetooth, Wi-Fi. Для защиты передаваемых данных используются скремблеры и шифраторы.

Бортовые супер-ЭВМ размещаются в специальных шкафах с замками. Несущие конструкции бокса в защищенном исполнении, используются кабели с экранированием.

Угрозы анализа сетевого трафика внешних сетей, угрозы анализа сетевого трафика внутренней сети, угрозы выявления паролей, удаленный запуск приложений и т.п. не являются актуальными.

Угрозы внедрения ложного объекта сети, угрозы навязывания ложного маршрута представляют собой низкий уровень опасности, поскольку все телекоммуникационное оборудование размещается на защищенном аттестованном объекте или на борту летательного аппарата. В стационарных суперкомпьютерных центрах реализована многоуровневая (эшелонированная) инженерно-техническая защита.

При проведении установки, наладки, монтажа, пусконаладочных и иных видов работ, при изменении конфигурации или при обновлении и установке дополнительных программных, технических и программно-технических средств высока угроза внедрения вредоносного ПО локально или по сети. В результате выполнения непреднамеренных или умышленных действий со стороны пользователей, администраторов, обслуживающего технического персонала также высока вероятность вывода из строя либо простоя в функционировании, что недопустимо для СК при выполнении разного класса инженерно-расчетных высоконагруженных задач.

Таким образом, актуальными угрозами безопасности для СК являются: угрозы НСД до загрузки ОС и во время загрузки ОС; угрозы НСД после загрузки ОС; отказ в обслуживании; угрозы внедрения вредоносных программ; угрозы внедрения по сети вредоносных программ (см. таблицу 3.4).

Таблица 3.4 – Возможность реализации угрозы и оценка ущерба

Угроза безопасности данных	Возможность реализации угрозы	Оценка ущерба
угрозы НСД до загрузки ОС и во время загрузки ОС	высокая	Критичный уровень. Если сбой функционирования инфраструктуры (тысячи узлов) происходит на этапе или во время загрузки ОС вычислительных, сервисных и клиентских хостов, то нарушается вся работа СК. Для современного суперкомпьютерного центра ущерб от простоя оценивается приблизительно 10 млн. – 100 млн. рублей
угрозы НСД после загрузки ОС	высокая	Если сбой функционирования инфраструктуры происходит после загрузки ОС вычислительных, сервисных и клиентских хостов, то нарушается работа лишь отдельных сегментов. Возможен откат и восстановление системы в режиме реального времени (время на восстановление на порядок меньше). Для современного суперкомпьютерного центра ущерб от простоя оценивается приблизительно 100 тыс. – 10 млн. рублей
угрозы анализа сетевого трафика внешних сетей	низкая	Не актуальная угроза
угрозы анализа сетевого трафика внутренней сети	низкая	Не актуальная угроза
угрозы выявления паролей	низкая	Не актуальная угроза
угрозы получения НСД путем подмены доверенного объекта	низкая	Не актуальная угроза
угрозы внедрения ложного объекта сети	низкая	Не актуальная угроза
угрозы навязывания ложного маршрута	средняя	Не актуальная угроза

Продолжение таблицы 3.4

угрозы отказа в обслуживании в результате внешних дестабилизирующих воздействий по каналам коммуникации и сетям питания, некорректного выполнения операций, неправильных действий оператора или администратора и т.п.	высокая	Критичный уровень. Если сбой функционирования инфраструктуры (тысячи узлов) происходит в результате внешних дестабилизирующих воздействий, то возможно физическое повреждение компонентов СК, необратимый их вывод из строя. Для современного суперкомпьютерного центра некомпенсируемый ущерб от простоя оценивается приблизительно Более 100 млн. рублей
угрозы удаленного запуска приложений	низкая	Не актуальная угроза
угрозы внедрения вредоносных программ в гостевых и хостовых ОС	высокая	Сбой функционирования инфраструктуры происходит лишь в отдельных сегментах инфраструктуры без изменений режимов работы аппаратуры. Для современного суперкомпьютерного центра ущерб от простоя оценивается приблизительно 100 тыс. – 10 млн. руб
угрозы внедрения по сети вредоносных программ гостевых и хостовых ОС	высокая	Сбой функционирования инфраструктуры происходит после загрузки ОС вычислительных, сервисных и клиентских хостов, то нарушается работа лишь отдельных сегментов 100 тыс. – 10 млн. руб

Модель угроз учитывает новый класс уязвимостей, связанных с высокой асинхронностью процессов СК и возможностью выполнения ложных транзакций в результате дестабилизирующих внешних воздействий и случайных сбоев аппаратуры. Необходимо реализовать доверенную среду выполнения программ на недоверенном оборудовании.

Требования политики безопасности формулируются в терминах, которые задают последовательность обработки операций и ограничений на возможность повышения привилегий процессов.

При этом на каждом уровне модели операций S1-S8 ведется протоколирование событий и результатов, включая такие параметры, как Time – время и Result – результат выполнения операций. Ограничение на повышение привилегий и контроль переходов при изменении контекста операций задаются значениями функции оценки безопасности состояний F_i , аргументы которой заданы в виде конъюнкции предикатов из восьми логических переменных $x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8$, а контроль выполнения процессов, порождаемых субъектами доступа, реализуется на основе принципа наименьших привилегий.

Гипервизор с помощью функции виртуализации управляет доступностью защищаемых ресурсов в прикладных ОС и с помощью встроенных средств защиты контролирует операции авторизованных субъектов над защищаемыми ресурсами в соответствии с политикой безопасности.

Высокопроизводительные системы в ряде случаев применяются в приложениях, при выполнении которых предъявляются повышенные требования к обеспечению безопасности, удовлетворяющая требованиям регуляторов и адаптированная для работы на многопроцессорных системах [127; 128].

Тогда модель угроз целостности среды выполнения программ для высокопроизводительных комплексов и автоматизированных систем, использующих элементы суперкомпьютерных технологий, в части описания “субъект защиты-объект защиты” может быть формализована в виде кортежа:

< Source, Services, Devices, {proc}, Actions, {hv}, {vm}, SecurityRoles >

(3.1)

Source – авторизованные субъекты доступа. $\text{Source} = S^H \cup S^O$, где S^H – множество субъектов гипервизора, а S^O – множество субъектов гостевой и хостовой ОС.

Services – шаблоны политик безопасности.

Devices – аппаратные устройства, управляемые гипервизором и контроллером транзакционной памяти.

{proc} – множество субъектов вредоносного воздействия .

Actions – операции субъекта.

{hv} – множество компонент гипервизора и контроллера транзакционной памяти.

{vm} – множество объектов воздействия.

Security-Roles – механизмы противодействия (идентификации состояний в зависимости от набора кортежа логических переменных функции F_i).

Новизна модели (по сравнению с базовой моделью угроз ФСТЭК):

- учет нового класса угроз, связанных с архитектурной и функциональной спецификой суперкомпьютеров и особенностями технологий аппаратной виртуализации, с десинхронизацией, неоднозначностью описания состояний процессов на разных уровнях иерархии выполнения команд на недоверенном оборудовании;
- учет динамического характера взаимодействия: объекты доступа (например, объекты файловой системы и сетевой подсистемы в результате воздействия вредоносного ПО, внедрения и запуска вредоносного кода) сами становятся инициаторами (субъектами) противоправных действий для проведения последующих атак.

Разработанные в РФЯЦ-ВНИИЭФ и РФЯЦ-ВНИИТФ сертифицированные СЗИ установлены и эксплуатируются сегодня на ряде защищенных комплексов, включающих высоконагруженные параллельные суперкомпьютеры [43]. Однако временная методика по оценке уровня защищенности и модель угроз, разработанные специалистами Росатома (для систем с литерой “С”), не учитывают ряд механизмов.

Новизна модели (по сравнению с моделью угроз Росатома)

- учет контролирующих механизмов как на уровне гипервизоров, так и на уровне контроллеров транзакционной памяти при работе в недоверенной среде;
- 8-уровневая среда функционирования с поддержкой разных режимов эмуляции и виртуализации с учетом потока входных данных;
- интеграция с аппаратной транзакционной памятью;
- введения многоуровневого контроля контекста и тайминга выполняемых операций на всех уровнях иерархии команд современных микропроцессоров;
- в основе реализации средств защиты информации Росатома с учетом модели угроз для объектов КИИ лежат только базовые программно-технические компоненты на базе Scientific Linux для платформы x86_64.
- При этом не учитываются особенности работы процессоров и контроллеров с разными уровнями иерархии команд, специфика конвейера обработки запросов, режимов эмуляции и виртуализации для разной аппаратуры;

- Подавляющее большинство аппаратных средств, используемых для создания супер-ЭВМ, зарубежного производства, либо сборка конструктивов осуществляется по иностранной лицензии на территории РФ, реализуемая же обязанка состоит из недоверенных элементов, импортируемых из-за рубежа;
- Без интеграции с аппаратной транзакционной памятью и введения многоуровневого контроля невозможно создать доверенную среду с использованием широко распространенных коммерческих программно-аппаратных средств.

Сравнение авторской модели угроз с существующими моделями приведено в таблице 3.5. **Видно, что все существующие на сегодняшний день наиболее распространенные модели угроз не учитывают специфики суперкомпьютеров разного класса, ориентированы только на традиционные кластеры и информационные системы (Рабочие станции, ИСПДн, АСУ ТП, ЕИАС, информационно-телекоммуникационные системы, государственные информационные системы и т.д.).**

Таблица 3.5 – Сравнение моделей угроз

Название модели	Уровни угроз безопасности	Оценка уровня защищенности	Объект защиты
Модель угроз NIST & MITRE (вербальная модель с элементами статистической обработки и визуализации выходных данных)	Критичный Высокий Средний Низкий	Балльные оценки экспертов или шкала таблицы показателей	Рабочие станции, мобильные устройства, виртуальная инфраструктура, серверы коммуникационной среды, СУБД, ERP-системы
Модель угроз CyberCorp & CERT (вербальная модель с элементами прогнозирования на основе экстраполяции данных)	Критичный Высокий Средний Низкий	Балльные оценки экспертов или шкала таблицы показателей	Рабочие станции, мобильные устройства, виртуальная инфраструктура, серверы коммуникационной среды, СУБД, ERP-системы
Модель угроз ФСТЭК (вербальная модель с элементами моделирования угроз)	Высокий Базовый повышенный Базовый Низкий	Балльные оценки экспертов или шкала таблицы показателей	Рабочие станции, ИСПДн, АСУ ТП, ЕИАС, информационно-телекоммуникационные системы, государственные информационные системы
Модель угроз Росатома (вербальная модель с элементами моделирования угроз)	Критичный Высокий Средний Низкий	Балльные оценки экспертов или шкала таблицы показателей	Рабочие станции, ИСПДн, АСУ ТП, ЕИАС, информационно-телекоммуникационные системы, государственные информационные системы, мейнфреймы
Авторская модель угроз (формализованная модель)	Максимальный Высокий Средний Базовый повышенный Базовый Низкий	Использование функции оценки безопасности состояний компонентов СК Fi	Промышленные кластеры, средние суперкластеры, суперкомпьютеры стратегического класса, бортовые Супер-ЭВМ

Существующие методики и алгоритмы обнаружения угроз [40; 41; 42] базируются на использовании скрытых марковских моделей процессов с использованием алгоритма Баумана-Уэлча, на основе выборочных центральных моментов ожидания 2-го, 4-го и 8-го порядков, на основе выборочной дисперсии σ , на основе длины вариационного ряда, на основе нейросетевых моделей (с использованием фильтра Калмана и алгоритма машинного обучения Random forest). Они требуют значительных ресурсов для создания обучающих кластерных зон. Традиционные методики и алгоритмы требуют значительных затрат на обучение в виде вычислительных ресурсов и не решают проблему функциональных ограничений (работают только для определенной платформы – например, PE-файлы ОС Windows). Мы же предлагаем алгоритм обнаружения угроз, основанный на использовании набора меток. Возможным сочетаниям меток сопоставляются признаки скрытых угроз и действия, отвечающие требованиям выбранной политики ИБ. Обучающие выборки являются презентативными.

При количестве измерений равным 8, получаемых с датчиков измерений на каждом уровне “песочницы”, показатель погрешности вычислений $e = 0, (1)/8 = 0,0139$, то есть меньше 1,5 процентов. Таким образом, все состояния в системе наблюдаемы и ошибка вычислений менее 1,5 процентов. Следовательно, с учетом квазидетерминированности системы, о которой упоминалось ранее, возможно снижение размерности решаемых задач исследования в виде перехода от сложных скрытых марковских моделей к упрощенной марковской модели, а именно, в виде марковского процесса с дискретными состояниями и непрерывным временем рассмотрим на примере мультиграфа состояний с учетом вероятных угроз атаки.

Угроза атаки создается уязвимостями с соответствующими параметрами — интенсивностями выявления и устранения уязвимостей. Более того, данное решение не зависит от аппаратной платформы и не имеет архитектурных ограничений. Математическое описание марковского процесса с дискретными состояниями и непрерывным временем рассмотрено на примере графа угрозы атаки; угроза атаки создается уязвимостями с соответствующими параметрами — интенсивностями выявления и устранения уязвимостей. На графе представлены четыре возможных состояния: S_0 — исходное состояние системы, S_i — в системе выявлена и не устранена i -ая уязвимость. Применительно к рассматриваемой задаче моделирования интерес представляет набор состояний S_i . Всего в системе 8 уровней иерархии.

Далее можно рассматривать в качестве вероятности возникновения угрозы атаки $P = P_i$, соответственно вероятность готовности к безопасной эксплуатации СК в отношении угрозы атаки $P_{0a} = (1 - P_{ya})$. Значение вероятности P_{0a} рассчитывается по следующей формуле

$$P_{0a} = K_{\Gamma} = \frac{\mu_1\mu_2 + \lambda_1\mu_2 + \lambda_2\mu_1}{(\lambda_1 + \mu_1)(\lambda_2 + \mu_2)}. \quad (3.2)$$

Параметр потока отказов ω определяется (с учетом квазидетерминированности процессов) следующим образом:

$$\omega = \sum_{i \in Q_+} P_i \sum_{j \in Q_-} \lambda_{ij}, \quad (3.3)$$

где Q_+ — множество состояний работоспособности системы, Q_- — множество состояний отказа системы, λ_i — интенсивность перехода из i -го работоспособного состояния, вероятность нахождения системы в котором

равна P_i в j -е неработоспособное состояние. Поток возникновения угрозы вычисляется по формуле:

$$\lambda_a = \omega = P_1\lambda_2 + P_2\lambda_1. \quad (3.4)$$

P_i – вероятность нахождения в S_i состоянии, λ_i – интенсивность переходов.

Остальные характеристики вычисляются следующим образом:

$$T_{0ya} = 1/\lambda_a, \quad \mu_a = \frac{\lambda_a K_r}{1 - K_r} \quad (3.5)$$

K_r – коэффициент оценки безопасной эксплуатации СК;

μ_a – интенсивность устранения атаки.

В процессе эксплуатации информационной системы реальная угроза атаки, в случае если она не будет реализовываться нарушителем, в среднем через интервал времени T_0 (среднее время наработки на отказ) будет многократно возникать. Всего таких интервалов в предложенной модели $N = 8$. $T_{0\text{сум}} = 8 * T_0$. Вероятность проведения успешной атаки в i -ом состоянии $PA_{yan} \approx 0,1$. Вероятность устранения атаки в i -состоянии $PO_{yan} = (1 - PA_{yan}) \approx 0,9$. Благодаря упрощенной модели описания возможно быстро перенести все требования по разработке автоматизированных систем в защищенном исполнении и их сертификации без дополнительной разработки новых стандартов, а перенос уже существующих подходов и критериев для традиционных кластеров и мейнфреймов с минимальными правками применительно к суперкомпьютерам с введением одного показателя – коэффициента оценки безопасной эксплуатации СК.

ВМС-контроллер (контроллер материнской платы) поддерживает и экспортирует гипервизору набор низкоуровневых интерфейсов системной шины для прямого управления режимами работы и контекстом безопасности выполняемых приложений (задач) контроллера аппаратной транзакционной памяти.

Контроллер транзакционной памяти отслеживает и запрещает конкурирующим процессам обращаться к одному участку памяти, распределяет и привязывает обрабатываемые запросы к доменам в соответствии с их уровнем привилегий. Осуществляют также контроль выполнения запросов агенты гипервизора, функционирующие на всех уровнях иерархии и задающий модальные правила ПБ.

Монитор безопасности также контролирует статус выполнения – поле TCU (все возможные состояния явно инициализированы) и поле Context_ID (hash с учетом возможных коллизий).

Однако вот такие признаки, как Context_type, tr, B, S – динамические в виду высокой параллельности и асинхронности доступа процессов СК, явно неинициализированы в ходе переходных состояний, при временном изменении полномочий и контекста выполнения операций меняется сама траектория вычислений, снято ли состояние блокировки или нет, случайные изменения в битах и т.п. Это явно образом не детерминируется, поскольку выполняются теньевые операции самим процессором, недоступные обычному пользователю или системному администратору.

Система динамическая, высокоасинхронная, картина порождения траекторий может размываться по всем доменам. Всегда существует вероятность ложного срабатывания и неверной адресации при изменении контекста. Нужно оценить, какова степень влияния числа доменов, учет бита S и бита B, типа контекста, биты контроля тайминга и состояний команд на всю систему в целом.

Гипервизором осуществляется обработка поступающих запросов. Заголовок пакета каждого запроса представляет собой дополнительные тегированные поля. Поскольку у нас задача вычисления сверки сведена к решению линейных алгебраических уравнений (4.9), то у нас связь линейной зависимости. Введем признаки с точки зрения согласованной вариации. Они будут представлять собой двоичные дроби, степень которой определена номерами битов нужных тегированных полей. Результирующее значение получается еще от умножения на значение доверительной вероятности, полученной в формуле (4.2).

Стабильность, обеспечиваемая кварцевым резонатором, порядка 10^7 . Поэтому минимальный интервал время мониторинга – 1 мкс. Специфика СК такова, что мы работает не со значением параметров 0 или 1, а с 0.11111111 или 0.88888889. Необходимо учитывать нечеткие значения вплоть до 8-го знака после запятой.

$$Z2(tr) = 0.5 + 0.25 = 0.75$$

$$Z2(Context_type) = 0.109375$$

$$Z3(B) = 0.125$$

$$Z4(S) = 0.976563$$

Тогда признаки с учетом доверительной вероятности, необходимые для анализа взаимной корреляции, следующие:

$$x1 = 0.75 * 0.88888889 = 0.66666666$$

$$x2 = 0.109375 * 0.88888889 = 0.09722222$$

$$y1 = 0.125 * 0.88888889 = 0.11111111$$

$$y2 = 0.976563 * 0.88888889 = 0.86805559$$

Теперь оценим взаимную корреляцию между $x1$, $x2$ и $y1$, $y2$.

Связь признаков проявляется в их согласованной вариации, при этом одни признаки выступают как факторные, а другие - как результативные.

Для оценки параметров уравнений регрессии наиболее часто используется метод наименьших квадратов (МНК), суть которого заключается в следующем требовании: искомые теоретические значения результативного признака должны быть такими, при которых бы обеспечивалась минимальная сумма квадратов их отклонений от эмпирических (фактических) значений, т.е.

$$S = \sum (y - \bar{y}_x)^2 \rightarrow \min \quad (3.6)$$

Теснота связи при линейной зависимости измеряется с помощью линейного коэффициента корреляции, который рассчитывается по одной из формул:

$$r = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sqrt{[\sum (x - \bar{x})^2] \cdot [\sum (y - \bar{y})^2]}} \quad (3.7)$$

Для линейной зависимости теоретическое корреляционное отношение тождественно линейному коэффициенту корреляции, т.е. $z = |r|$.

Таблица 3.6 – Характер связи и значение

Характер связи	Значение
$z = 0$	Отсутствует
$0 < z < 0,2$	Очень слабая
$0,2 < z < 0,3$	Слабая
$0,3 < z < 0,5$	Умеренная
$0,5 < z < 0,7$	Заметная
$0,7 < z < 0,9$	Сильная
$0,9 < z < 1$	Весьма сильная

В ходе генерации внешних дестабилизирующих воздействий, регистрировались значения нужных переменных в буфере запросов. Замеры проводились 40 раз. 10 раундов. Выбирались взвешенные значения полученных данных в каждом раунде.

Полученные результаты подвергались регрессионному анализу по четырем признакам x_1 , x_2 и y_1 , y_2 и приведены в таблице 3.7.

Таблица 3.7 – Показатели взаимной корреляции

Параметры корреляции	Результативные значения с наибольшим отклонением								
X1	0.66666766	0.66666771	0.66666765	0.6666674333	0.6666674333	0.6666674333	0.6666674333	0.6666674333	0.6666674333
X2	0.09722222	0.09622222	0.09702222	0.09722225	0.09722442	0.08992222	0.09722219	0.09690155	0.09810123
Y1	0.11111122	0.12111111	0.11234111	0.11001111	0.10111131	0.11111141	0.11211111	0.10011111	0.10001110
Y2	0.86805459	0.86705559	0.86905559	0.91000559	0.86815559	0.87815559	0.88805559	0.86805559	0.88805549
R (x1y1)	0,651 – заметная связь								
R (x2y1)	0.122 – слабая связь								
R (x1y2)	0.091 – практически отсутствует								
R (x2y2)	0.737 – сильная связь								

Данные подтверждают, что есть сильная связь между изменением типа контекста и признаком системности, заметная связь между снятием блокировки и изменением переходного процесса. При этом контекст и блокировка – почти ничем не связаны. Конфигурация контекста выполнения и временные диаграммы состояний СК в виде блокировок – независимые сущности. Случайные или группированные ошибки в полях Context_type или tr приводят к изменению только бита S или B, соответственно.

Степень взаимной корреляции высока между логическими переменными (Context_type, S – последовательность из 3 бит) и (tr, B – последовательность из 3 бит). Остальные переменные не коррелируют.

Даже если возникают случайные изменения Ord (1 бит) и модификация тегированных полей Status (размер 3 бита), dom_id (размер 2 бита).

Структура передаваемых пакетов в нашей модели безопасных операций при работе на недоверенном оборудовании следующая:

Opcode опкод	Context_ ID хеш	CF Поле контр.-ных битов сверточных кодов	Dom_id Идентиф- икатор домена	S Признак систем.	Ord Признак вложенн- ости	Context_type признак контекста операций	Status статус выпол. запроса	B признак блокир.	Tr признак перехода
-----------------	-----------------------	--	--	-------------------------	------------------------------------	--	---------------------------------------	-------------------------	---------------------------

Рисунок 3.3 – Вид тегированной структуры пакета запросов

На основе всего описанного выше делаем вывод о том, что при декомпозиции информационных процессов взаимодействия в виде 8-уровневой иерархической структуры на основе предложенной модели безопасных операций появляется «взаимно-однозначное соответствие» между процессами выполнения программ на СК, изменениями состояний гипервизора и аппаратуры СК. Только при $N = 8$ сложная многокомпонентная система становится детерминированной.

3.3 Описание информационных процессов с использованием модели безопасных операций

Модель безопасных операций [24; 29; 47; 51], выполняемых на разных уровнях иерархии суперкомпьютерной системы, позволяет описать информационные процессы в виде мультиграфа запросов.

Методология предложенного подхода включает в себя понятия: многоядерность, мультитредовость, или потоковость, представление информационных потоков в виде графа, вершины которого соединены множеством дуг, то есть в виде мультиграфа, а также использование маркеров для обнаружения вредоносного ПО.

Термин «многоядерность» очевиден, поскольку имеется несколько процессорных мультитредовых ядер.

Термин «мультитредовость» означает, что в каждом ядре одновременно, без переключения регистровых контекстов, выполняется множество тредовых вычислений в параллельном режиме.

Термин «потоковость» может быть использован в смысле обеспечения возможности обработки потоков данных с использованием моделей графов потоков данных.

Термин «маркер» означает набор определенных признаков, представленных в виде кортежа логических переменных.

Вместо классического сигнатурного распознавания используются генеративные таблицы.

Термин «генерируемые (от англ. генеративные) таблицы» подразумевает массив данных, хранимых в гипервизоре в виде таблиц.

Признак допустимости операций задается в виде логической функции (3.8). Полное динамическое состояние процесса определяется тройкой признаков его поведения $\langle W, B, T \rangle$, где W – Work State, рабочее состояние, может иметь значения: Свободен, Зарезервирован, Пассивен, Активен, Задержан и Ожидание; B – Block State, состояние блокировки рабочего состояния; T – Transition State, признак переходного процесса в указанное рабочее состояние или состояние блокировки, связанное с наличием выполняемых в данный момент, но еще незавершенных команд процесса – потока.

Признаки состояния процесса-потока отображаются в одноименных полях регистра t_reg . Явно инициализированы все возможные состояния процессов или потоков с учетом динамического характера процесса выделения вычислительных ресурсов.

Таблица 3.8 – Поля контроля состояния процессов-потоков

Наименование	Описание
TCU	Первые три бита W кодируют состояние процесса 000 – Свободно (нет занятых ресурсов); 001 – Пассивно; 010 – Активно; 011 – Задержано (нет незавершенных команд); 100 – Ожидание (есть запущенные, но еще не завершенные команд); 101 – Приостановлен (нет запущенных и незавершенных команд, блокировка будет снята самим пользователем); 110 – Задержано (есть незавершенные команды, на неопределенный срок блокировка); 111 – Спящий процесс (ресурсы заняты, есть незавершенные команды – принудительное завершение). Последний бит В – признак блокировки: 1 – блокировка; 0 – нет блокировки.
TR	Признак переходного процесса в рабочее состояние Т и/или в состояние блокировки в этом рабочем состоянии, связано с возможным наличием уже выданных на выполнение, но еще не завершенных команд: 01– идет переходной процесс; 10– переходного процесса нет; 11 – ошибка при завершении выданных команд; 00 – ошибка при инициализации TCU (Timing and Control Unit).

Таким образом, не остается состояний неопределенности, которые не разрешимы на всем наборе признаков поведения процесса, нет неявных операций по изменению контекстно – зависимых переходов, которые не формализованы и не контролируются (таблица 3.8).

Главные компоненты любого суперкомпьютера – вычислительные узлы с многоядерными процессорами, многосокетными материнскими платами, памятью, внутриузловой коммуникационной сетью и сетевыми интерфейсами, а также межузловая коммуникационная сеть. Все операции по работе с локальной или глобально-адресуемой распределенной памятью осуществляются контроллером аппаратной транзакционной памяти. Передача сигнальных пакетов (управляющих сообщений), а также данные передаются по высокоскоростной магистральной шине. Параллельные программы (например, OpenMP, MPI, GASNet и т.д.) выполняются на 1000 ядрах суперкомпьютерной системы. При запуске задач указываются названия узлов и списки процессорных ядер, на которых будут запущены приложения:

Узел 1: список ядер 1 ... Узел 2: список ядер 2 ... Узел N: список ядер N.

Процессорные модули могут функционировать в разных режимах. Более того, в процессе выполнения задач процессы могут быть привязаны к разным ядрам узлов всей системы в целом, могут мигрировать в результате изменения контекста выполняемых операций процессором, балансирования нагрузки, работы с глобально адресуемой памятью (тип 3D-Tor) не с локальными сегментами памяти, а с областью данных и кода другого вычислительного узла кластера.

Для учета всех режимов функционирования многоядерных процессорных устройств в разработанной модели операций введем четыре уровня привилегий команд, а именно: Priv0 – уровень привилегий команд процессора (IPL_LEVEL), Priv1 – уровень привилегий ядра ОС (KERNEL_LEVEL), Priv2 – уровень привилегий администратора безопасности (SUPERVISOR_LEVEL), Priv3 – уровень привилегий пользователей (USER_LEVEL).

С учетом того, что существенной особенностью операций в суперкомпьютерных системах является возможность изменения роли субъектов и объектов информационного взаимодействия, для контроля неизменности объектов предлагается использовать специальные механизмы идентификации контекста выполнения процессов.

Процессорные устройства суперкомпьютерных комплексов могут функционировать в разных режимах: на уровне IPL в режиме хост-контроллер (одна управляющая ОС), «гнездовой» контроллер (nested virtualization host, то есть несколько управляющих ОС на уровне IPL), виртуальный вычислительный узел многосокетной платы в реальном режиме (Direct IO computing node, один экземпляр гипервизора на уровне 0 кольца защиты для группы виртуальных машин вычислительного кластера) или в режиме эмуляции с поддержкой запуска на одном сервере нескольких экземпляров задач (VT computing node, несколько гипервизоров на уровне нулевого кольца защиты). Таким образом, с учетом четырех колец защиты максимальное число возможных режимов работы микропроцессорного устройства возрастает до 8. Итак, каждому режиму функционирования однозначно соответствует набор переменных Dom_id, s. Число уровней привилегий при обработке запросов в гипервизоре становится равным восьми. Построена логическая функция безопасности $F_i(R)$, где $R = \{x_1 \dots x_8\}$ – входной вектор-параметр.

При изменении контекста выполнения запроса, происходит переключения между разными режимами, что отражается в наборе кортежа данных генеративных таблиц в виде семи переменных $x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8$, которые задают функцию оценки безопасности состояния современной суперкомпьютерной системы как непротиворечащие требованиям политики безопасности.

В результате любой инициатор процесса доступа может использовать только разрешенные последовательности операций, признак которых задается в виде логической функции, которая определена на кортежах значений восьми переменных:

$$\mathbf{Fi} = (\text{Opcode}, \text{Dom_id}, s, \text{Ord}, \text{Context_type}, \text{Context_id}, \text{tcu}, \text{tr}) \quad (3.8)$$

$x1 = \text{Opcode}; x2 = \text{Dom_id}; x3 = s; x4 = \text{Ord}; x5 = \text{Context_type}; x6 = \text{Context_id}; x7 = \text{tcu}; x8 = \text{tr}.$

s – предикат, определяющий контекст выполнения процесса (потока):

$$s = \begin{cases} 0, & \text{если } \max(\text{Priv}_i, \text{Priv}_j) = \text{Priv}_j, \text{ повышение уровня привилегий;} \\ 1, & \text{если } \min(\text{Priv}_i, \text{Priv}_j) = \text{Priv}_j, \text{ понижение уровня привилегий;} \end{cases} \quad (3.9)$$

Ord – предикат, задающий признак родительского или дочернего процесса (потока):

$$\text{Ord} = \begin{cases} 0, & \text{процесс или поток родительский;} \\ 1, & \text{в противном случае;} \end{cases} \quad (3.10)$$

Параметр **Context_type** принимает следующие значения и определяет изменения контекста выполнения процесса (потока):

- $\text{Context_type} = 1$ (в двоичной кодировке 01) соответствует операциям чтения или записи в области памяти приложений;
- $\text{Context_type} = 3$ (в двоичной кодировке 11), осуществляются операции чтения или записи в привилегированную область памяти;

- Context_type = 2 (в двоичной кодировке 10), ошибка записи в буфер данных;
- Context_type = 0 (в двоичной кодировке 00) режим ожидания новых запросов, при этом не осуществляются операции записи данных.

tcu – переменные для контроля результатов выполнения запроса, параметр **tr** введен для контроля контекста выполняемых операций.

Context_id – уникальный идентификатор запроса.

Переменная **tcu** принимает следующие значения:

X01X – приостановка выполнения родительского приложения с блокировкой на неопределенный срок и перехват управления сторонним ПО с повышением уровня приоритета выполнения до максимума;

011X – ожидание завершения с запуском дочерних потоков (попытка снять блокировку доступа к ресурсам и запустить вредоносное ПО, пока штатный родительский процесс находится в режиме TIME_WAIT);

111X – принудительное завершение всех «спящих» процессов; 0101 – ресурс занят приложением – штатный режим запуска задачи (синхронный режим доступа к ресурсам);

100 – ресурс занят приложением – штатный режим запуска задачи (асинхронный режим доступа к ресурсам); 0X1X – штатное завершение активной задачи.

Таблица 3.9 – Соответствие набора переменных Dom_id, s уровню привилегий и режиму функционирования процессора

Dom_id	s	Режим функционирования процессора	Соответствие уровню иерархии обработки в гипервизоре	Уровни привилегий
00	0	хост-контроллер оборудования (SMM- режим Bare Metal гипервизора – самый привилегированный)	Уровень S8	IPL (BMC-контроллер, уровень инструкций процессора)
00	1	«гнездовой» контроллер оборудования (аппаратный гипервизор - монитор виртуальных машин)	Уровень S7	
01	0	Direct IO computing node (управляющая ОС на физическом сервере в реальном режиме)	Уровень S6	Kernel_level (нулевое кольцо защиты, ядро управляющей ОС)
01	1	VT computing node (управляющая ОС на физическом сервере в режиме эмуляции)	Уровень S5	
10	0	Менеджер ресурсов в режиме прямого доступа к оборудованию	Уровень S4	Supervisor_level (первое кольцо защиты, взаимодействие ядра ОС с физическим оборудованием)
10	1	Менеджер ресурсов среды виртуализации (паравиртуализация)	Уровень S3	
11	0	Управляющая ОС кластера виртуальных машин	Уровень S2	Application_level (третье кольцо защиты, уровень приложений)
11	1	Гостевые «гнездовые» ОС виртуальных машин	Уровень S1	

Opcode либо равен 0, если нет потока команд или пустая команда (NOP), либо 1, если содержит последовательность команд ($\neq \text{NOP}$), M – базовое множество метаопераций, $\text{Opcode} \in M$. Таким образом, существует упорядоченный набор кортежей переменных, который позволяет однозначно идентифицировать состояние как разрешенное или запрещенное для всех возможных изменений контекста операций в разных режимах функционирования процессоров (см. таблицу 3.9).

С использованием введенных обозначений модель функционирования узлов суперкомпьютерной системы может быть представлена конечным автоматом вида:

$$\text{mod}_i = (E_i, R_i, \text{start}, \text{Priv}_i, \mathbf{F}_i, \mathbf{P}_i, \mathbf{V}_i), \quad (3.11)$$

где переменные E_i и $\mathbf{V}_i \in \mathbf{E}$ – задают множество событий или входных воздействий, изменяющих состояния многоядерного процессора, переменная start – задает начальное состояние узла, переменная Priv_i – уровень привилегий в R_i состоянии; $\mathbf{P}_i : R_i \rightarrow \{1|0\}$, $\mathbf{P}_i$ – логическая функция оценки допустимости состояния, указанная в формуле (3.8); отображение $\mathbf{F}_i: R_i \times \mathbf{V}_i \rightarrow R_j$ – задает функцию перехода из состояния R_i в R_j под внешним воздействием $\mathbf{V}_i$. Предложенная модель операций позволяет оперативно построить отображение $R_i \rightarrow \{1|0\}$ как конъюнкцию простых предикатов, характеризующих состояния суперкомпьютерной системы.

Вредоносное ПО может противодействовать своему обнаружению с помощью компрометации счётчика тактов и временной выгрузки из памяти. Так как современные СЗИ функционируют на более высоких уровнях взаимодействия процессов в среде выполнения команд (уровни S1-S4), поэтому они не могут блокировать действия вредоносного программного обеспечения на уровнях S5 – S8.

Таблица 3.10 – Сравнение моделей безопасных операций с точки зрения реализаций функций управления и разграничения доступа

Название модели	Изоляция процессов, мультидоменная защиты	Контроль доступа к объектам на всех уровнях иерархии	Контроль полномочий на всех уровнях иерархии	Контроль контекста выполнения операций на уровне процессора и контроллера памяти	Контроль тайминга выполнения операций на уровне процессора и контроллера памяти
Авторская модель безопасных операций	8 доменов защиты + механизмы верификации на каждом уровне доменной защиты	Число уровней иерархии субъектов однозначно соответствует число уровней иерархии объектов защиты	Пронумерованы все режима работы системы. 8-уровневая иерархическая песочница	Введены и формализованы новые параметры: Delta Diff Context_id Context_type Ord Dom_id	Введены и формализованы новые параметры: TR TCU Opcode
Модель Грэхама-Деннинга	нет	нет	нет	нет	нет
Модель Кларка-Вильсона	нет	нет	нет	нет	нет
Субъектно-ориентированная модель изолированной программной среды	Нет механизмов верификации на каждом уровне доменной защиты	Контроль только на основе меток	Контроль на основе мандата	нет	нет
Классическая модель Take-Grant	Нет механизмов верификации. Нет механизмов верификации на каждом уровне доменной защиты	нет	нет	нет	нет
ЭМЗАС-модель Воронежский институт МВД, ФСТЭК	Не предложены критерии идентификации угроз в мультидоменной системе	Контроль только на основе меток	Контроль на основе мандата + элементы ролевой ПБ	нет	Нет
Модель Росатома	Не предложены критерии идентификации угроз в мультидоменной системе	Контроль только на основе меток	Контроль на основе мандата + элементы ролевой ПБ	нет	Нет

Сравнение моделей безопасных операций с точки зрения реализаций функций управления и разграничения доступа приведено в таблице 3.10. Исходя из данных, приведенных в ней, можно сказать, что во всех существующих традиционных моделях безопасных операций **не разработаны и не формализованы функции контроля полномочий субъектов и доступа к объектам на всех уровнях иерархии СК, а также функции контроля контекста и тайминга выполнения операций.**

Главная проблема заключается в том, что нет доверия к работе сложных систем, так как отсутствуют средства контроля выполнения операций на всех этапах их «порождений». Некоторые команды в определенный момент времени могут нарушать требования ПБ, формируя «ложные транзакции». Как решить эту проблему в случае экзамасштабных суперкомпьютерных систем?

Специфика процессов в них – огромное число потоков для решения инженерно-расчетных задач, которые выполняются на тысячах процессорных ядрах. Выход одного ядра может привести к недоступности сервисов на 10 с, что весьма критично для ресурсоемких распределенных вычислений. Более того, в современных суперкомпьютерных системах объем обрабатываемых данных составляет Пбайты (10^{15} вычислений операций с плавающей точкой в секунду).

Вентильная сложность чипов процессорных устройств и плат памяти стала сравнимой с размерами обрабатываемых данных, что делает уязвимыми для компьютерных атак на компоненты СК в процессе функционирования. Проблема «больших данных» приводит к тому, что нельзя пренебрегать малыми разрядами кодируемых данных, поскольку изменения в небольшом участке может привести к переносу разрядов и появлению значащих битов в последовательности команд, обрабатываемых процессорным устройством.

Процессорные ядра микропроцессоров могут быть разными по типу поддерживаемых в ядре процессов выполнения команд одного или множества процессов-тредов – сверхлёгкие, легкие, средней тяжести, тяжелые:

- сверхлегкие ядра графических процессоров, для которых характерна асинхронность тредов типа WARP;
- легкие ядра векторных процессоров типа применяемых в суперкомпьютере BG/Q, в которых используются мощные устройства обработки коротких векторов SIMD с шириной обработки 512 разрядов;
- средние ядра RISC-микропроцессоров типа MIPS и суперскалярных процессоров;
- тяжелые ядра гибридных процессоров с большим количеством контроллеров памяти и мощным сетевым интерфейсами.

Для контроля границ адресации и смещения в очереди сообщений (конвейере) обрабатываемых запросов и блокирования выполнения «ложных транзакций» формируются генеративные таблицы на этапе подготовки задачи и хранятся в специальных тредовых регистрах t_reg .

Сначала передается стандартная процедурная составляющая – опкод инструкций (набор кода данных и кода команд, модификаторов и флагов условных переходов). Затем в структуру кадра (фрейма) каждого запроса пользователя вводятся 8 дополнительных регистровых полей. Модуль защиты загружается с уровнем привилегий IPL и осуществляет монопольный контроль выделения вычислительных ресурсов сервера, просматривая данные из дополнительных регистровых полей. Ведется мониторинг ссылок безопасности на всех уровнях иерархии конвейера обработки запросов.

Параметры Dom_id и s позволяют задать тот интерфейсный уровень, на котором обрабатывается запрос по схеме клиент – серверное взаимодействие. Context_type задает контекст выполнения операции (системная область памяти или нет), требуется ли переключение в привилегированный режим. В параметре Context_id хранится вычисленное хеш-значение, уникальное для каждого процесса в момент его создания.

Параметр Ord идентифицирует создание базового или порожденного процесса. Поля tsc и tr регистра t_reg позволяют отслеживать статус выполнения трассы. Введен механизм эмуляции с верификацией выполнения исходящих директив на 8 уровнях предложенной модели операций.

Процессы обработки запросов пользователей декомпозируем на 8-уровневую иерархическую структуру, то есть представляет собой конвейер, который имеет 8 интерфейсных уровней. На основе предложенной декомпозиции *модель безопасных операций* можно конструктивно представить с помощью мультиграфа запросов, который описывает разрешенные механизмы инициализации процессов доступа к прикладным и системным информационным ресурсам. В результате предложенной формализации описание скрытых угроз сводится к введению контекстно-зависимых переходов в мультиграфе запросов $G = G(k)$ (см. рисунок 3.11). Характеризуем его математической моделью дискретной динамической системы:

$$x(t+1) = F(x(t); p) = F(x(t); \quad (3.12)$$

$k(t), v)$ с начальным условием $x(0) = x_0$, q – число уровней квантования аргументов функции F : $x(t) = (x_1(t), x_2(t), \dots, x_n(t))$ – состояние ДДС;

k – исполняемые коды. $k = (k_1, k_2, \dots, k_n)$ – «управляющий» параметр формализован в виде набора кортежей;

$\mathbf{p} = (k, M)$ – вектор-параметр, в котором введены дополнительные поля в кодировании очередей сообщений обработки запросов на этапе верификации в эмуляторе. Помимо входных данных в виде опкода, добавлен «управляющий» параметр;

$v = \{\mathbf{M}\}$ – набор метаопераций, т.е. поток входных данных в виде опкода.

$(t_0 + k * \Delta)$ – длительность выполнения k инструкций в процессорных тактах;

n_0 – число инструкций в трассе до трансляции в набор директив;

U_i – i -ая инструкция штатного ПО на S_i уровне, где $i = 1, \dots, n_0$;

V_j – j -ая инструкция стороннего агента где $j = 1, \dots, n_v$;

n_m – число инструкций в обработчике S_i на разных интерфейсных уровнях после сборки команд непосредственного сценария выполнения исходящих директив;

p – вероятность того, что при выполнении инструкции трассы произойдет переключение из одного режима функционирования процессорного устройства в другое (проход по нижней дуге); $p = p_i + \delta p$, где p_i – вероятность переключения в i -ый режим работы, а δp – корректирующий параметр, учитывающий вероятность задержки в распространении сигналов при изменении режима работы аппаратуры (тактовой частоты процессора, скорости кэширования, тактовой частоты работы памяти и т.д.);

q – вероятность выполнения инструкции трассы без изменения контекста выполняемых операций (проход по верхней дуге).

В системе наблюдаются переходные процессы, поскольку согласно (3.11) под внешними управляющими воздействиями меняется состояние процессора как конечного автомата с программным управлением.

Внутренний цикл работы процессора на аппаратном уровне определяется временными интервалами (t – периоды синхронизации).

Введенный автором дополнительный уровень доменной защиты S8 – это уровень, на котором осуществляется верификация и контроль сборки команд непосредственного сценария выполнения исходящих директив (см. рисунок 3.2). Уровень квантования Δ равен $1/8$. При $\Delta < 1/8$ уровне триггерных схем по переднему фронту сигнала синхронизации генерация машинных кодов доступ блокируется, никакие новые запросы не обрабатываются. Только на интервалах, кратных $1/8$, можно влиять на процесс обработки команд за счет неконтролируемых контекстно-зависимых переходов на разных интерфейсных уровнях. Минимальная скорость загрузки кэшируемых данных порядка 1 Гбайт за такт процессора.

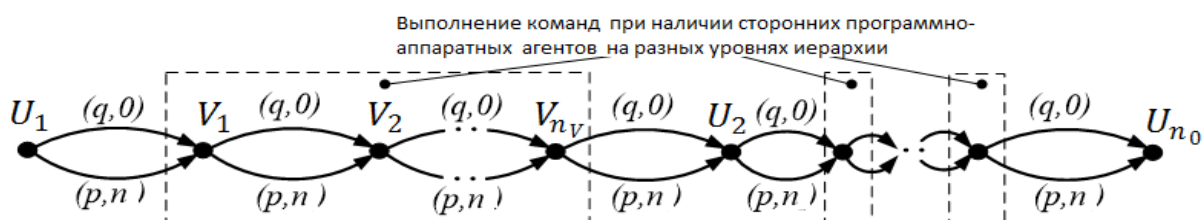


Рисунок 3.4 – Мультиграф запросов $G(k)$

3.4 Выводы по разделу:

1. При декомпозиции информационных процессов взаимодействия в виде 8-уровневой иерархической структуры на основе предложенной модели безопасных операций появляется «взаимно-однозначное соответствие» между процессами выполнения программ на СК, изменениями состояний гипервизора и аппаратуры СК. Отображение становится изоморфным;
2. Ряд особенностей технологии аппаратной виртуализации ускоряет работу виртуальных машин и повышает уровень безопасности. Аппаратная поддержка виртуализации и механизм многоуровневой защиты позволяет снизить накладные расходы на создание изолированной среды исполнения;

3. Использование виртуализации было неприменимо к микропроцессорам предыдущего поколения из-за низкой производительности. Введение дополнительных уровней привилегий и уровней защиты сильно замедляло работу системы в целом;
4. Высокая производительность СК, наоборот, позволяет быстро анализировать дескрипторные таблицы, вычислять хеш-значения процессов. Их отличает: самодиагностика, многоуровневая защита, программирование с использованием нефункциональных, непроцедурных языков – цепочки вычислений в виде селекторов, подстановок справа, слева функциональных вычислений, многоуровневое распараллеливание алгоритмов и т.д.;
5. Разработана модели безопасных операций, рассматривающей с позиций «целостности» среды исполнения как аппаратуру СК, как на уровне микропроцессора, так и системного ПО, описывающая ее процессы в виде декомпозиции на 8-уровневую иерархическую структуру: каждый уровень привилегий процесса однозначно соответствует номеру домена защиты и режиму работы микропроцессора. Введены интервальные ограничения по времени при обработке запросов;
6. Предложено использовать модель угроз целостности среды исполнения процессов СК, в которой вместо классической связки «субъект, объект, предикат» реализована новая парадигма написания правил политики безопасности и заданы новые сущности – «субъект», «объект» и «дескриптор оценки безопасности состояний»;
7. Для каждой i -ой угрозы вычисляется значение функции оценки безопасности состояний, аргументы которой заданы в виде конъюнкции предикатов из восьми логических переменных, субъектами являются гостевые и управляющие ОС, а объектами – компоненты гипервизора и контроллера транзакционной памяти.
8. Введена новая характеристика для создания защищенных суперпроизводительных систем: коэффициент готовности к безопасной эксплуатации СК.

ГЛАВА 4. МЕТОДЫ ГИБРИДНОЙ (РЕАКТИВНОЙ И ПРОАКТИВНОЙ) ЗАЩИТЫ СТАЦИОНАРНЫХ И БОРТОВЫХ СУПЕРКОМПЬЮТЕРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ С УЧЕТОМ САМООРГАНИЗАЦИИ И АДАПТАЦИИ К ТИПУ КОНФИГУРАЦИИ И ПОТОКУ ОБРАБАТЫВАЕМЫХ ДАННЫХ

Главный вопрос, на который предстоит ответить, состоит в следующем:

Какие типы стационарных и бортовых суперкомпьютеров имеются, в рамках каких суперкомпьютерных центров и объектов инфраструктуры государственной важности они используются?

Какие новые угрозы информационной безопасности исходят непосредственно от суперкомпьютеров, какие суперкомпьютеры наиболее опасны?

Какие новые возможности обеспечения кибербезопасности могут предоставить суперкомпьютеры, суперкомпьютеры какого типа для этого лучше подходят?

Важным результатом данной работы, тем не менее, считаем сделанный вывод о необходимости активизации в России работ по суперкомпьютерам, способным работать с большими объемами данных. Лучше всего, если это будут специальные суперкомпьютеры на отечественной элементной базе, но возможны и специальные решения с применением технологий создания кластерных суперкомпьютеров.

Считаем, что направления работ в ведущих зарубежных странах по суперкомпьютерам такого типа и их программному обеспечению должны тщательно отслеживаться и анализироваться, на уровне того, как в свое время отслеживались работы по ядерному оружию.

Рост производительности микропроцессоров теперь происходит только за счет увеличения количества ядер. Необходимо разрабатывать принципиально новую модель обеспечения ИБ суперкомпьютеров с виртуализацией с элементами принципиально новой архитектуры.

Понятие «принципиально новый подход» предполагает то, что в ней реализованы следующие принципы: принцип избыточного параллелизма, принцип неоднородности памяти, принцип оптимального планирования и асинхронности на основе мультиграфа запросов, декомпозицию информационных процессов на 8-уровневую иерархическую структуру, механизм маркерного сканирования и введения интервальных ограничений по времени и повышение энергоэффективности операций при обработке запросов [23; 26; 27; 28; 94].

Для технологии 28 нм и кристалла 20x20 мм приведены данные по энергоемкости разных операций на кристалле. Выполнение одной 64-разрядной операции над операндами с плавающей точкой в функциональном устройстве требует 20 pJ. Передача данных по 256 разрядной шине на разные расстояния – 26 pJ, 256 pJ и 1 nJ. Обращение к статической памяти на кристалле – 50 pJ, а внешнее обращение на сигнальные линки – 500 pJ (256 разрядов), к памяти – 16 nJ (256 разрядов).

Затраты на передачу данных и выполнение обращений к памяти являются главной проблемой суперкомпьютерных систем. Это объясняет уделяемое большое внимание технологиям передачи данных на всех уровнях, а также разработке методов локализации данных и вычислений с целью минимизации перемещения данных и команд.

4.1 Пример оценки результативности новых решений и технологий

Поскольку суперкомпьютер является как источником угрозы, так и средством обеспечения кибербезопасности, то естественно было бы сначала разобраться с понятием суперкомпьютера, его основными характеристиками и типами.

Суперкомпьютеры служат для решения определенных задач, а от типов решаемых задач зависит требуемая степень обеспечения безопасности. По этой причине ставится вопрос о систематизации информации о разных вычислительных центрах, а также критических инфраструктурных образованиях, куда входят суперкомпьютеры.

Суперкомпьютерные центры образуют сети, подчиненные обычно различным ведомствам. Доступ к таким сетям может происходить из глобальных сетей, что требует применения защитных средств. Это также рассматривается.

Например, главной угрозой последних лет авторы считают поставленные в Китае и США работы по встраиванию закладок в изготавливаемые кристаллы графических процессоров (GPU). Эти закладки программируемые, могут по сложным адаптивным алгоритмам анализировать информационные потоки, обрабатываемые данные, состояние памяти, причем совсем незаметно, поскольку затраты на это очень незначительны по сравнению с общей мощностью GPU. Примечательно, что в этих закладках применяются методы программирования задач обработки символьной информации на функциональных непроцедурных языках, это технологии, бурно развивавшиеся в 70-х и 80-х годах, сейчас произошло их возрождение, но это связано не только с программированием закладок.

Другой пример – обратный, использование суперкомпьютера для обеспечения кибербезопасности. Многие специалисты склоняются к тому, что для этого надо уметь быстро анализировать огромные массивы данных о работе сетей, объемы оцениваются в петабайты, а когда говорят о потоках обрабатываемой информации – то возникает уже использование единиц экзабайтов, зетабайтов и йоттабайтов.

С такой задачей может справиться не просто суперкомпьютер, а как уже отмечалось, суперкомпьютер, оптимизированный на эффективную работу с данными, в котором решена проблема преодоления “стены памяти”. Такие суперкомпьютеры обычно имеют массово-мультитредовую архитектуру, разрабатываются лишь в США и Китае. Эти суперкомпьютеры обладают еще одной важной особенностью для обеспечения кибербезопасности – в них поддерживается многоуровневая аппаратная защита программ и данных, в существующих вариантах это 4-х уровневая защита, в перспективных будет 8-ми уровневая.

Наиболее подробной и структурированной информация о суперкомпьютерных центрах получилась по США, что является отражением высшего в мире уровня развития суперкомпьютерных технологий (СКТ) в этой стране. Информация этого плана по Китаю интересна тем, что это страна, догоняющая США в области СКТ, причем во многом сравнимая в этой области с США, а по некоторым технологическим направлениям и перегнавшая.

Для Китая характерна сильнейшая поддержка работ в области СКТ со стороны государства и высочайшая мобилизация значительных людских ресурсов для работы в данной области. Также характерно более активное применение новых революционных идей в области СКТ, в чем США, например, более консервативны, хотя по уровню креативности они, безусловно, являются страной – мировым лидером.

При рассмотрении центров выделялись научно-технические промышленные и исследовательские центры. Отдельно рассматривались специальные центры военных, разведслужб и высшего государственного управления. По понятным причинам, по линии специальных центров информация имеется фрагментарная. Выделялись центры с большим потреблением энергии (более 50 Мватт), обращалось внимание на тип суперкомпьютеров, которые в них предполагается использовать, режимы их использования.

Обе страны (США и Китай) имеют кибервойска и сформированные командования этих войск. Ясно, что задачи этих войск – оборона (кибербезопасность) и нападение (разработка угроз и проведение атак).

Если рассматривать обеспечение кибербезопасности, то в самом “невыгодном” положении находится США по причине своего лидирующего положения в мире и большому количеству секретов, которые многие хотели бы узнать. По этой причине работы по обеспечению кибербезопасности в этой стране можно считать образцовыми, хотя они часто подвергаются критике, а это еще одно подтверждение их образцовости. Про группы американских хакеров говорят меньше, хотя они существуют и очень сильны.

Если рассматривать вопросы разработки угроз и атак, то в прессе чаще упоминается Китай, специалисты которого атакуют США. Имеется информация об успешных атаках и серверов российских организаций, причем не только ведущих НИИ, но и учебных заведений и институтов Российской академии наук. По имеющимся данным, группы китайских хакеров контролируются государством, количество их участников значительно. Например, только в группе «Красные хакеры-братья Мао» (это подразделения военных хакеров) насчитывается около 1 млн. специалистов, если включить их агентов по всему миру.

Еще одно преимущества Китая в плане угроз, атак и ведения разведки – наличие огромного количества китайских студентов, аспирантов и специалистов за пределами страны в самых разных организациях, особенно в США.

Появление супер-ЭВМ привнесло изменения в реализации подходов, методов и способов обеспечения информационной безопасности. В США и Китае в этой связи принята новая концепция обороны, включающая в себя создание суперпроизводительных компьютерных систем «нового поколения», что позволит их использовать как главное стратегическое оружие в будущих кибервойнах.

Например, в связи с тем, что производительность супер-ЭВМ превысила петафлопс, а алгоритм шифрования с использованием законов простых чисел имеет линейную зависимость времени вычислений последовательностей простых чисел, то данное обстоятельство повышает эффективность взлома киберсистем с открытым ключом.

Другой пример – появление суперкластеров и суперкомпьютеров с глобально адресуемой памятью расширяет возможности эффективной работы с большими объемами данных непосредственно в оперативной памяти. Такие возможности необходимы для анализа поведения систем на предмет обнаружения вторжений в их работу.

Например, в США и Китае для установки на суперкластерах и суперкомпьютерах был разработан новый класс операционных систем, соответственно - Cray MTK 4.0, XTS-3000, Blacker, а также Kylin. С точки зрения защищенности, разработанные операционные системы обладает следующими улучшенными характеристиками:

- использование не менее четырех уровней привилегий позволяет реализовать многоуровневую («усиленную») ролевую политику безопасности (в «традиционных» ОС класса Unix и Windows NT используются только 2 уровня привилегий);

- реализуют поддержку массивно-мультитредовых процессоров с глобально адресуемой памятью.
- указанные функциональные возможности закладываются уже на начальном этапе «жизненного цикла» разработки операционной системы;
- создается изолированная программно-аппаратная среда для выполнения задач, осуществляя при этом «прозрачный» контроль за действиями программных агентов на разных интерфейсных уровнях операционной системы; собственная система обеспечения отказоустойчивости; «прозрачная» технология сборки ОС.

В «новых» ОС существуют 4 уровня привилегий: USER_LEVEL, SUPERVISOR_LEVEL, KERNEL_LEVEL, IPL_LEVEL. С точки зрения архитектуры, указанные ОС являются микроядерными. “Микроядерная” означает то, что в привилегированном режиме остается работать только очень небольшая часть ОС, называемая микроядром.

Микроядро защищено от остальных частей ОС и приложений. В состав микроядра обычно входят машинно-зависимые модули, а также модули, выполняющие базовые функции ядра по управлению процессами, обработке прерываний, управлению виртуальной памятью, пересылке сообщений и управлению устройствами ввода-вывода, связанные с загрузкой или чтением регистров устройств. Уровень, позволяющий управлять защищенным режимом процессора, осуществлять загрузку и контролировать работу системы на уровне инструкций процессора называются IPL (Instruction Processor Level, IPL-Level). Набор функций микроядра обычно соответствует функциям слоя базовых механизмов обычного ядра (Kernel-level). Такие функции операционной системы трудно, если не невозможно, выполнить в пространстве пользователя.

Все остальные более высокоуровневые функции ядра оформляются в виде приложений, работающих в пользовательском режиме (User_Level).

Менеджеры ресурсов, являющиеся неотъемлемыми частями обычного ядра – файловая система, подсистемы управления виртуальной памятью и процессами, менеджер безопасности, – становятся «периферийными» модулями, работающими в режиме супервизора (Supervisor-Level). Обратим внимание, что в суперкомпьютерных центрах ведется также работа по автоматизированной отработке и оценке встроенного в системы оружия программного обеспечения (ПО). Для решения этих задач применяются мощные суперкомпьютеры, входящие в сложные комплексно-моделирующие стенды (КМС). Фактический и планируемый рост мощности вычислительных ресурсов этих центров представлен. Отметим, следующие особенности:

1. Большая часть ресурса обеспечивается суперкомпьютерами «заказного» типа, особенно от фирмы Cray. В этих суперкомпьютерах имеется, по-крайней, мере нестандартное решение по организации коммуникационной сети.

2. Эти сети также кроме более высоких характеристик (разы по пропускной способности линков) поддерживают в большей или меньшей степени межузловую глобально-адресуемую память.

3. Имеются суперкомпьютеры кластерного типа, наиболее впечатляет стандартное де-факто решение для всех центров от фирмы Arpro с гибридным суперкластером Xstream-X серии, в которых есть легкие (на CPU) и тяжелые (на CPU + GPU) вычислительные узлы, а также тяжелые узлы по разделяемой на узле оперативной памяти и количеству процессорных ядер, работающих с этой памятью.

4. В общедоступный ресурс выведены именно «заказные» суперкомпьютеры как наиболее дефицитный и ценный ресурс.

Следующий промышленный суперкомпьютер этой линейки – Cray XK6, он уже осваивается в Окриджской лаборатории Министерства энергетики США. Он включает графические ускорители, а дальнейший путь развития – гибридные суперкомпьютеры с разными “заказными” узлами и даже потом гибридными микропроцессорами, объединяющими возможности суперскалярных CPU, массово-мультитредовых CPU и графических ускорителей GPU.

Другой пример – комментарии по поводу неожиданного перехода в середине этого года 45-летнего Стива Скотта (Steve Scott), долгое время занимавшего пост технического директора компании Cray, в компанию NVIDIA. Стив Скотт будет участвовать в разработке гибридного высокопроизводительного микропроцессора, который включает как массовую мультитредовость для обеспечения толерантности работы с памятью и сетью (как в микропроцессоре Threadstorm фирмы Cray, являющийся базовым для Cray XMT), так и высокие вычислительные возможности за счет операций с векторами и наличия множества синхронных тредов (как в графических процессорах Fermi и Kepler, в создании которых NVIDIA успешно работала).

По такому пути пошли и в китайском проекте CT-2, усилив возможности J7 (российский проект – прототип массово-мультитредового микропроцессора) такими мощными вычислительными средствами, как синхронные треды и SIMD-устройства.

Американские специалисты пойдут к разработке гибридного микропроцессора с другой исходной позиции, добавив в графический процессор с синхронными тредями (Fermi NVIDIA) возможности обеспечения толерантности за счет множества асинхронных тредов, работающих еще и с глобально адресуемой памятью (Cray MTA/XMT).

Цель одна – создание идеального для приложений гибридного микропроцессора. Будет называться такой гибридный микропроцессор Cray SCORPIO, обещанный по программе DARPA HPCS векторно-мультитредовый процессор для суперкомпьютеров с перспективной архитектурой. Здесь также ясно, что будет серьезное соревнование с Китаем, но уже сейчас еще неизвестно, кто победит. Еще есть предположение, что в такие микропроцессоры, как и просто в графические, будут встроены недеklarированные средства для скрытых компьютерных атак. Это касается как американских, так и китайских подобных изделий. Таким образом, американский центр ориентируется на применение массово-мультитредового суперкомпьютера с глобально адресуемой памятью, что служит наглядным примером оружия для ведения кибервойн.

По этой причине резко выросла важность имитационного и эмуляционного исследования эффективности сочетаний решений по этим направлениям. Принципиально новым является то, что ее цели планируется достичь путем развития технологий встроенных систем, для которых в первую очередь требуется энергетическая экономичность и которые, согласно новой доктрине, имеют высокий приоритет при разработке.

4.2 Разработка прототипа с реализацией новых архитектурных принципов проектирования и механизмов интегрированной защиты суперкомпьютерных комплексов для АС разного класса с гибкой конфигурацией окружения и контекстом вызовов сервисов, сбором метрики приложений

В общем, единственный реальный способ анализировать программу, исполняющуюся в таких условиях, – начать с неявного предположения, что среда не может произвольно измениться между двумя любыми операциями. Альтернативой является запрет загрузки кода и изоляция динамически создаваемого кода в его собственной среде.

Предыдущие попытки реализации этих процедур не завоевали широкой популярности, так как механизмы изоляции имели проблемы с программируемостью и производительностью, делавшие их менее привлекательными, чем выполнение без изоляции.

Наиболее распространенный механизм – это традиционный процесс ОС, но его высокая стоимость ограничивает его применимость. Аппаратное управления памятью защищает состояние процессора, но также удорожает передачу управления и данных между процессами. Микропроцессор J7 (см. рисунок 4.1 и рисунок 4.2) имеет два мультитредовых ядра (MTcore0 и MTcore1), разработан в свое время с учетом особенностей 90 нм технологии ASIC фирмы Fujitsu (серия 101), рассчитан для работы на частоте 1 ГГц [37, 38]. Версия J7 (семейство «Ангара»).

В одном мультитредовом ядре имеются четыре конвейера команд, каждый из которых работает с 16 тредовыми устройствами, на каждом из которых может выполняться один процесс-тред.

Каждое тредовое устройство содержит регистры управления выполнением потока команд загруженного на него треда (слово состояния со счетчиком команд, признаками и масками исключительных ситуаций и другие регистры), а также достаточно большие наборы 64-разрядных архитектурных регистров для хранения чисел с фиксированной и плавающей точкой, адресуемых однобитовых регистров-признаков и регистров адресов передачи управления.

Переключение с выполнения команд одного тредового устройства на выполнение команд с другого происходит без перезагрузки регистров, за один такт процессора [39].

Кроме того, несколько тредовых устройств могут за такт выдать одновременно команды на выполнение в функциональных устройствах.

Информационная зависимость выполняемых команд с одного тредового устройства контролируется аппаратно посредством таблиц признаков занятости регистров (см. рисунки 4.1 и 4.2).

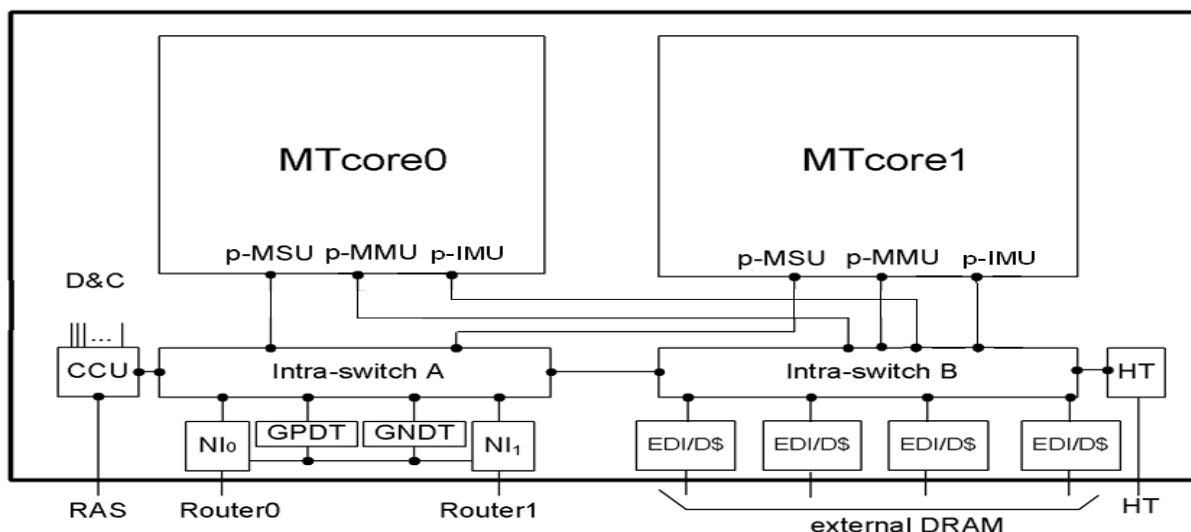


Рисунок 4.1 – Микроархитектура двухядерного микропроцессора J7

Пояснения к рисунку 4.1:

MTcore0, MTcore1 – мультитредовые ядра, каждое из которых содержит 4 домена защиты и 64 тредовых устройства;

p-MSU – порт мультитредового ядра от блока приема/выдачи сообщений (MSU);

p-MMU – порт мультитредового ядра от блока трансляции виртуальных адресов (MMU);

Intra-switch A – внутрикристалльный коммутатор для передачи сообщений, относящихся к выполнению сетевых операций;

Intra-switch B – внутрикристалльный коммутатор для передачи сообщений, связанных с выполнением операций обращений к памяти;

CCU – блок контроля работы микропроцессора;

D&C – линии диагностики и контроля;

RAS – линии подключения к системе обеспечения надежности, готовности и сервиса (RAS-системе)

NI₀, NI₁ - адаптеры межузловой сети, подключены к маршрутизаторам этой сети;

GPDT – главная таблица доменов защиты задач, отображает уникалы задач на домены защиты мультитредовых ядер, в которых эти задачи решаются;

GNDT – главная таблица дескрипторов таблиц отображения виртуальных номеров узлов задачи на логические;

EDI/D\$ - блоки интерфейса с внекристалльной памятью, включающие кэш данных, блок выполнения атомарных операций и работы с теговыми битами, контроллеры памяти;

HT – блок гипертранспорта (HyperTransport). **p-IMU** – порт мультитредового ядра от блока приема/выдачи кэш-строк команд.

Конец пояснений к рисунку 4.1

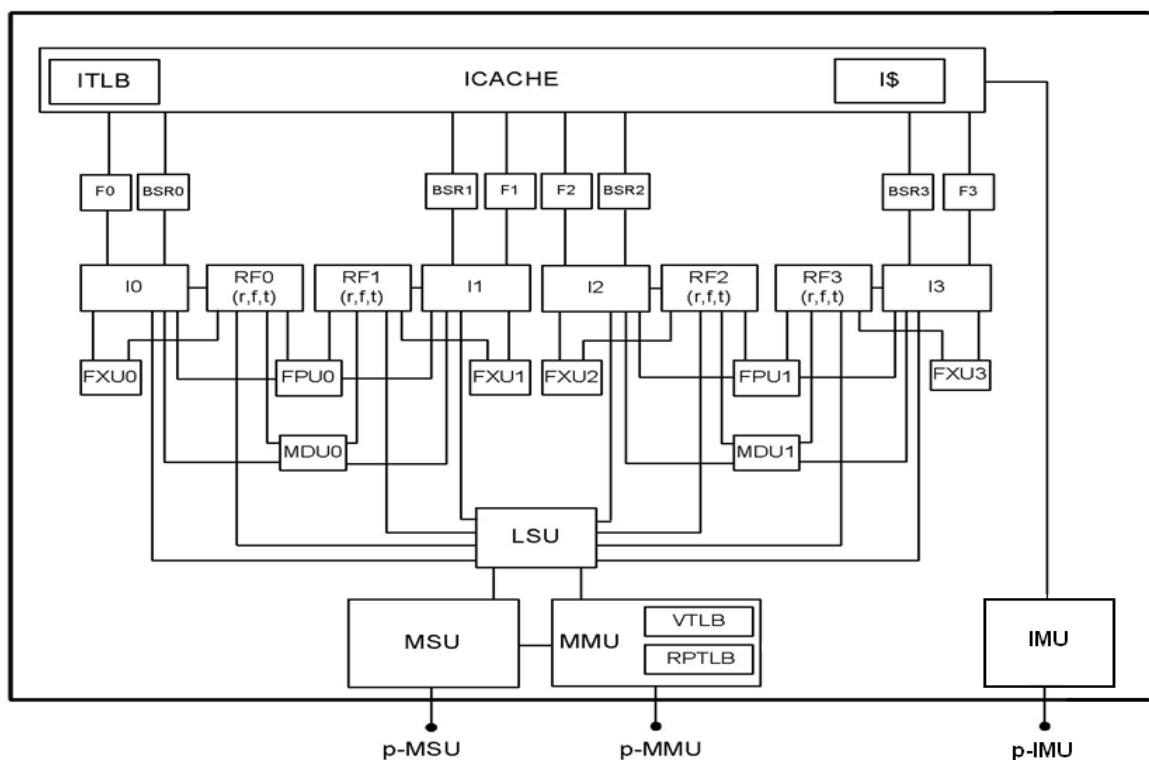


Рисунок 4.2 – Микроархитектура ядра микропроцессора J7

Пояснения к рисунку 4.2:

ICACHE – блок общего для тредовых устройств кэша команд мультитредового ядра;

ITLB – кэш-память дескрипторов страниц команд;

I\$ - кэш-память команд;

F0, F1, F2 и F3 – четыре независимых блока подкачки команд из кэш-памяти и выдачи команд для выполнения из тредовых устройств, каждый из этих блоков работает с 16 тредовыми устройствами;

BSR0, BSR1, BSR2 и BSR3 – блоки обеспечения связи с разными регистровыми ресурсами микропроцессора при выполнении привилегированной команды доступа к ресурсам;

I0, I1, I2 и I3 – блоки проверки готовности выбранных из тредовых устройств команд на выполнение в функциональных устройствах ядра, а также завершения команд после их выполнения в функциональных устройствах;

RF0, RF1, RF2 и RF3 – регистровые файлы тредовых устройств вместе с таблицами отслеживания готовности регистровых операндов команд (г-файл общих регистров, f – файл для чисел с плавающей точкой, t – регистры адресов переходов);

FXU0, FXU1, FXU2 и FXU3 – устройства выполнения сложения-вычитания, логических операций для целых чисел и двоичных кодов;

<F0, BSR0, I0, RF0, FXU0>, ... <F3, BSR3, I3, RF3, FXU3> - конвейеры выполнения команд мультитредового ядра, соответственно 0, 1, 2 и 3;

MDU0 и MDU1 – устройства выполнения умножения и деления, операций сдвига над целыми операндами и двоичными кодами, общие для конвейеров 0-1, а также 2-3.

FPU0 и FPU1 – устройства выполнения операций для операндов с плавающей точкой, общие для конвейеров 0-1, а также 2-3.

LSU – устройство выполнения команд обращения к памяти и удаленных вызовов процедур;

MSU – блок приема/выдачи сетевых сообщений;

MMU – устройство трансляции виртуальных адресов;

VTLB – таблица дескрипторов V-сегментов виртуальной памяти задачи;

RPTLB – таблица дескрипторов R-сегментов и R-страниц общего для задач виртуального адресного пространства.

IMU – блок приема/выдачи кэш-строк команд

Конец пояснений к рисунку 4.2

В одном ядре микропроцессора может одновременно выполняться несколько задач. Каждой задаче ставится в соответствие один домен защиты, причем одна из задач ядра – обязательно операционная система.

Выполняемая в микропроцессоре задача пользователя может одновременно выполняться в доменах защиты его разных ядер, информация о привязке задачи к доменам защиты хранится в специальной таблице микропроцессора.

Предложим новый подход в области создания защищенных СК, основные принципы которого включают в себя новые понятия «субъект-объект» и модели доступа, предполагающая декомпозицию выполняемых операций на 8-уровневый конвейер обработки динамически реконфигурируемых вычислительных структур, переход на соединение в ОС функций управления памятью, доступом, поддержку многоуровневой доменной защиты и планирования параллельной обработки, поддержку высокопараллельных процессов и высокой асинхронности предполагает переход на неклассические высокопараллельные модели организации программ с глобально адресуемой памятью, обладающие высокой асинхронностью параллельных процессов, использование новых технологий передачи данных, новых конструктивов 3D сборки электронных блоков, 4

кольца защиты (в «традиционных» ОС класса Unix и Windows NT используются только 2 кольца защиты), реализацию доменной защита данных, команд и тредовых устройств, разработку ОС и подсистемы обеспечения отказоустойчивости, применение «прозрачной» технологии сборки конструктива на микросхемах и платах (отечественная элементная база).

Создание таблиц занятости регистров с двойной индексацией (помимо счетчиков занятости) выполнено с целью придания системе еще более выраженного асинхронизма и распараллеливания операций на каждом тредовом устройстве (графы множественного ветвления, усложненные case-алгоритмы). При такой технологии нет задержек при обращении к регистровым данным. Модули платы имеют многослойную структуру (слои модулей памяти, слои подсистемы ввода-вывода непосредственно на кристалле, слои процессорных элементов).

Мобильность понимается как «встроенность» производимых вычислений.

Мы сделали следующий ряд ключевых архитектурных и функциональных доработок в части принципов работы вычислительного устройства, повышающие его уровень защищенности, надежности и производительности на основе избыточного параллелизма, многоуровневой доменной защиты, неоднородности памяти [22; 31]:

1. Встроена поддержка глобально-адресуемой памяти размером в несколько десятков Пбайт (32 Пбайт);
2. Введена поддержка восьми уровней привилегий, что позволяет реализовать многоуровневую («усиленную») ролевую политику безопасности. При этом каждый процесс может иметь два статуса – базовый (управляющий) или дочерний (подчиненный), всего таких процессов с учетом 8 уровней иерархии выполнения команд равно 16. Число доменов защиты тоже равно 16. Таким образом, существует явная привязка запускаемых процессов к аппаратным доменам защиты;

3. Работа осуществляется с виртуальным единым адресным пространством в несколько Петабайт. Контроль переходов осуществляется с помощью инициализированных модулем верификации команд маркеров, а не посредством использования программных косвенных ссылок языка Си/Си++ при обработке списков запросов, как это сделано в классических ОС. При таком способе функционирования нет ограничений на размер передаваемых данных (например, в процессорах семейства x86 размер стека 256 байт, при превышении указанного резерва возникает отказ в обслуживании).

4. Предложен способ интеграции со всеми типами гипервизоров (XEN, KVM, VMWare ESX, Microsoft Hyper-V и т.п.);

5. Разработана тегированная архитектура массово-мультитредового процессора с поддержкой технологии аппаратной виртуализации на примере экспериментального прототипа;

6. Введен контроль выполнения операций процессором на всех уровнях иерархии выполнения команд: сначала идет проверка всего набора переменных генеративных таблиц для каждого запускаемого процесса. Если операция нарушает требования ПБ, то опкод операции не передается на выполнение аппаратным ядрам микропроцессорного устройства.

7. В случае идентификации попытки сформировать некорректный запрос (неправильный формат данных, переполнение буфера и т.п.). Запрос блокируется и не передается на дальнейшую обработку аппаратным ядрам микропроцессорного устройства или контроллерам периферийного оборудования;

8. Разработан новаторский способ функционирования распределенной микроядерной ОС Microtek: в ней могут обрабатываться огромные потоки информации (блоки размером в несколько десятков Пбайт). Более того, если не инициализированы все поля записи генеративных таблиц на этапе подготовки задачи (в результате попытки сформировать запрос в виде некорректной операции, нарушения сегментов

адресации, переполнения буфера обрабатываемых данных и т.д.), то данный запрос блокируется и не отправляется на выполнение аппаратному устройству;

9. Все запросы к аппаратным устройствам представлены в виде запросов с оценкой допустимости выполнения операций в виде разрешенных и запрещенных значений функции F_i ;

10. Разработан прозрачный контроль контекстно-зависимых переходов при выполнении операций посредством контроля изменений значащих разрядов в полях t_{cu} , tr регистра t_reg ;

11. Разработан механизм управления сообщениями (порождениями директив) за счет сборки команд непосредственного сценария выполнения исходящих директив на уровне иерархии S_8 ;

12. Модуль верификации контролирует выполнение директив на уровне исполнительного региона процессора.

4.3 Метод реактивной защиты, основанный на виртуализации среды исполнения процессов и контроле запросов на выделение ресурсов в соответствии с правилами политики безопасности в виде модальных структур темпоральной логики

Значения функций фильтрации $V_f = \{1/8, 1/4, 3/8, 1/2, 5/8, 3/4, 7/8\}$.

Это набор всех интервальных значений, кратных $1/8$. Интервально-временные ограничения обусловлены особенностью аппаратной реализации управления режимами функционирования процессорных устройств, о чем уже упоминалось ранее. В зависимости от конфигурации системы выделены четыре класса суперкомпьютеров (три конфигурации стационарных компьютеров и одна бортовых супер-ЭВМ).

Таблица 4.1 – Типы конфигураций СК

Название системы	Число используемых интерфейсных уровней N
Cray XMT 3 Scorpio, IBM BG/Q (легкие ядра) Стационарные суперкомпьютеры	6-7
Ангара ЕС1740, СТ-2 (тяжелые ядра) Стационарные суперкомпьютеры	8
HP Cascade, VMWare Esxi Linux Cluster + Xen / KVM (средние ядра) Стационарные суперкомпьютеры	4-5
SGI, NVidia, Galaxy, Комдив, Грифон, Байкал-М (сверхлегкие ядра), бортовые супер-ЭВМ	2 - 4

Алгоритм вычисления характеристической функции [23; 44] оценки безопасности состояний адаптирован к типу конфигурации СК и потока обрабатываемых данных (таблица 4.1).

Суть новой модели безопасности и введенной логики контроля и выполнения запросов состоит в следующем. Выдаваемые запросы изначально связываются с некоторой информационной структурой, называемой тегом. При прохождении запросом уровня защиты он не выполняется, а только происходит планирование действий для него на этом уровне и фиксация информации о контексте их выполнения, производится соответствующее этому изменению тега запроса.

По мере продвижения запроса через уровни в теге накапливается информация в виде его межуровневой трассы. Работа с тегами запросов на уровнях защиты производится агентами гипервизора.

В конечном итоге запрос попадает на последний, восьмой уровень, собственно в гипервизор, который является самым привилегированным и абсолютно закрытым для несанкционированного вмешательства. Непосредственное выполнение операций запроса над запрашиваемым в нем объектом или объектами происходит именно в гипервизоре на восьмом уровне. Однако это происходит не сразу, лишь после того, как будет проведена проверка тега запроса. Проверка тега производится сравнением его с эталонным значением для правильной трассы, таблица таких значений подготавливается для гипервизора заранее, в виде хеш-функций (сверток) таких путей, по заранее подготовленным мультиграфам.

Выбрано восемь интервалов наблюдений. При выбранной конфигурации изменение контекста выполняемых операций на всех уровнях иерархии становится детерминированным, а не стохастическим. Для вычисления характеристической функции используется только 8 значащих логических переменных. Вычисления производятся с точностью до 8-го разряда. Следовательно, все операции выполняются над конечными числами, а не бесконечно большими величинами.

Фиксация информации о контексте выполнения запросов от вложенных гостевых у управляющих ОС представлена в виде набора дополнительных восьми полей в пакетах запросов, изменение в значащих битах логических переменных которых приводит к изменению тегов запросов.

События, приводящие к изменению тега запроса, связаны с повышением уровня привилегий, обращением к привилегированной области памяти, с изменением статуса прохождения запроса, наличием переходного процесса и запуском подпроцессов при прохождении межузловой трассы, модификацией значений хеш-функций (сверток) для правильной трассы обработки запроса. Физический смысл тега запроса – это и есть описатель состояний процессов, своеобразного контекста выполнения запроса, порождаемого субъектами доступа.

Заранее строится ориентированный мультиграф. Для каждого запроса вычисляется уникальный хеш на этапе инициализации и записывается в генерируемую таблицу гипервизора, который является эталонным значением для правильной трассы. При изменении хеш-значения в указанной логической иерархической структуре, меняются идентификация вершин трассы и нумерация маршрута прохождения, что свидетельствует о подозрительной активности в системе.

Для систем со сверхлегкими ядрами характерна поддержка vGPU, то есть разделение виртуальных графических ядер между несколькими пользователями, поддержка хостовой и гнездовой виртуализации для узкоспециализированных задач. Используются уровни S1, S2, S3, S4. 4 интервала фильтрации $f = \{0, 0.125, 0.25, 0.5\}$.

Для систем со средними ядрами характерна поддержка суперскалярных и векторных вычислений, поддержка многоплатформенных кластерных решений. Используются уровни S1, S2, S3, S4, S5, S6. 5 интервалов фильтрации $f = \{0, 0.125, 0.25, 0.5, 0.625\}$

Для систем с легкими ядрами характерен дополнительный уровень S7 по сравнению со средними ядрами. Это связано с тем, что производители IBM и Cray используют встроенный аппаратный модуль для получения быстрой производительности на уровне привилегий IPL (инструкций процессора), кодирования/декодирования команд, взаимодействия с сетевыми сопроцессорами и контроллерами памяти, введена трехуровневая обработка векторов (короткие, средние и длинные), введено много встроенных команд-функций (intrinsic functions), введены микрокоманды. 6 интервалов фильтрации $f = \{0, 0.125, 0.25, 0.5, 0.625, 0.75\}$.

В системах с тяжелыми ядрами используются средства работы с очень большими данными также выделенные магистрали для работы с сетями передачи данных (датчиками, через которые поступают потоки данных).

Для обычных счетных задач традиционно применяются процессоры с суперскалярными ядрами, а для вычислительных задач, когда доля вычислительных операций высока, также для невычислительных задач, когда ведется логическая обработка и, как правило, для этих задач характерна плохая пространственно-временная локализация обращений к памяти, используются массово-мультитредовые процессоры класса российского «Ангара J7» или китайского СТ-2. В таких системах используется 8 уровней привилегий. 7 интервалов фильтрации $f = \{0, 0.125, 0.25, 0.5, 0.625, 0.75, 0.875\}$.

На каком из уровней необходимо вводить верификацию?

Вместо классического подхода в виде скрытых марковских процессов с использованием алгоритма Баумана-Уэлча мы предлагаем использовать вычисления траекторий на основе темпоральной логики STL, заданных на структурах Крипке с набором вероятностных значений оценки защищенности. Вычисление доверительной вероятности производится по формуле Корнфельда (4.1) и заключается в выборе доверительного интервала в пределах от минимального до максимального результата измерений. Пусть $x_1, x_2, \dots, x_n$ – результаты измерений. Интервал от $\tilde{x} - \Delta x$ до $\tilde{x} + \Delta x$ – доверительный интервал, где $\tilde{x} = \frac{1}{2} *$, $\Delta x = \frac{1}{2} * (x_{MAX} - x_{MIN})$.

Доверительная вероятность для традиционного кластера вычисляется по формуле:

$$p = 1 - \sum \left(\frac{1}{2}\right)^{n-1}, 1 \dots n$$

(4.1)

В традиционных системах учитываются только 2 режима функционирования: R-режим и S-режим, то есть $N = 2$.

Тогда $p = 1 - 1/3 = 2/3 = 0,66666666$. $q = 1 - p = 0,33333333$.

В общем виде для N-режимов процессора и n – наблюдаемых событий формула вычислений представляет собой:

$$p = 1 - \sum \left(\frac{1}{N}\right)^{n-1}, 1 \dots n . \quad (4.2)$$

В зависимости от типа сборки суперкомпьютерной вычислительной системы можно выбирать оптимальный уровень защиты:

1. Для стационарных суперкомпьютеров, имеющих значительные вычислительные ресурсы и мощности, можно использовать конфигурации от четырех до восьми уровней защиты.

2. Для бортовых суперкомпьютеров, которые располагают ограниченными вычислительными ресурсами и которые должны удовлетворять требованиям “жесткого” реального времени, можно уменьшить набор доменов защиты и использовать от двух до четырех уровней, работая напрямую с оборудованием, исключая режимы эмуляции.

В случае суперкомпьютерных систем нельзя пренебрегать разрядами с точностью до 8 знака. С учетом 8-уровневой модели обработки запросов и необходимости 8 значащих разрядов для обработки и хранения данных в цикле вычислений дескрипторов оценки безопасности для стационарных суперкомпьютеров в этом случае $p = 1 - \left(\frac{1}{8}\right)^{n-1} = 1 - 1/9 = 8/9 = 0,88888889$. $q = 1 - p = 0,11111111$. Следовательно, при такой конфигурации стохастический процесс перехода из одного режима функционирования в другой под внешним специальным воздействием или в результате проведения атак становится квазидетерминированным. Доверительная вероятность равна почти 0,9.

Показатель защищенности компонентов среды выполнения программ достигает максимума, когда система перестает быть уязвимой к любым видам воздействий и типам атак. Если $N = 6$, $p = 0,857$, если $N = 7$, то $p = 0,875$.

Для бортовых суперкомпьютеров, если мы уменьшаем число доменов защиты с восьми до четырех ($N = 4$), тогда значение $p = 0,8$, $q = 0,1$. Коэффициенты имеют целочисленное значение, нет разрядов после запятой. Так что можно уменьшить точность вычислений. При этом показатель защищенности компонентов среды выполнения программ уменьшается с 0,9 до 0,8.

При этом состояния СК наблюдаемы: изменение контекста выполняемых операций детерминировано и может быть записано в виде «числа» – набора изменений существенных логических переменных в полиноме Жегалкина, а не случайная величина.

То есть существует "инвариант" - эталон, который является результирующей сверткой вычисляемых значений функции оценки безопасности, заданной на наборе кортежей 8 переменных, при всех возможных вариациях (случайное изменение таймера, разброс характеристик трасс выполнение программ, применение обфускации с целью затруднения обнаружения средств скрытого воздействия и т.д.).

Для каждого значения можно определить, на каком уровне S_i действует вредоносное ПО, к каким сегментам данных или кода обращается, меняет ли контекст при обработки запросов, работает с объектами памяти, идентификаторами процессов, осуществляет подмену при переключение контекста путем перехвата управления, создание подпроцессов в рамках штатного процесса, "инжектирует" чужое адресное пространство и т.д. Обучение и самодиагностика введется с введением паттернов – характерных признаков (или маркеров).

Поля TCU – параметры, характеризующие статус выполнения приложения (Status), кроме бита блокировки В. Это характеристики, которые не входят в структуру при выполнении вычислений. Их можно назвать атрибутивными признаками si – состояния, но на процедуру маркировки они влияют.

Мы также абстрагируемся от специфики команд процессора. Считаем, что запускаем приложение в многоуровневой “песочнице”. Поэтому структура опкода (поле Opcode) не влияет на модальные правила.

Темпоральная логика учитывает мультиветвления по времени и историю порождения разных конфигураций. У разных конфигураций – разные уровни привилегий базового и наследованных от него дочерних процессов. Как только меняется контекст выполнения операций, то меняется и delta-параметр. Вводится контроль – принцип наименьших привилегий Priv. Если $\text{Priv}(si) > 3$, то в штатном режиме записи - $\text{Context_type} = 1$. Если $\text{Priv}(si) < 3$, то Context_type должен быть равен 3.

Это идеальная картина, когда у нас 8 уровней изоляции в виртуальной “песочнице”. Но всегда возможны случайные или преднамеренные воздействия на СК.

В качестве хеш-функции используется Стандарт ГОСТ 34.11– 2012, который пришел на смену ГОСТ 34.11– 94, являющемуся потенциально уязвимым. Этот новый отечественный стандарт хеширования обязателен к применению в СК общего и специального назначения в государственных и коммерческих организациях, для которых необходимо прохождение сертификации в уполномоченных органах (ФСТЭК, ФСБ и подобных). ГОСТ 34.11– 2012 был разработан Центром защиты информации и специальной связи ФСБ России и состоит из двух хеш-функций с длинами результирующего значения в 256 и 512 бит, которые отличаются начальным внутренним состоянием и его частью, принимаемой за результат вычислений. Основное отличие состоит в реализации функции сжатия.

Основная операция функции сжатия состоит из трёх преобразований: подстановки на байтах, транспонирования матрицы байт и умножения 64-битных векторов на матрицу 64×64 в GF(2):

1. S – нелинейная биекция. 512 бит аргумента рассматриваются как массив из шестидесяти четырёх байт, каждый из которых заменяется по заданной стандартом таблице подстановки;
2. P – переупорядочивание байт. Байты аргумента меняются местами по определённом в стандарте порядку;
3. L – линейное преобразование. Аргумент рассматривается как 8 64-битных векторов, каждый из которых заменяется результатом умножения на определённую стандартом матрицу 64×64 над GF(2).

В формуле (3.9) приведен общий вид многочлена Жегалкина. С учетом описанного ранее способа кодирования хеш представлен в виде полинома Жегалкина 8-степени

$$\text{Context_id} = \text{hash}(s_{0..n-1}) = s_0 + zs_1 + z^2s_2 + \dots + z^{n-1}s_{n-1}, \quad (4.3)$$

где $z = 2$, $n = 8$ а s_i – код i -ого символа строки s .

Отличительным математическим свойством хешей является то, что у одинаковых строк значения хеш-функции обязательно равны. Поэтому основная операция, которую позволяют выполнять, – быстрый поиск и сравнение двух подстрок на равенство.

В классических ОС задается только ссылка на объект. Мы предлагаем вычислять хеш для каждого выполняемого процесса обработки. Более того, введена новая логика управления доступам. Контроль доступ субъекта к объектам управляющих и гостевых ОС на основе тегированных полей в пакетах запросов осуществляется на всех уровнях иерархии. Дополнительные теги в пакетах запросов представляют собой префиксы, заданные набором кортежей в виде 8 логических переменных. В цикле 8 этапов вычислений.

На каждом уровне иерархии n вычисляется на основе хеш-подстроки $hash1$ (уровень 1) и записывается в поле `context_id`, далее рекурсивно вычисляется хеш подстроки $hash2$ (уровень 2) и т.д. В качестве результирующего значения получаем $hash8$ (последний уровень иерархии). Необходимо сравнивать хеш-значения, находить одинаковые подстроки на основе бинарного поиска характерных маркеров – определенного набора кортежей переменных с учетом изменений значащих битов.

Первые 6 символов подстановки справа задают контекст выполнения:

- символ – признак непустой операции
- символ – признак повышения уровня привилегий
- 3,4 символы – кодируют уровень привилегий при работе с физической памятью
- 5,6 – признак работы с памятью: коды данных и программ привилегированные или нет.
- 7 символ – признак наличия дочерних процессов
- 8 символ – начало поля `Context_id`,

Оставшиеся 6 символов подстановки слева определяют статус обработки и переменные окружения выполняемого процесса:

- 8 символ – конец поля `Context_id`
- 5,6,7 символы – коды состояния процесса
- 4 символ – признак блокировки
- 2,3 символы – признак переходного процесса
- 1 символ в конце заголовка используется в качестве “знака препинания” для разделения тегов запросов.

Таким образом, всего символов в исходной строке-ключе для вычисления полиномиального хеша равно 16. Число столбцов соответствует числу уровней иерархии обработки и равно восьми. Умножение представляет собой циклический сдвиг на 2 разряда вправо, а деление – сдвиг на 2 разряда влево.

Пользуясь определением хеш-функции, мы можем записать:

$$\text{Context_id}(s_{0..n-1}) = s_L + z s_{L+1} + \dots + z^{R-L-1} s_{R-1} + z^{R-L} s_R = t_X + z t_{X+1} + \dots + z^{Y-X-1} t_{Y-1} + z^{Y-X} t_Y \quad (4.4)$$

Запишем хеш подстроки $s_{0..R}$:

$$\text{Context_ID}(s_{0..R}) = s_0 + z s_1 + \dots + z^{L-1} s_{L-1} + z^L s_L + z^{L+1} s_{L+1} + \dots + z^{R-1} s_{R-1} + z^R s_R \quad (4.5)$$

Разобьем это выражение на две части (подстановки слева и справа симметричны) и вынесем из второй скобки множитель z^L :

$$\text{Context_ID}(s_{0..R}) = (s_0 + z s_1 + \dots + z^{L-1} s_{L-1}) + z^L (s_L + z s_{L+1} + \dots + z^{R-L-1} s_{R-1} + z^{R-L} s_R) \quad (4.6)$$

Выражение в первой скобке есть не что иное, как хеш подстроки $s_{0..L-1}$, а во второй – хеш нужной нам подстроки $s_{L..R}$. Итак, мы получили:

$$\text{Context_ID}(s_{0..R}) = \text{Context_ID}(s_{0..L-1}) + p^L \text{Context_ID}(s_{L..R}) \quad (4.7)$$

Отсюда вытекает следующая формула в виде симметричного кольца вычетов по модулю z :

$$\text{Context_ID}(s_{L..R}) = (1/z^L)(\text{Context_ID}(s_{0..R}) - \text{Context_ID}(s_{0..L-1})) \quad (4.8)$$

Поскольку вычисления осуществляются с точностью до 8 знака, а максимальное значение длины строки $S_{\max} = 16$, то $(1/z^L) = 1/8$, $z = 2$, $L = 3$.

Шаг квантования при выполнении вычислений равен $1/8$. По одному контрольному измерению на каждом уровне иерархии.

Всего уровней 8. Тогда и общее число измерений для каждого процесса равно 8.

Следовательно, задача вычисления дескриптора оценки функции безопасности на основе эталонных значений “трассы прохождения запросов” решается за полиномиальное время, в виде набора восьми переменных в пакетах запросов и восьми интервальных ограничений в цикле обработке для получения результирующей выборки, а также за счет использования только восьми значащих разрядов (а не бесконечно больших чисел) для хранения и обработки результатов.

Результирующая свертка с учетом изменений значащих битов переменных многочлена Жегалкина представляет собой суперпозицию вычислений хеш-функции для i -го процесса. Пусть запущены i -ый, j -ый и k -ый процессы обработки пользовательских запросов гипервизором.

Если существует инвариант, то тогда $R1 = R2 = \text{const}$, $R1 - R2 = 0$. Необходимо решить систему линейных алгебраических уравнений:

$$\left\{ \begin{array}{l} [1 - (1 / 8^n)] * \prod_{n=1}^N (\text{Context_ID}(s_{0..n}) - \text{Context_ID}(s_{0..n-j+1})) = 0 \\ \dots \\ [1 - (1 / 8^n)] * \prod_{n=1}^N (\text{Context_ID}(s_{0..n}) - \text{Context_ID}(s_{0..n-j+1})) = 0 \\ \dots \\ [1 - (1 / 8^n)] * \prod_{n=1}^N \text{Context_ID}(s_{0..n}) - \text{Context_ID}(s_{0..n-k+1})) = 0 \end{array} \right. \quad (4.9)$$

Это означает, что при попадании с одного уровня иерархии N на другой, значение полиномиальной хеш-функции остается константой, а значит, значение самой исходной строки-ключа не модифицируется, как бы ни менялись наборы подстановок или значения значащих битов логических переменных при прохождении пакетов запросов с 1 по 8 уровень.

Это достижимо только в том случае, если применяются сверточное помехоустойчивое кодирование пакетов запросов, исправляются как конфигурации ошибок, одиночные биты, так и группирующиеся ошибки.

Если сверточный (n, k) – код исправляет все конфигурации ошибок и группирующиеся ошибки, кратность которых не превышает значение t , то полученный сверточный $(j \cdot n, j \cdot k)$ – код - перемежения исправляет все конфигурации ошибок и группирующиеся ошибки, кратность которых не превышает значение $j \cdot t$. В нашем случае $n=12$ (длина передаваемых символов в закодированном пакете на выходе), $k = 6$ (число информационных символов в виде контролируемых битов на входе). $t = 8$ (отслеживаются ошибки, кратность которых не превышает 8) .

При $j = 2$ мы получаем сверточный алгоритм перемежения, который исправляет все ошибки в пакетах размером 12 символов (*условие 3*). У нас размер тега, чьи битовые поля служат для хранения логических переменных, участвующих в вычисление хэш-значения, хранимого в Context_ID, равен 12. Для поддержки самокорректирующихся помехоустойчивых кодов необходимо добавить еще одно поле – CR для хранения контрольных битов.

Оценим асимптоматику вычислений. Произведение n значений хеш-функций является конечной величиной. Выражение слева представляет собой убывающую геометрическую прогрессию и равна $8/9$. Тогда результирующая свертка тоже вычислима.

В штатном режиме значение строк Context_ID уникальное и неизменяемое для каждого процесса. Если на каком-то уровне иерархии обработки меняется значение полиномиального хеша, то это сигнализирует о попытке несанкционированного доступа одного процесса к участку памяти «чужого» процесса с целью записать данные или выполнить код, заблокировать или уничтожить информацию, перехватить управления за счет изменения контекста и переменных окружения, скомпрометировать ключ и поменять заголовок в пакете запросов и т.п.

Критерии верификации небезопасных операций, реализованные в методе реактивной защиты [21; 23; 25; 95; 96] , показаны в таблице 4.2.

Таблица 4.2 – Критерии верификации небезопасных операций на структурах Крипке с использованием модальных выражений темпоральной логики CTL

Тип угроз	Критерии верификации небезопасных операций (примеры структур Крипке)	Комментарии
Угрозы НСД до загрузки ОС и во время загрузки ОС	$AG [S7 \Rightarrow AX[A(Ord = False \vee S8) \vee AG \neg L8]]$ $\&\&$ $FG (Start_mon \& Ord = False)$	Между уровня иерархии обработки запросов (команд) S7 и S8 всегда выполняется свойство $Ord = 0$, поскольку на данном уровне иерархии функционирования системы здесь не запускаются любые порожденные процессы, а только инициализирован и работает один базовый процесс – верификатор команд. При включении ЭВМ первым инициализируется и запускается доверенный модуль верификатора команд. С момента запуска (событие $Start_mon$), всегда выполняется условие $Ord=0$ Если не выполняется хотя бы одно из условий, операция блокируется и отправляется оповещение администратору.
Угрозы НСД после загрузки ОС	$If \neg EF (delta > 1) \&\&$ $Diff\{Priv(si), Priv(si+1)\} > 0,$ $delta = \max\{Priv(si),$ $Priv(si+1)\} - Priv(si) = 1;$ $else Diff\{Priv(si), Priv(si+1)\}$ $< 0, to \ delta = Priv(si+1) -$ $\min\{Priv(si) = 1,$ $else \ delta = 0.$	Не существует такого режима работы для квантора пути w, при котором изменение $delta$-параметра превысит значение 1. $delta$-параметр при переходе из si в $si+1$ представляет собой выражение: • если повышение привилегий на траектории вычислений • если понижение привилегий на траектории вычислений • если нет изменений на траектории вычислений в уровне привилегий Верификатор команд реализует функции защиты при обработке запросов контроллером аппаратной транзакционной памяти и при выполнении последовательности команд в исполнительном регионе процессора. Отслеживаются все изменения прав субъектов при выполнении привилегированных команд на уровне процессора и контролера транзакционной памяти в соответствии с требованиями политик безопасности.

Продолжение таблицы 4.2

Угрозы отказа в обслуживании с использованием средств скрытого информационного воздействия, в результате внешних дестабилизирующих воздействий по каналам коммуникации, некорректного выполнения операций, неправильных действий оператора или администратора и т.п.	AG (process_is_on $\Rightarrow$ P Hyper_Permission, delta) && AG (req $\Rightarrow$ (start U ack)) && $\neg$F terminate (AG si & Context_type = 2 & tr = 0 OR tr = 3 B = 1)	В любом режиме, прежде чем запустится новый процесс, то на это обязательно предварительно была получена санкция со стороны гипервизора. Во всех режимах, требующих изменения привилегий для запуска процесса, после того, как запрос req установится, он никогда не будет обработан, пока на него не придет подтверждение. Условие подтверждения – delta = 1. Для всех состояний si на всех траекториях с ошибками или если ресурс заблокирован системой, то запрос принудительно завершается.
Угрозы внедрения вредоносных программ в гостевых и хостовых ОС	G delta (tr & $\neg$tr) – && $\neg$F decrypt (hcode_TCU=False) – && $\neg$F terminate (AG si & Context_type = 2 & tr = 0 OR tr = 3 B = 1)	Всегда будет вычисляться функция delta для тех случаев, когда в системе не возникают ошибки, то есть значения tr = {00, 11}. В этом случае вычисление не проводится, запрос вообще не обрабатывается и удаляется. В остальных случаях, когда Событие 10 – начинается переходный процесс, событие 01 – завершается переходный процесс завершился, проводится контроль на соответствие уровня доступа к запрашиваемым данным с уровнем привилегий инициатора доступа.
Угрозы внедрения по сети вредоносных программ гостевых и хостовых ОС	G delta (tr & $\neg$tr) – && $\neg$F decrypt (hcode_TCU=False) – && $\neg$F terminate (AG si & Context_type = 2 & tr = 0 OR tr = 3 B = 1)	Никогда не проводится вычисление дескриптора оценки безопасности для тех состояний, когда нет активных задач, то есть два старших бита поля TCU (спецификатор hcode_TCU) обнулены. Два младших бита TCU (спецификатор lcode_TCU) кодируют состояние выполнения. Бит В – состояние блокировки (если бит равен 1, то ресурс занят). Для всех состояний si на всех траекториях с ошибками или если ресурс заблокирован системой, то запрос принудительно завершается.

Если сравнивать с существующими на сегодня классическими методами реактивной защиты на основе сигнатурного анализа, методами реактивной защиты на основе поведенческого анализа и эвристическими методами реактивной защиты, то в них не реализованы функции защиты и верификации при обработке запросов контроллером аппаратной транзакционной памяти и при выполнении последовательности команд процессором, поэтому они не способны выявлять и блокировать угрозы ИБ на всех уровнях иерархии субъектов и объектов СК.

Изменения значащих битов в наборах кортежей, содержащихся в тегированных полях, идентифицирует конкретные характеристики при работе с объектами памяти или сетевыми стеками, своего рода «электронные отпечатки пальцев». Но возможны ошибки 1-го и 2-го рода, поэтому необходимо оценить корреляции между параметрами в наборах кортежей, случайное изменение которых может привести к модификации всего пакета запросов.

Также всегда остается возможность успешного подбора такой строки, которая не эквивалентна исходной, но значения их хеш-функций их равны.

Для разрешения подобных коллизий выбраны следующие архитектурные и функциональные особенности:

- Хеш-функция, реализованная в ГОСТ 34.11– 2012, получившая название «Стриборг», обладает высокой криптостойкостью к атаке поиска прообраза и к атаке поиска коллизий по сравнению с алгоритмами предыдущего версий. Это доказано в ходе сертификационных испытаний Центром специальных разработок и защиты информации ФСБ РФ;
- Помимо дескрипторных таблиц гостевых и управляющих ОС, гипервизор сохраняет эталонные значения данных в своей генеративной таблице. Ведется аудит журналов безопасности,

контролируется выполнение операций чтения/записи, изменения полей и строк в системных таблицах. Если вредоносное ПО перехватило управление на уровнях S1-S4, внесло изменения в конфигурацию системы, то монитор безопасности выявить несовпадения при сравнении с ключевыми строками в своей «таблице двойников»;

- Если даже сторонний агент удалось успешно провести атаку по поиску коллизий хэш-функции, модифицировать компоненты гипервизора на уровнях S5-S7, то ему удастся внести запись в системные таблицы по другой ключевой строке, но с тем же хеш-значением. Информация обновится в дескрипторные таблицы ОС и монитора виртуальных машин. Но указанных уровнях все прикладные и системные вызовы перехватываются и эмулируются монитором безопасности. Наряду с программными агентами, в системе есть аппаратный агент – контроллер транзакционной памяти, который напрямую взаимодействует с монитором безопасности гипервизора, функционирующим на уровне S8;
- Монитор безопасности хранит еще один образ конфигурации программно-аппаратных агентов в виде 128-битной матрицы доступа к ячейкам транзакционной памяти. Она хранится в памяти контроллера транзакционной памяти. Запись данных в ПЗУ контроллера возможно, если уровень привилегий равен 8. Прежде чем выполнить код на физических ядрах процессора, он проверяет ключевые строки системных таблиц, сформированных на предыдущих уровнях иерархии обработки, в случае несовпадений блокирует запрос, делает «откат» настроек, не допуская выполнение «ложных» транзакций;

- Стороннему агенту надо успешно модифицировать код монитора безопасности, перехватить управление, поменять информацию во всех дескрипторных таблицах (гостевых и управляющих ОС, генеративной таблице гипервизора, матрице доступа контроллера транзакционной памяти), чтобы не были выявлены несовпадения, скрыть свое присутствие в системе. Для этого ему нужно, как минимум, успешно провести атаки на 3 интерфейсных уровнях.

Синхронизация реализуется с помощью гипервизора с поддержкой расширенных префиксных команд HLE и RTM в качестве прослойки между гостевыми и управляющими ОС и контроллером транзакционной памяти.

На этапе подготовки к запуску задач на СК осуществляется создание снимков запускаемых процессов и сохранение их в генерируемой таблице гипервизора в виде массива 16-символьных строк-ключей для вычисления результирующей сверки хеш-функции. Цикл мониторинга ссылок безопасности представляет собой 8 интервалов наблюдений и 8 контрольных измерений. Шаг квантования кратен $1/8$.

Алгоритм помехоустойчивого кодирования зависит от класса решаемых СК задач, требований к среде передачи данных, конфигурации системы:

1. Автономных суперкомпьютеров, не выполняющих ответственные приложения в интересах национальной безопасности, можно использовать стандартные сверточные коды Ивадаре;
2. Автономных суперкомпьютеров, выполняющих критически важные приложения, целесообразно использовать диффузные сверточные коды Коленбурга-Месси;
3. Для распределенных суперкомпьютеров, использующих высокоскоростные каналы связи (спутниковые, радио – Ethernet, 10-100 Гигабитный проводной Ethernet, Infinband и т.д.), следует применять алгебраические сверточные коды, которые способны исправлять ошибки типа B1, B2, B3.

4.4 Метод реконфигурации среды выполнения программ проактивной защиты, основанный на парадигме темпоральной логики с вычислением доверительная вероятности в качестве показателя защищенности

Метод реконфигурации среды выполнения программ проактивной защиты СК [24; 27; 28; 80; 81] основан на методах темпоральной логики для проверки истинности принимаемых решений на каждой возможной интерпретации (структуре Крипке) с учетом специфики СК в виде временной сложности, связанной с параллельностью вычислений, и пространственной сложности, обусловленной большим количеством ядер процессоров и структурных связей между ними.

Базовый процесс инициализуется значением `context_id` (исходная вершина графа для старта вычислений). Нумеруются все процессы уникальным хеш-значением, который вычисляется для каждого вновь запускаемого приложения. Изменение в поле `Ord` для каждого процесса сигнализирует о появлении уже дочерних подпроцессов в рамках базового процесса и так далее по уровню вложенности.

Хеш вычисляется рекурсивно для всех процессов (подпроцессов) и получается результативное значение для штатного измерения трассы прохождения запроса (на основе прошлого опыта, предобучение). В процессе работы идет обучение системы, каждое новая конфигурация классифицируется и в будущем распознается с учетом характерного набора маркеров.

Кодирование признаков идентификации – набор или конъюнкция простых предикатов, на которых задана формула F_i . Подформула ψ_i представляет собой рекурсивный вызов одной и той же алгоритмической процедуры, только с разным набором характерных признаков и свойств.

Тип контекста определяет режим выполнения операций чтения/записи: работа с сегментами памяти приложений, работа с системной областью памяти устройств, неинициализированные сегменты памяти для подготовки загрузки приложений, ошибки записи в буфер (неверные указатели и переменные окружения, переполнение, утечка памяти и т.д.). Проверка адекватности рассмотренной модели исследования объектов проводилась на основе имитационной модели построения защищенных операционных систем нового поколения. Ядро любой операционной системы – это привилегированный компонент системы, управляющий доступом к аппаратным ресурсам, выделяющий и освобождающий память, создающий потоки и управляющий ими, осуществляющий внутрипроцессную синхронизацию потоков и управляющий вводом-выводом. С учетом специфики бортовых систем под управлением операционных систем реального времени (LynxOS, VxWorks, ОС 2000, ОС 3000), представляющие собой многомодульную архитектуру в виде микроядра и дополнительных расширений для специализированных пользовательских интерфейсов, следует отметить, что нет системных решений, позволяющий создать суперкластер с набором сервисов в виде виртуальных машин, выполняемых на разных уровнях иерархии доменов защиты.

Кроме того, ряд особенностей технологии аппаратной виртуализации ускоряет работу виртуальных машин и повышает уровень безопасности.

Аппаратная поддержка виртуализации и механизм многоуровневой защиты позволяет снизить накладные расходы на создание изолированной среды исполнения и возможность создавать гибридный суперкластер с поддержкой разных аппаратных вычислителей, установленных на стационарных объектах или на борту: от DSP-процессоров (Toshiba, Hitachi и др.), классических процессоров (Intel x86_64) до тегированных процессоров (Эльбрус) и тайловых процессоров (Monarch, Tiler).

В них также не реализованы механизмы многоуровневой защиты от атак при выполнении системных вызовов в среде многоуровневого контекста вложенных гостевых и управляющих операционных систем.

Тегированная архитектура на примере процессора МЦСТ «Эльбрус» не поддерживает технологию аппаратной виртуализации. Однако отсутствие поддержки аппаратной виртуализации делает подобные архитектурные решения узкоспециализированными и не поддерживающим эмуляцию разного оборудования и поддержку широко используемых гипервизоров.

Кроме того, ряд особенностей технологии аппаратной виртуализации ускоряет работу виртуальных машин и повышает уровень безопасности.

Аппаратная поддержка виртуализации и механизм многоуровневой защиты может снизить накладные расходы на создание изолированной среды исполнения.

Однако в любой операционной системе при сопряжении кода ядра с оборудованием на физическом уровне возникают запрещенные состояния – нулевой кортеж данных, к которым процессор запрещает обращаться даже программам в нулевом кольце защиты.

Кроме того, переключение контекста активных задач в защищенном режиме может выполнять только процессорный модуль, поскольку при теновом копировании данных исполняемого кода программист не может получить прямой доступ к информации. Это требует реализации многоуровневой иерархии обработки запросов [21].

Данный подход был неприменим к микропроцессорам предыдущего поколения из-за низкой производительности. Введение дополнительных уровней привилегий и уровней защиты сильно замедляло работу системы в целом. Высокая производительность СК, наоборот, позволяет быстро анализировать дескрипторные таблицы, вычислять хеш-значения процессов.

Их отличает: самодиагностика, многоуровневая защита, привязка каждого тредового устройства к определенному домену, программирование с использованием нефункциональных, непроцедурных языков – цепочки вычислений в виде селекторов, подстановок справа, слева функциональных вычислений, многоуровневое распараллеливание алгоритмов и т.д.).

При выполнении программы загрузки тредовое устройство работает в особом режиме привилегий – IPL_LEVEL. В этом режиме используется физическая адресация при доступе к памяти команд и памяти данных. Затем загружаются диспетчер работы с виртуальной инфраструктурой и хостовые ОС с поддержкой гипервизоров.

При этом для модулей ядра используется уровень KERNEL_LEVEL, для диспетчера работ с оборудованием – SUPERVISOR_LEVEL. На заключительном этапе осуществляется запуск гостевых ОС на уровне USER_LEVEL. Единственным «камнем преткновения» в многоядерных многопроцессорных системах являются задержки проблема эффективной реализации внутрикристальной сети и работы с памятью. Многоядерность микропроцессоров, применяемых в вычислительных узлах, стала вынужденной мерой для обеспечения роста их пиковой производительности, и вызвана она прекращением прямого влияния закона Мура на рост производительности процессорных ядер. Однако многоядерность породила проблему эффективной реализации внутрикристальной сети и усугубила проблемы работы с памятью.

Возможными решениями могут быть применение в процессорных ядрах мультитредовых и потоковых архитектур.

Мультитредовая организация позволяет одновременно выполнять не один, а несколько потоков команд, что дает возможность увеличить множество выполняемых команд, но важнее – усилить поток одновременно выполняемых операций с памятью.

Потоковая архитектура предполагает применение решающих полей элементарных процессоров в виде статических графов потоков данных. Это позволяет сократить общее количество обращений к памяти, поскольку на решающем поле данные передаются с одного быстрого ресурса на другой без обращения в память.

Мы предлагаем метод реконфигурации среды выполнения программ проактивной защиты СК с учетом требований мобильности и обеспечения удельных характеристик производительности программы для безопасного расширения функциональности системы или приложения.

Только аппаратная поддержка сегментированных стеков может снизить сложность компилятора и накладные расходы на этот механизм во время исполнения.

В любой операционной системе при сопряжении кода ядра с оборудованием на физическом уровне возникают запрещенные состояния – нулевой кортеж данных, к которым процессор запрещает обращаться даже программам в нулевом кольце защиты.

Более того, переключение контекста активных задач в защищенном режиме может выполнять только процессорный модуль, поскольку при теновом копировании данных исполняемого кода программист не может получить прямой доступ к информации. Данные коллизии разрешает новый подход – маркерное сканирование, в котором используются генеративные таблицы.

Помимо кодирования адресного разделения колец защиты памяти, в ОС разных классов реализован строго типизированный интерфейс сопряжения с аппаратным ядром процессора и управления.

Контекстом исполнения бинарного кода с учетом профиля компиляции и сборки – использование маркеров системных объектов.

Как уже упоминалось ранее, параметр `Context_type` однозначно определяет локальное пространство исполнительного слоя ядра (`Context_type >= 0`) или адреса верхнего регистра системной памяти, выделенной для критически важных устройств ОС (`Context_type < 0`).

В «теновом» регистре содержатся «снимки» активных процессов-поточков ядра. Таким образом, просматривая значения идентификаторов `Context_ID`, можно получить информации обо всех процессах-потоках, что делает ОС гибкой и «прозрачной» для программистов и вирусных аналитиков. Внутренний цикл работы процессора на аппаратном уровне определяется временными интервалами (T – периоды синхронизации). При выполнении одной инструкции другие запросы блокируются на время выполнения задачи ядра. Мультиплексирование каналов связи с внешним оборудованием позволяет создавать очередь параллельной обработки (конвейер). На уровне декодирования идет преобразование $2*2*2$ в последовательную выборку разрядов, т.е. $2+2+2 = 6$, тогда время выполнения команды ($T_0 - T_5$) занимает максимально 3-4 такта, 1 такт на контроль выполнения и 1 такт для подачи сигнала стробирования, чтобы открыть доступ к кристаллу памяти процессора. По сути, набором портов является минимальный набор файлов, который указывает вашей системе, как корректно откомпилировать и установить программу с учетом расширения специфической исполняемой среды с поддержкой среды. Набор для интеграции с решения с промышленными гипервизорами включает:

- Файл *smw-test.cfg*

Подсистема отказоустойчивости является автономной функциональной структурой, то есть осуществляет функционирование от ВКС и имеет собственный контур управления (осуществляется системным оператором или сервисным инженером). Все параметры по разворачиванию и настройке сети сохраняются в конфигурационном файле *smw-test.cfg*;

- Файл *OS-param*. Этот файл содержит информацию о файлах, которые должны существовать на вашей системе, и их контрольные суммы,

для проверки того, что файлы в процессе загрузки не были повреждены, настройки операционной системы – режим загрузки, настройки параметров памяти на разных интерфейсных уровнях привилегий, путь к исполняемому образу ОС (Single Image Distributive). В файле существует 4 секции для настройки модулей на уровнях IPL_LEVEL, KERNEL_LEVEL, SUPERVISOR_LEVEL, USER_LEVEL, которые соответствуют Dom_id = 00, Dom_id= 01, Dom_id= 10, Dom_id = 11;

- Загружаемый образ ОС (Single Image Distributive) – серверная ОС представляет собой Unix –подобную систему с поддержкой XEN или KVM, VMWare ESXi или Windows Server с поддержкой Hyper-V. Под управлением указанных гипервизоров запускаются гостевые ОС виртуальных машин пользователей;
- Каталог OS-files. Этот дополнительный каталог (базовый дистрибутив со всеми папками входит в состав загрузочного образа) содержит самые «свежие» патчи, использование которых необходимо для обновления версии ОС. Патчи – это программные модули обновлений, в которых содержатся изменения, которые нужно внести в операционную систему с целью корректного функционирования и разрешения предшествующих ошибок в программном коде;
- Файл OS-plist. Это список всех файлов, которые будут загружены. В нем также содержатся указания системе портов на удаление определенных файлов во время реконфигурирования ОС. Данный файл используется в целях протоколирования и аудита безопасности.

Проверка на наличие уязвимостей должна быть выполнена перед установкой новых модулей. Проверка безопасности и обновление базы данных должны выполняться при повседневной проверке безопасности системы. С этой целью все операции по настройке ОС осуществляются только администратором системы, разделяются share и root- директории.

Все операции через консоль инсталляции ОС осуществляются с использованием `private_key` (секретных ключей) по протоколам зашифрованного обмена. Образ серверной ОС загружается на мастер-узле (0 узел среди вычислительных узлов). Рассылка образа ОС на другие вычислительные узлы через высокоскоростную коммутационную сеть осуществляется на основе файла `smw-test.cfg` посредством сетевой загрузки.

Некоторые командные процессоры для ускорения поиска выполнимых файлов и команд кэшируют имена программ, доступных для вызова из каталогов, перечисленных в переменные окружения `PATH`. Для обмена файлами по протоколу NFS система портов использует утилиту `fetch`, которая принимает во внимание различные переменные окружения.

Пользователям сервисных узлов, которые подключены к сети Ethernet, могут взаимодействовать с сервисными узлами высокоскоростной коммутационной сети с помощью дополнительного сетевого моста.

При этом команда для взаимодействия по протоколу NFS - `make fetch`. Просто запустите эту команду в каталоге самого верхнего уровня (`/usr/ports`), и требуемые файлы будут загружены. Эта команда будет работать также и с вложенными категориями, например: `/usr/ports/net`.

Заметьте, что если порт имеет зависимости от библиотек или других портов, то он не будет также загружать дистрибутивные файлы этих портов. Программа загрузки выполняется в каждом ядре, выполнение осуществляет тредовое устройство с нулевым физическим адресом.

При выполнении программы `boot` тредовое устройство работает в особом режиме привилегий – `IPL_LEVEL`. В этом режиме используется физическая адресация при доступе к памяти команд и памяти данных.

На данном уровне функционирует только модуль доверенной загрузки и защиты «Альфа-монитор». Затем загружаются диспетчер работы с виртуальной инфраструктурой и хостовые ОС с поддержкой гипервизоров (уровни `KERNEL_LEVEL`, `SUPERVISOR_LEVEL`). На заключительном этапе осуществляется запуск гостевых ОС на уровне `USER_LEVEL`.

Контекстно-зависимые переходы формализуются в виде кортежа. В каждом домене защиты формализуются клиент-серверное взаимодействие, абстрагируясь от аппаратурной специфики в виде логической функции (3.5), приведенной в третьей главе диссертационного исследования. Систему можно разделить на несколько подсистем, каждая из которых выполняет управление определенным видом ресурсов (памятью, задачами, файлами, средствами коммуникаций). Менеджеры более высокого уровня используют средства модулей низкого уровня своей подсистемы, а также и других подсистем. Микроядерная распределенная архитектура является альтернативой классическому способу построения операционной системы.

Под классической архитектурой в данном случае понимается рассмотренная выше структурная организация ОС, в соответствии с которой все основные функции операционной системы, составляющие многослойное ядро, выполняются в привилегированном режиме.

Суть микроядерной архитектуры состоит в следующем. В привилегированном режиме остается работать только очень небольшая часть ОС, называемая микроядром. Микроядро защищено от остальных частей ОС и приложений. Уровень, позволяющий управлять защищенным режимом процессора, осуществлять загрузку и контролировать работу системы на уровне инструкций процессора называется IPL (Instruction Processor Level, IPL). Набор функций микроядра обычно соответствует функциям слоя базовых механизмов обычного ядра (Kernel-level).

Менеджеры ресурсов, являющиеся неотъемлемыми частями обычного ядра – файловая система, подсистемы управления виртуальной памятью и процессами, менеджер безопасности, – становятся «периферийными» модулями, работающими в режиме супервизора (Supervisor-Level) [23].

В отличие от традиционного конвейера обработки запросов, где передается только сигнатуры операций, в программно-техническом решении «Альфа-монитор» формируются генеративные таблицы на этапе подготовки задачи и хранятся в специальных тредовых регистрах `t_reg`.

Они представляют непроцедурную нефункциональную, а семантическую часть кодирования цепочки клиентских запросов. После чего передается стандартная процедурная составляющая – опкод инструкций (набор кода данных и кода команд, модификаторов и флагов условных переходов).

Модуль доверенной загрузки и защиты загружается с уровнем привилегий IPL и осуществляет монопольный контроль выделения вычислительных ресурсов сервера, просматривая данные из t_reg. Используется 4 кольца защиты процессора: Dom_id = 00, Dom_id= 01, Dom_id= 10, Dom_id = 11.

С учетом поддержки аппаратной виртуализации «склейка» параметров Dom_id и s (строки записываются в стек в обратном порядке) позволяет задать тот интерфейсный уровень, на котором обрабатывается транзакция по схеме «клиент-серверное взаимодействие». На основе декомпозиции выделяется 8-уровней функционирования компонентов гипервизора **modi**. Заданы 4 непривилегированных уровня – наборы 111, 110, 101, 100, 4 привилегированных уровня – 011, 010, 001, 000.

Контекст изменения операций - изменения значений значащих разрядов в кортежах логических переменных при "проходе запроса с 1 по 8 уровень иерархии.

Бальные оценки безопасности предлагаем кодировать числом, представляющим собой значение перевода из двоичной системы в десятичные изменения значащих разрядов набора кортежей логических переменных, содержащихся в префиксе фрейма запросов.

При $n = 8$ подсчета цикл регистрации статистических параметров состоит из 8 контрольных измерений. Цикл состоит из восьми уровней иерархии обработки запросов.

Тогда одно измерение количества инструкций приходится на один такт процессора, а вся процедура для вычисления усредненных взвешенных параметров за K циклов составляет $K * 8$. H – величина энтропии системы и представляет собой отношение суммы инструкций до трансляции в набор директив и без переключения в S-режим к сумме инструкций после преобразования в микрокоманды с изменением контекста выполняемых операций:

$$H = \sum n_0 / \sum n_m \leq 8, i = \{0, 1, 2, 3, 5, 6, 7, 8\} \quad (4.10)$$

Если H равно 8, то в системе нет присутствия сторонних агентов. Если $H < 8$, то в коде присутствуют inline-функции для работы со стеком, ассемблерные вставки, функции заглушки обработки прерываний в виде пустых операций NOP и т.п., которые себя никак не компрометируют, но на более низком интерфейсном уровне они транслируются в набор результативных директив.

Скрываемый модуль также может менять показания счетчика тактов не на постоянную величину, а на случайную, так, чтобы, с одной стороны, уравнивать средние длительности выполнения трассы, а с другой – стабилизировать разброс длительностей выполнения трассы. С учетом модели операций логическая функция оценки безопасных состояний задана на множестве кортежей в виде конъюнкции восьми переменных $x_1x_2x_3x_4x_5x_6x_7x_8$.

В результате значение характеристической функции $f(x) = 1 \mid p = 0,88888889$; $\Psi(n) = 8$, где $f(x)$ – характеристическая функция представляет собой полином Жегалкина 8 степени:

$$F(x) = x_1 + x_1x_2 + x_1x_2x_3 \dots + x_1x_2x_3x_4x_5x_6x_7x_8 \quad (4.11)$$

Таблица 4.3 – Вид ДНФ для разного типа конфигураций СК

Тип конфигурации и показатель защищенности	Вид ДНФ
4 тип (до 4 доменов защиты) Бортовые супер-ЭВМ	$F(x) = x_1 + x_1x_2 + x_1x_2x_3 \dots + x_1x_2x_3x_4$
1 тип (до 5 доменов защиты) Стационарные суперкомпьютеры	$F(x) = x_1 + x_1x_2 + x_1x_2x_3 \dots + x_1x_2x_3x_4x_5$
2 тип (до 7 доменов защиты) Стационарные суперкомпьютеры	$F(x) = x_1 + x_1x_2 + x_1x_2x_3 \dots + x_1x_2x_3x_4x_5x_6x_7$
3 тип (8 доменов защиты) Стационарные суперкомпьютеры	$F(x) = x_1 + x_1x_2 + x_1x_2x_3 \dots + x_1x_2x_3x_4x_5x_6x_7x_8$

Примечание. Мы говорим о вычислении с точностью до 8-го знака в том случае, если используем стационарные СК. В случае применения бортовых Супер-ЭВМ, то для ускорения работы и оптимизации вычислений целесообразно пренебречь погрешностью в пользу высокой производительности в режиме реального времени. Например, число 0,88888889 представляет собой приближенно периодическую дробь $0,(8)_n$.

Тогда мы можем проводить вычисления для мобильных систем с точностью до 1-го разряда после запятой, в зависимости от типа архитектуры процессора варьируется и число используемых интерфейсов N, влияющих на мощность пространства вычислений (таблица 4.3). Профиль защиты в зависимости от типа суперкомпьютеров и их конфигурации приведен в таблице 4.4.

Таблица 4.4 – Профиль защиты и набор конфигураций для разного класса СК

Класс суперкомпьютеров	Доверительная вероятность уровня защищенности по формуле Корнфельда p	Вероятность проведения успешной атаки $q = (1 - p)$	Максимальное кол-во доменов защиты	Максимальное кол-во уровней иерархии (маркеры привилегий)	Максимальное кол-во интервальных ограничений фильтрации верификатора команд
Стационарные суперкомпьютеры: 1 тип – малый кластер	0,8 - 0,83	0,2 - 0,17	4 или 5 (средний уровень защиты) 4 – для коммерческого малого кластера для сверхбыстрой обработки данных 5 – для малого промышленного кластера с повышенными требованиями по отказоустойчивости и защищенности	6	6
Стационарные суперкомпьютеры: 2 тип – средний кластер	0,857 - 0,875	0,143 - 0,12	6 или 7 (высокий уровень защиты) 6 – для среднего коммерческого суперкластера без специальных требований по отказоустойчивости и надежности 7 – для суперкластера крупного наукоемкого технологического объекта, для крупных информационно-аналитических центров	7	6

Продолжение таблицы 4.4

Стационарные суперкомпьютеры: 3 тип: стратегический класс	0,9	0,1	8 (максимальный уровень защиты) для масштабных суперкомпьютеров Росатома, разведывательных стационарных центров Минобороны (уровня бригады, Генерального штаба и управлений родов войск), специальных вычислительных центров СОПМ МВД, Росгвардии, ФСБ, ФСО, СВР и др.	8	7
Бортовые Супер- ЭВМ	0,75 - 0,8	0,2 - 0,25	3 или 4 (базовый уровень защиты) 3 – Для серийных многоплатных бортовых вычислителей, объединенных в локальную сеть и с поддержкой RadioEthernet, Bluetooth, Wi-Fi и др. 4 – Для компактных высокопроизводительных комплексов с высокими требованиями ГОСТ РВ по отказоустойчивости и защищенности военного назначения (специализированная сборка)	4 или 5 4 – если используется только локальное подключение (проводное Ethernet или Infiniband) 5 – если бортовые вычислители имеют подключение к сетям RadioEthernet, Bluetooth, Wi- Fi, инфракрасные порты	4 или 5
			2 (минимальный уровень защиты) для автономных одноплатных портативных бортовых вычислителей (мини-сборка)	3	3

Таблица 4.5 – Сравнение методов проактивной защиты

Название метода	Число доменов в защите	Параметр учета переходности	Признак системности	Признак блокировки операций	Контроль контекста и тайминга выполнения операций
Авторский метод проактивной защиты Показатель гарантированной защищенности P = 0,9	8	Да	Контроль со стороны верификатора команд выполнения операций процессором и контролером транзакционной памяти	Да	Реализованы расширенные функции идентификации и верификации
Эволюционный метод проактивной защиты на основе генетических алгоритмов Показатель гарантированной защищенности P = 0,5	2	нет	нет	нет	нет
Метод проактивной защиты на основе прогнозирования в виде экстраполяции данных Показатель гарантированной защищенности P = 0,5	2	нет	нет	нет	нет
Метод проактивной защиты на основе скрытых марковских цепей Показатель гарантированной защищенности P = 0,6	3	нет	только регистрация изменения системного бита	нет	нет

В таблице 4.6 приведено сравнение методов проактивной защиты. **Ни один из существующих методов проактивной защиты не адаптирован для решения задач идентификации и блокирования угроз ИБ с учетом специфики суперкомпьютеров. Наилучшие показатели защищенности, количество реализованных функций безопасности демонстрирует авторский метод.** Наряду с проверкой целостности в виде изменения значений вычисленных хеш-функций для каждого процесса, реализован контроль доступа в виде маркировки разрешенных переходных процессов при переключениях между доменами защиты и миграции потоков. Ведется история разных конфигураций системы, распознаются новые события за счет вариантного нового набора предикатов и значений их траекторий на структурах Крипке.

Перейдем непосредственно к построению взвешенного мультиграфа-модели выполнения операций СК, представляющего собой дерево. Стартовая точка – это вершина древовидной структуры, описывает контроллер транзакционной памяти (ТП), который напрямую взаимодействует с модулем верификатора гипервизора. Контроллер ТП в связке с верификатором реализует механизмы защиты: изолированное мультидоменное адресное пространство представляется в виде неперекрывающихся областей памяти, каждой из которых соответствует вершина графа, лежащая во втором уровне корневой структуры. Вершины третьего уровня соответствуют подмножествам компонентов, из которых состоит гипервизор. Поскольку у нас 8 уровней иерархии обработки запросов, то блок разбит на восемь подмножеств детализации уровней возможных состояний, обозначенных $S_8, \dots, S_1$.

Последний уровень корневой структуры мультиграфа представлен агентами гипервизора (листья графа), через которые происходит связь с внешней средой и все возможные атаки на СК. Продвижения на каждый следующий уровень – это более высокий уровень абстракции описания обработки запросов, переход от нижнего уровня спецификации к верхнему.

Каждое состояние может иметь не одну, а множество цепочек – продолжений, и является корнем своего дерева историй (вычислений).

Метод реактивной защиты включает в себя алгоритмы *вычисления дескрипторов оценки безопасности состояний* и идентификации угроз на основе *маркировки состояний* выполнения запросов СК.

Таблица 4.3 – Уровень привилегий и режим обработки транзакций

Признак привилегированности	Уровень привилегий Priv(si)	Идентификатор домена dom_id	Бит S
Y Для всех процессов в доменах Индекс Context_type =1	7	00	0
	6	01	0
	5	10	0
	4	11	0
N Для всех процессов в доменах Индекс Context_type =3	3	00	1
	2	01	1
	1	10	1
	0	11	1

Context_type - индекс	Режим работы
0	Ожидание транзакций(инициализация)
1	Операции чтения/записи в память приложений
2	Ошибка записи в буфер данных
3	Обращение к привилегированной памяти устройств

При использовании стационарных и бортовых СК с общей памятью не обойтись без синхронизации доступа параллельных процессов к общему ресурсу (памяти). Последние аппаратные реализации транзакционной памяти осуществляется с помощью префиксов, модифицирующих исполнение инструкций.

Вводятся специальные команды XBEGIN, XEND, XABORT – новые расширения, которые схожи по мнемонике с командами SYSETER, SYSEXIT, VXMON, служащими для поддержки аппаратной виртуализации. Изоляция участков памяти реализуется на основе нескольких колец защиты.

Реализация запасных функций для распараллеливания предполагает выполнение процессов в разных режимах работы процессора, чтобы корректно разрешать ситуации с одновременным обращением процессов к одному и тому же участку памяти. Использование многоуровневой виртуальной “песочницы” полностью исключает конфликты при обращении к памяти параллельных процессов и следующие недостатки:

- ограниченность применения – в транзакции нельзя выполнять операции, действие от которых невозможно отменить. Эту проблему решает модуль верификации, который функционирует на уровне S8. Прежде чем выполнить операции, он эмулирует, проверяет корректность запроса, а только потом передает на выполнение физическим ядрам процессора;
- сложность отладки – поставить точку прерывания внутри транзакции невозможно. Если мы управляем режимами работы процессора с помощью аппаратного гипервизора, а не напрямую сразу выполняем транзакцию, то можно проводить отладку за счет эмуляции. Фактически мы работаем в виртуальной среде с 8 режимами эмуляции, переключаясь между которыми, можно безопасно распараллеливать выполнение процессов, не боясь, что они будут конфликтовать;
- синхронизация процессов и исключение обращение к одному участку памяти. Процессы в разных кольцах защиты физически разделены. Они не могут обращаться к одному и тому же участку памяти, модифицировать одни и те же данные. Используются 4 кольца защиты. С учетом эмуляции в гипервизоре работы с физическим оборудованием число уровней иерархии становится 8.

- Параллельные процессы не могут «расплываться» по всем участкам памяти. Они привязаны к определенному кольцу защиты, контролируется контекст выполнения операций запросов на всех уровнях иерархии, выполняются в определенном режиме работы процессора и обращаются только к выделенным им сегментам памяти данных и программ, а не случайно перемещаются по всей оперативной памяти;
- без поддержки аппаратной технологии виртуализации невозможно было реализовать эмуляцию разного типа процессоров.

4.5 Выводы по разделу:

1. Полное решение проблемы защиты – разработка суперкомпьютеров с аппаратно обеспечиваемыми уровнями защиты программ и данных. В настоящее время реализуются четыре уровня защиты (пользовательский, исполнитель ОС, ядро ОС, программа начальной загрузки);

2. Разработан метод реактивной защиты (без запаздывания после обнаружения атаки), заключающийся в виртуализации среды исполнения процессов СК, если вычисленный дескриптор состояния попадает в зону «риска», и основанный на контроле запросов на выделение ресурсов в соответствие с оценкой безопасности и статистических характеристик выполнения трасс команд;

3. Разработан метод реконфигурации среды выполнения программ проактивной защиты СК с учетом требований мобильности и обеспечения удельных характеристик производительности программы на основе применения дескрипторов безопасности состояний и интервально-временных ограничений с введением 8-уровневой иерархии обработки запросов;

4. При разработке метода реконфигурации среды выполнения программ проактивной защиты суперкомпьютерных систем впервые предложена сборка команд непосредственного сценария выполнения исходящих директив. Для решения разного класса задач могут быть

адаптированы разные типы вычислителей, наборы шаблонов и конфигураций с подборкой лучших показателей по защищенности, производительности и энергоемкости;

5. Предложена и запатентована технология, позволяющая за счет использования резервов высокой производительности, принципиально новых архитектурных и функциональных доработок суперкомпьютеров создать защищённый мультидоменный гипервизор, который в связке с аппаратурой, прикладным и системным ПО обеспечивает надёжную защиту от атак, связанных с высокой асинхронностью процессов суперкомпьютеров, и возможностью выполнения ложных транзакций в результате дестабилизирующих внешних воздействий, что имеет наиболее важное значение для АС разного класса;

6. Снижены стоимость и продолжительность разработок для отрасли промышленности за счёт переноса большей части испытаний с опытных образцов на программное обеспечение. Поддерживается весь цикл разработки, тестирования, а также верификации прикладного и системного ПО при проведении спец. исследований в многоуровневой “песочнице” с контролем контекста и тайминга выполнения запросов (операций).

7. Авторский многокомпонентный защищенный мультидоменный гипервизор поддерживает как зарубежные коммерческие кластерные платформы, так и специализированные отечественные комплексы.

ГЛАВА 5. РАЗРАБОТКА ПРОГРАММНО-ТЕХНИЧЕСКИХ РЕШЕНИЙ ПО ПОВЫШЕНИЮ УРОВНЯ ЗАЩИЩЕННОСТИ СОВРЕМЕННЫХ СУПЕРКОМПЬЮТЕРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ В ВИДЕ МНОГОМОДУЛЬНОГО ЗАЩИЩЕННОГО МУЛЬТИДОМЕННОГО ГИПЕРВИЗОРА

5.1 Краткое описание экспериментального стенда

Разработчикам и пользователям, с одной стороны, необходимо наглядно показать эффективность и «прозрачность» механизмов защиты как на верхнем, так и на нижнем и среднем уровнях интерфейсных взаимодействий модулей, то есть пользователи и разработчики должны иметь ясные представления о функциональной реализации, четко ставить и решать задачи по повышению уровня защищенности на основе прототипа и методики по использованию и разработки альтернативных механизмов защиты в коммерческих и государственных вычислительных системах.

Для разработки макета со встраиванием дополнительных модулей защиты с целью противодействия скрытым угрозам необходимо предварительно выбрать аппаратно-программную платформу, позволяющую провести апробацию результатов диссертационного исследования в рамках промышленного масштаба.

В основе построения экспериментального стенда лежит подход объединения ресурсов для вычисления и хранения на каждом узле системы.

Каждый узел системы является как частью вычислительной подсистемы, так и частью распределенной системы хранения данных пользователей. Предусмотрены также автоматизированные рабочие места операторов (пользователей), исполняя сразу два вида нагрузки. Для высокоскоростной передачи данных между вычислительными узлами используется технология коммутации Mellanox InfiniBand FDR,

обеспечивающая скорость неблокируемой коммутации до 56 Гбит/с на каждый порт. Для повышения скорости доступа вычислительных узлов к сети до 100 Гбит/с в каждый сервер СК установлен адаптер с двумя портами InfiniBand FDR. Это обеспечивает необходимую производительность для ресурсоемких приложений пользователей и для распределенной системы хранения суперкомпьютера.

В составе стенда были как модули с физическими кластерами, так и сегменты с использованием open-source гипервизоров Kvm, Xen и LXC. Проприетарные гипервизоры (например, Microsoft Hyper-V и VMWare ESXi) мы не рассматриваем, поскольку указанные иностранные компании в условиях санкций ушли с российского рынка. Более того, Microsoft и VMWare не предоставляют исходники своего ПО, что делает затруднительным их использование на объектах КИИ.

Оценка результатов успешного обнаружения сторонних программных агентов проводилась на испытательном стенде, который объединяет 50 сервисных узлов и узлов подсистемы хранения данных, 2 сервера управления, 30 вычислительных узлов и 20 рабочих станций, и представляет собой виртуальную защищенную сеть.

Схема экспериментального стенда и набор инструментального ПО приведены в Приложении 3.

В качестве средства для исследования исполняемых файлов с применением технологии реверс-инжиниринга был выбран дизассемблер IDA Pro фирмы Datarescue [35]. IDA является интерактивным дизассемблером и представляет собой среду дизассемблирования, обладающую развитым пользовательским интерфейсом.

Кроме того, IDA поддерживает специальный язык скриптов – IDA C, имеющий C – подобный синтаксис и широкий набор встроенных функций для обращения к автоматическому анализатору и интерактивной оболочке.

Архитектура IDA также позволяет разрабатывать дополнительные модули, использующие Disassembler Engine и существенно облегчающие рутинные операции анализа. Кроме того, IDA может сгенерировать текстовый файл листинга или ассемблерный текст, пригодный для повторного ассемблирования. IDA имеет мощный анализатор кода, который автоматически идентифицирует функции, глобальные переменные, аргументы и локальные переменные функций, генерирует перекрестные ссылки на код и данные. Уникальной особенностью IDA является возможность распознавания библиотечных функций различных компиляторов, что значительно облегчает анализ программы. IDA поддерживает отладочную информацию, а также содержит встроенный отладчик.

5.2 Повышение уровня безопасности вычислительных и сервисных узлов суперкомпьютеров при использовании традиционных средств защиты за счет реализации методики тестирования с использованием средств виртуализации

В рамках инженерных экспериментальных исследований был разработан многокомпонентный защищенный мультидоменный гипервизор в нескольких вариантах исполнений: программном и программно-аппаратном [32; 36; 93]. Программный модуль представляет собой монитор безопасности и гипервизорные модули под названием «Альфа-монитор», который непосредственно взаимодействует с физическим (виртуальным) оборудованием на уровне гипервизора и получает управление на этапе инициализации, до загрузки хостовых и гостевых ОС серверов сервисных и вычислительных узлов.

Гипервизор в составе авторского программно-технического комплекса «Альфа-монитор» состоит из проксирующих модулей, верификатора команд и монитора безопасности.

Верификатор комнд устанавливается на всех физических узлах суперкомпьютера на BMC-контроллер (контролер материнской платы) в виде “прошивки” программного модуля управления защищенным режимом процессорных блоков, который реализует интерфейсные низкоуровневые функции при работе с модулями аппаратной транзакционной памяти.

Проксирующие модули и монитор безопасности устанавливаются на хостовых и гостевых ОС вычислительных и сервисных узлов суперкомпьютера (см. рисунок 5.1).

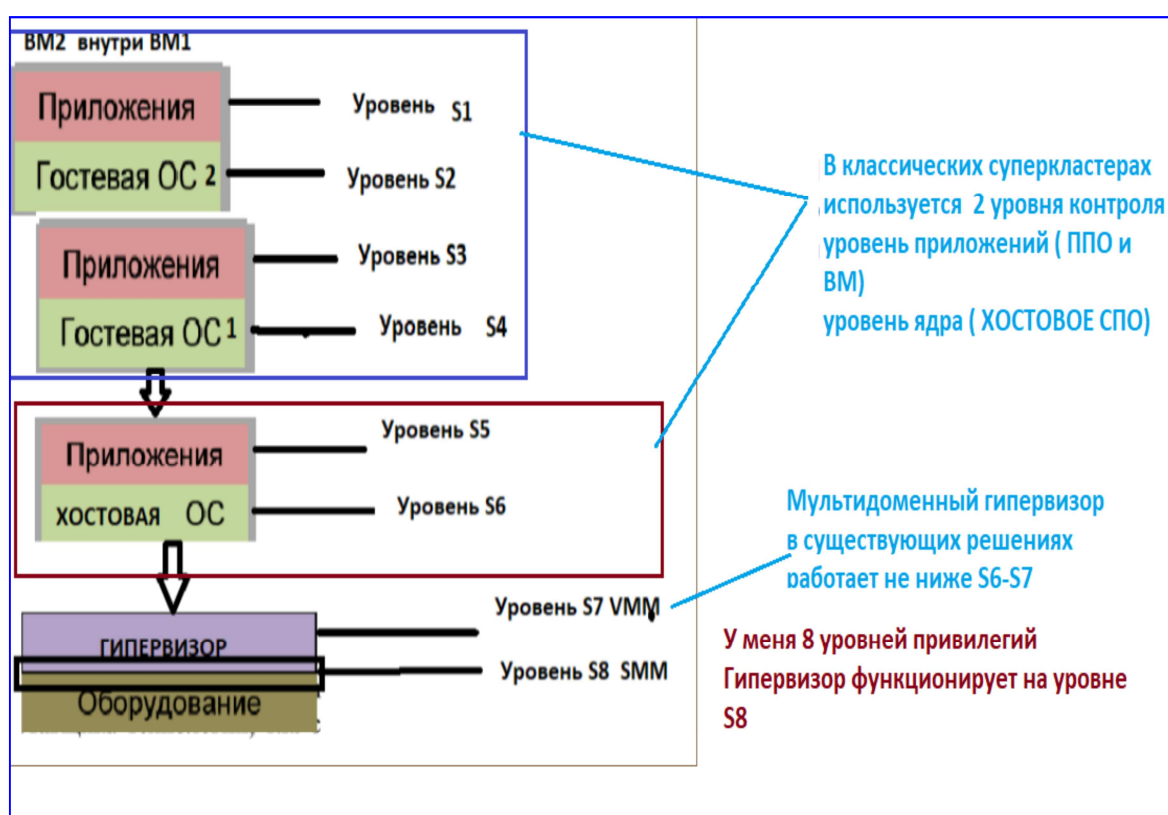


Рисунок 5.1 – Интерфейсные уровни функционирования компонентов многомодульного защищенного гипервизора

Каждый запрос пользователей суперкомпьютерной системы формально описывается в виде кортежа Opcode, Dom_id, s, Ord, Context_type, Context_id, tcu, tr.

На рисунке 5.2 иллюстрируется скрытый механизм несанкционированного повышения полномочий доступа к объектам ядра. В литературе, относящейся к предметной области исследований ОС [4; 5; 6; 7; 8; 9; 10; 13; 14; 15; 16; 17; 18], описываются механизмы безопасности менеджера объектов (reference monitor), подсистемы ввода-вывода нижнего уровня, но не учитываются функциональные возможности ядра гостевой и хостовой ОС, а также диспетчера оборудования.

Методика тестирования уровня защищенности предполагает вычисление результирующей свертки на основе атрибутов протекающих процессов, конфигурации среды выполнения с заданной глубиной (масштабируемостью). Ошибки 1-го и 2-го рода могут возникать на локальном i -ом уровне, но они детектируются и исправляются на уровне системной задача анализа контроля целостности поступающей информации с датчиков со всех уровней иерархии обработки запросов.

Весовой коэффициент защиты определяется как разность между исходным (эталонным) значением хеша и полученным после прохождения пакета запросов и вычисления свертки. Для количественной оценки величины возмущений будем использовать относительный показатель ϵ : $(S_i - S_a) / S_a$, где S_i – состояние до атаки, S_a – состояние после атаки. Этот параметр используется в качестве корректирующего параметра для обучения системы на каждом уровне (слое) иерархии обработки запросов.

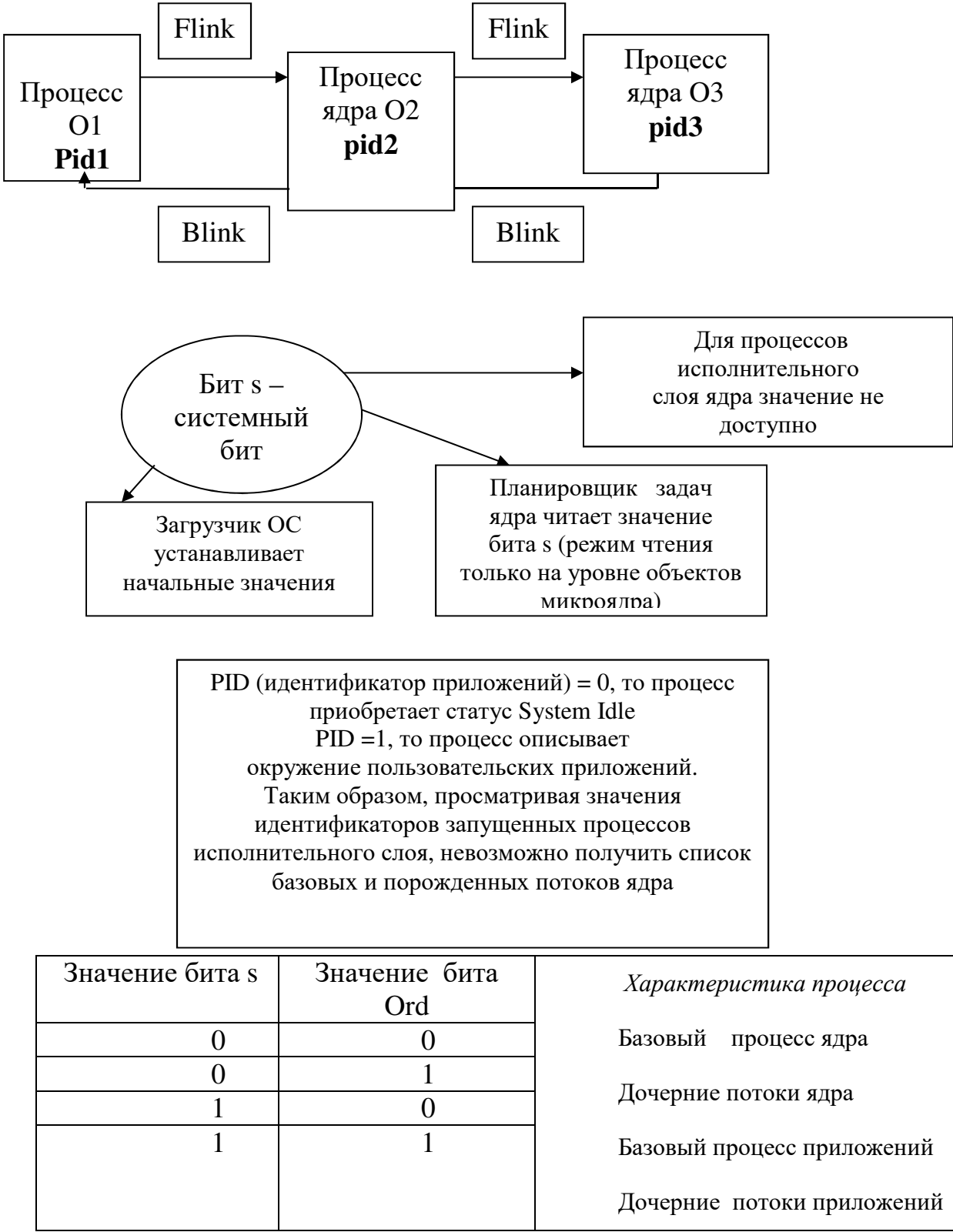


Рисунок 5.2 – Кодирование процессов-потоков

Планировщик задач осуществляет выделение квантов времени каждому потоку и оптимизирует выполнение кода на разных виртуальных или физических процессорных модулях. Однако он не управляет переключением контекста системных процессов-потоков. Аппаратная поддержка режимов синхронизации, операции «теневого» копирования реализуется на уровне исполнительного региона процессора. Логика данного функционала реализуется в виде макроса $\max(\text{Mod1}, \text{Mod2})$. Если $s = 0$, $\min(\text{Mod1}, \text{Mod2})$, если $s = 1$, где Mod1 – привилегии доступа базового потока ядра, получившего управление, Mod2 – привилегии доступа поток ядра, ожидающего выполнения. Выражение $\max(\text{Mod1}, \text{Mod2})$ означает, что во время переключения системного контекста потоков процессор повышает уровень полномочий до максимального уровня привилегий, если бит $s = 0$, понижает до минимального уровня полномочий, если бит $s = 1$.

Значит, можно реализовать скрытый канал управления информационными потоками ядра и при этом реализуется принцип «невидимости» (см. рисунок 5.3).

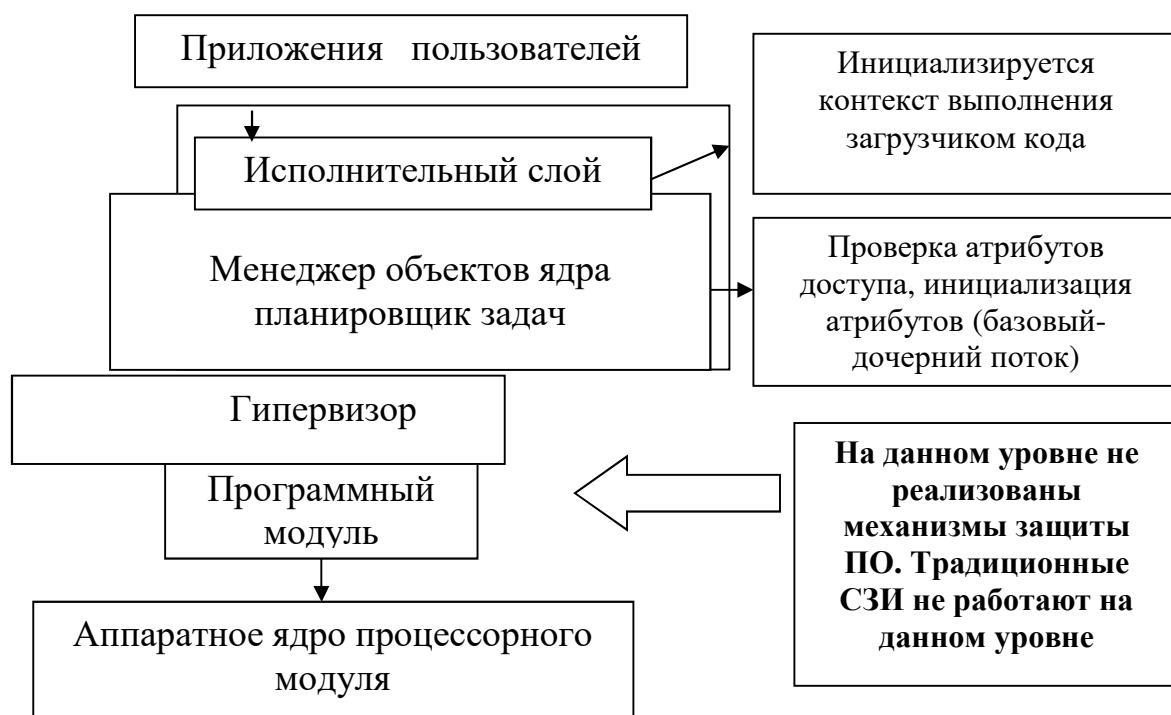


Рисунок 5.3 – Пример реализации принципа «невидимости»

Рассмотрим описание дескрипторов процессов на разных уровнях спецификации. На верхнем уровне спецификации модель описывается вектором $\{SID, uid, guid, pid, RWE, Owner\}$, где SID – идентификатор контекста безопасности (метка доступа), $Uid, guid$ – идентификатор пользователей и группы пользователей (субъектов), Pid – идентификатор приложения, RWE – атрибуты объекта доступа (права на чтение, запись и выполнение), $Owner$ – ролевой атрибут (право владельца и передачи полномочий). Здесь мы вводим только открытые описатели систем. Для каждого кода программ и кода данных создаются поля (CPL, RPL, DPL) – текущий уровень привилегий, запрашиваемый уровень привилегий на повышение полномочий и уровень доступа к данным. Условия проверки: $CPL \leq \max(CPL, DPL)$ и $RPL < (CPL, DPL)$.

На данном уровне создается дескрипторное описание правил безопасности операций ввода-вывода нижнего уровня, делается привязка к имени приложения и к идентификаторам.

Право владельца $Owner$ отображается в внутренний идентификатор программного приложения, а определяющим моментом проверки на среднем уровне является сканирование и сравнение полей записи атрибутов доступа, а не работа с атрибутами доступа.

Монитор безопасности, используемый в традиционных системах, присваивает атрибуты доступа к объекту O_i (например, права на запись, чтение и выполнение) в соответствии с политикой безопасности, контролирует временные отметки, регистрирует процесс, контролирует права субъекта. Также учитываются уровни привилегий кода данных и кода программ (DPL, CPL), которые могут принимать значения: $00 = 0$ – нулевое кольцо защиты, $01 = 1$ – 1 кольцо защиты, $10 = 2$ – второе кольцо защиты, $11 = 3$ – третье кольцо защиты.

Поле Context_id инициализируется процессорным модулем при выполнении операций синхронизации и изменения системного контекста и представляет собой в отличие от полей записи таблицы потоков планировщика задач поименованный набор как положительных, так и отрицательных значений индексов процессов (потоков).

5.3 Синтез структуры системы контроля и реализация алгоритма идентификации угроз на основе набора маркеров в многоуровневой песочнице под управлением защищенного мультидоменного гипервизора

Загрузка/выгрузка контрольных образцов вредоносного ПО производится в режиме эмуляции с поддержкой 4 типов тредов. Длина каждого цикла измерения равна 8 тактам процессора. Параметры выполнения трасс на основе счетчика команд вносятся в базу данных для дальнейшего обучения. Циклический счетчик фреймов реализован в аппаратуре ВМС-контроллера в режиме реального времени, чтобы минимизировать побочные и неконтролируемые прерывания, которые маскировали истинное время выполнения системных команд.

Критерии идентификации сторонних программно-аппаратных агентов в методике тестирования уровня защищенности:

1. На основе значений функции оценки безопасности состояний F_i , аргументы которой заданы в виде конъюнкции предикатов из восьми логических переменных $x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8$.
2. На основе выборочных центральных моментов ожидания u 8-го порядка:

$$Z = \{\vec{x}_n : u(\vec{x}_n) \geq \mu\}. \quad (5.1)$$

3. На основе выборочной дисперсии $\hat{\sigma}^2$:

$$Z = \{\vec{x}_n : \hat{\sigma}^2(\vec{x}_n) \geq d\}. \quad (5.2)$$

4. На основе длины вариационного ряда $\hat{l}$:

$$Z = \{\vec{x}_n: \hat{l}(\vec{x}_n) \geq w\}. \quad (5.3)$$

Обнаружение осуществляется на основе списка демаскирующих событий (признаков) по значениям траекторий вычислений темпоральной логики CPL на структурах Крипке, по временным параметрам (изменение длины трассы выполнения инструкции) и величины энергозатрат на выполнение инструкций в пДж. На одну операцию в процессоре тратится порядка 29-45 пДж (технология сборки 40-22 нм). Частое переключение между режимами функционирования и перезапуск задач на всем линейном функциональном устройстве требует на 2-3 порядка энергозатрат 1,1 – 19 мкДж, что позволяет обнаружить нелегитимное ПО и работающее закладочное устройство.

В момент его работы наблюдается «резкий» всплеск энергетической активности по усредненным показателям за длительный интервал наблюдений. Из выбираемых характеристик большого числа тестовых измерений в качестве пороговых значений μ, d, w выбираются те, сумма вероятностей ошибок 1-го и 2-го рода которых минимальна. В качестве порогового значения по энергоэффективности выбран параметр среднего значения $e > 1100$ пДж.

На рисунке 5.4 наглядно иллюстрируется способ функционирования ОС программно – аппаратного комплекса «Альфа-монитор» с реализацией алгоритма идентификации угроз ИБ на основе набора маркеров, который позволяет обнаружить вредоносное программное обеспечение в суперкомпьютерных системах. Процедура поиска разных классов вредоносного ПО представляет собой цикл (стрелки 6, 7 и 10), в котором анализируются списки запросов, на основе значения функции оценки состояния F_i : если $F_i = \text{Да}$, процесс добавляется в список разрешенных, если $F_i = \text{Нет}$, формируется список запрещенных процессов (см. таблицу 5.1).

В результате работы алгоритма создается база знаний вредоносного программного обеспечения (стрелка 8), которая периодически обновляется.

Стрелка 1 – условно показана подготовка загрузки управляющей ОС на узлах многопроцессорной вычислительной системы, загрузка модуля верификации команд.

Стрелки 2, 3 – обозначен процесс получения списков запросов на выделение ресурсов.

Стрелка 9 – в случае обнаружения подозрительных действий происходит оповещение администратора безопасности.

Ограничение на повышение привилегий и контроль переходов при изменении контекста операций задаются значениями логической функции F_i , а контроль выполнения потоков, порождаемых субъектами доступа, реализуется на основе принципа наименьших привилегий.

На основе заданных таблиц правил ПБ контролируется активность сетевых приложений, обращений к устройствам ВМ и отслеживаются входящие и исходящие пакеты данных.

Подсчет числа «зараженных» серверов виртуализации осуществляется по формуле
$$a = \frac{e^{K(t-T)}}{1 + e^{K(t-T)}}$$
, где параметр K содержит начальное значение среднего количества атакованных серверов виртуализации за определенный период, $a(t)$ – количество уязвимых серверов виртуализации, которые были успешно атакованы во время t , T – время проведения наибольшего количества атак на серверы. На основе собранной статистики формируется отчет об обнаруженных атаках.

Критерии идентификации угроз на наборах кортежей логических переменных ($x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8$) приведены в таблице 5.1.

Таблица 5.1 – Критерии идентификации угроз на наборах кортежей логических переменных (x1, x2, x3, x4, x5, x6, x7, x8)

x1	x2	x3	x4	x5	x6	x7	x8	Пояснение
NOP	XX	X	X	X	0	0000	00	Ожидание поступления запросов
!=NOP	0X	X	X	1	!= 0	XX1X	X1	Осуществляется попытка злоумышленника изменить конфигурацию хостовой ОС физического сервера
!=NOP	0X	X	X	-1	!= 0	XX1X	X1	Осуществляется попытка злоумышленника перехвата обращений хостовой ОС к драйверам физических устройств
!=NOP	1X	X	X	X	!= 0	0101	01	Ресурс занят приложением - штатный режим запуска задачи пользователем или администратором – асинхронный режим доступа к ресурсам
!=NOP	01	X	X	X	!= 0	0101	01	Ресурс занят приложением - штатный режим запуска задачи хостовой ОС – асинхронный режим доступа к ресурсам
!=NOP	01	X	X	X	!= 0	0100	10	Ресурс занят приложением - штатный режим запуска задачи хостовой ОС – синхронный режим доступа к ресурсам
!=NOP	XX	X	X	X	!= 0	0X1X	10	Штатное завершение активной задачи
!=NOP	XX	X	X	X	!= 0	111X	10	Принудительное завершение всех «спящих» процессов
!=NOP	XX	X	X	X	0	XXXX	XX	Попытка скрыть процесс-поток в дескрипторных таблицах хостовой ОС, присвоив статус «System Idle Process»
!=NOP	00	X	X	X	!= 0	XX1X	X1	Попытка запуска недоверенного модуля на уровне IPL

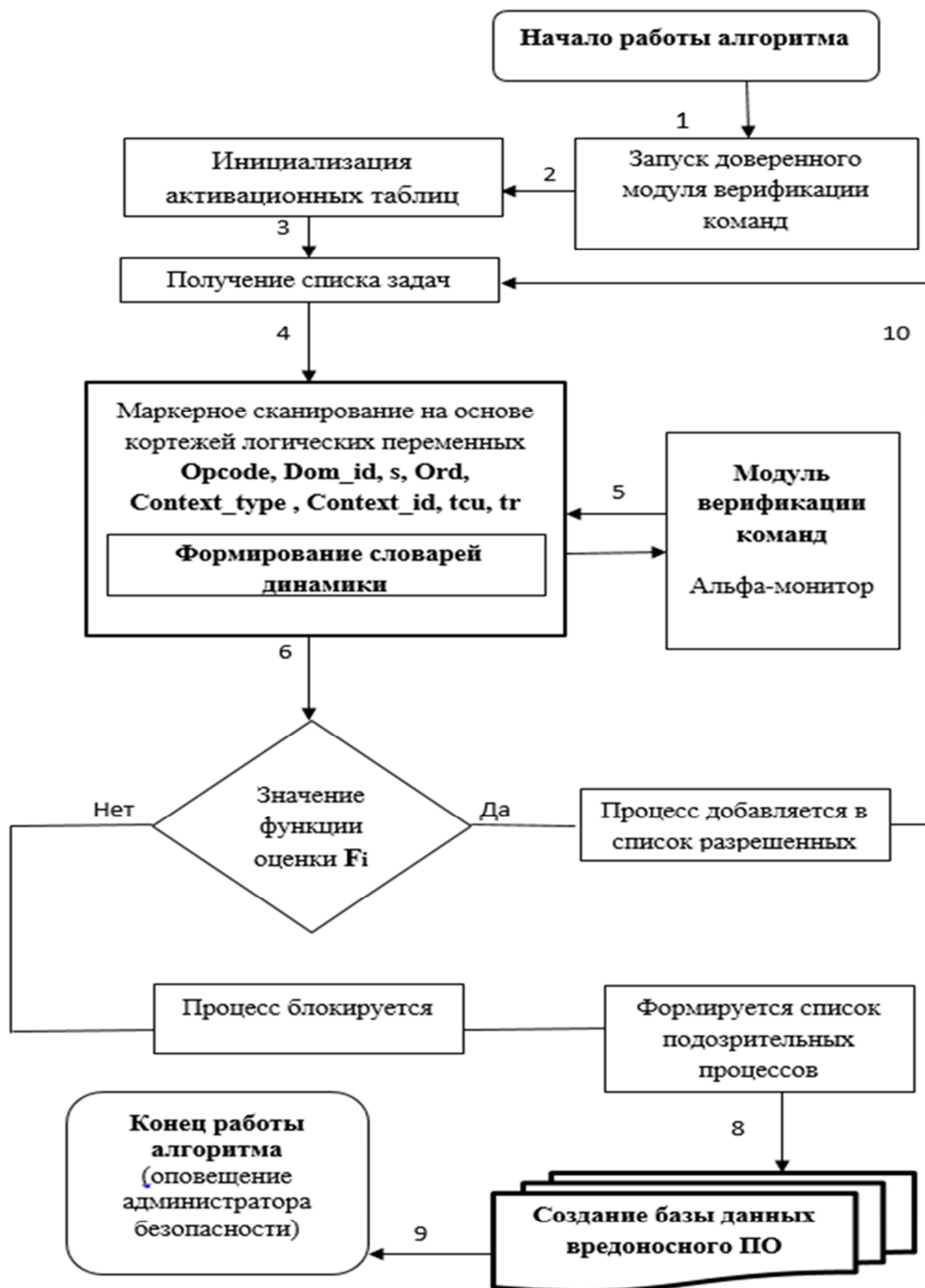


Рисунок 5.4 – Блок-схема функционирования алгоритма “маркерного” сканирования

За счет функционирования модулей контроля и защиты ПО обеспечивается уменьшение ущерба, наносимого компонентам суперкомпьютерной системы реализацией разного типа угроз.

Пространственно – временную сложность работы алгоритма и потерю производительности при использовании авторской модели операций будем оценивать с помощью критериев пространственно-временной локализации. *Пространственная локализация (spatial locality, SL)* - тенденция приложения выдавать обращения к памяти, в которых адреса находятся вблизи от адресов недавно выданных обращений. Здесь не будем давать подробное описание элементов этой формулы, а поясним качественно.

$$SL = \sum_{i=1}^{\infty} stride_i / i \quad SL \in [0, 1] \quad (5.4)$$

Измеряемая SL зависит от разброса адресов, выданных в окрестности недавно выданных обращений некоторого обращения к памяти. Это поясняется на рисунке 5.5 в пространстве такты выполнения программы – выдаваемые адреса обращений к памяти.

Обращения r2 и r4 – это обращения в перед которыми на некотором интервале рассматриваются адреса до получения выданных обращений.

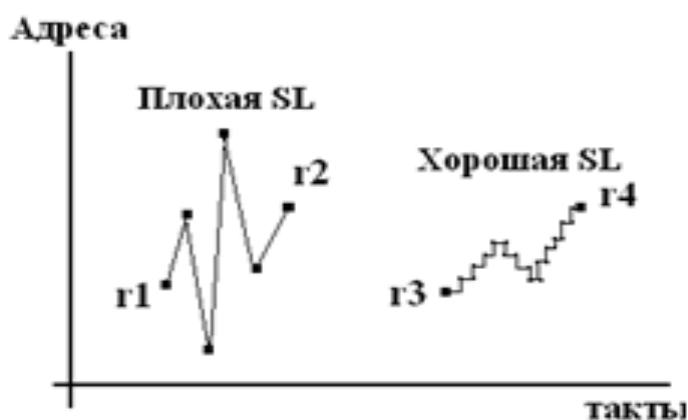


Рисунок 5.5 – Иллюстрация пространственной локализации на профиле обращений к памяти программы

Временная локализация (temporal locality, TL) – тенденция приложения выдавать обращения к памяти к тем же адресам, которые были в недавно выданных обращениях.

Для оценки быстродействия предложен тест Арех-Мар [126], который искусственным образом синтезирует разные пространственно-временные режимы локализации обращений к памяти, используются обращения на считывание, при этом измеряет среднее количество тактов процессора, приходящихся на выполнение одного обращение. В трехмерном базисе строится поверхность распределения плотности вероятности обращений к памяти на основе формулы (5.4).

В целом, тест Арех-Мар представляет собой следующую процедуру (см. рисунок 5.6). Выделяется большой массив M объемом около половины физически доступной оперативной памяти. В этом массиве периодически производится прохождение с единичным шагом ячеек памяти из областей с заданным параметром теста объемом L (меняется в границах, гораздо меньше M). Размещение этих областей в M меняется случайным образом, что управляется параметром A (или α), который больше нуля, но меньше или равен единицы. Параметр L задает пространственную локализацию, а параметр A определяет временную. Работа теста Арех-Мар состоит в том, что он проходит весь диапазон значений L и A и для каждой пары выбранных значений определяет среднее время выполнения обращения к памяти на считывание. Эта точка рассматривается как одна из точек АРЕХ-поверхности, которую строит этот тест в конечном итоге.

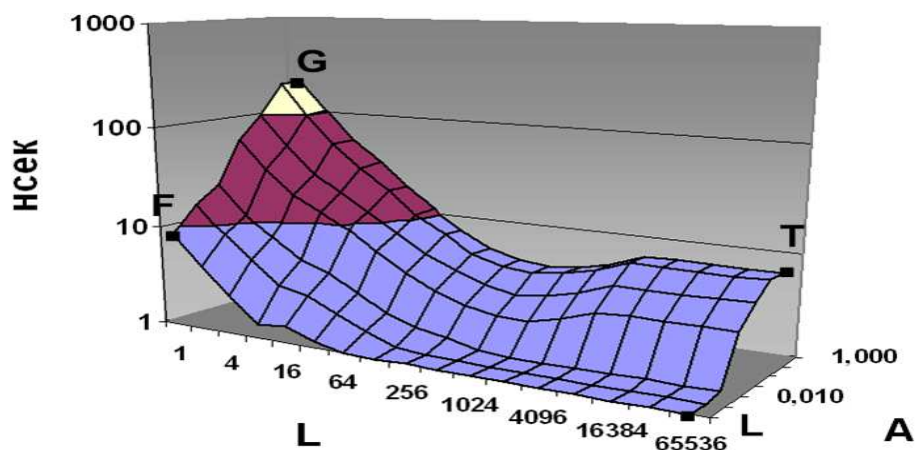


Рисунок 5.6 – Пример APEX-поверхности

5.4 Разработка и реализация программно-технических решений, результаты тестовых испытаний

Управление сервисными и вычислительными узлами, объединенными высокоскоростными линиями сети, осуществляется с помощью многомодульного защищенного мультидоменного гипервизора.

Разработаны два варианта исполнения: программный и программно-аппаратный.

Программная реализация состоит из проксирующих модулей и модуля верификации команд гипервизора, программных агентов сбора метрики приложений и мониторинга событий безопасности на всех восьми уровнях иерархии обработки операций. Аппаратная составляющая программно-технических решений представляет собой устройство в защищенном исполнении. Вычислительное устройство секционного типа.

Реализация метода реконфигурации среды выполнения программ проактивной защиты СК с учетом требований мобильности для решения разного класса задач позволила создать изолированную исполняемую среду для разных типов суперкомпьютеров с гибкой конфигурацией окружения и контекстом вызовов сервисов, сбором метрики приложений для линейки отечественных специализированных СБИС: ЕС8430 «Ангара», КОМДИВ

(серия 1890BM6Я, серия 1890BM7Я), КОМДИВ (серия 1900BM2Т, серия 1907BM044, серия 1907BM028, серия 1907BM038), Байкал-М ARM 7/8, МЦСТ Эльбрус 4С/8С/16С. Учтен и настроен расширенный контроль конфигураций в соответствии с требованиями политик безопасности и техническими условиями по эксплуатации изделий.

На основе модели безопасных операций реализованы функции оценки безопасности состояний системы в виде вычисления результирующей свертки на основе атрибутов протекающих процессов и конфигурации среды выполнения с заданной глубиной (масштабируемостью).

На основе модели угроз разработана и реализована технология следящих действий со стороны гипервизора и управляющих ОС. Показатели защищенности адаптированы к классу решаемых задач, требованиям к среде передачи данных при обмене данными между бортовыми (мобильными) и стационарными суперкомпьютерными вычислительными системами.

На основе метода реактивной защиты спроектированы и внедрены технологические модули «Альфа-монитор» для российских суперкомпьютерных платформ семейства Ангара ЕС1740, для китайских суперкомпьютеров серии Tian-he/СТ-2, японских суперкомпьютеров серии Tsubame 3.0. Вычислительные блоки суперкомпьютеров защищены схемой коррекции, внедрены расширенные механизмы обеспечения безопасности выполнения операций (запросов) как на уровне прикладного и системного ПО, так и на уровне аппаратуры.

Разработаны и протестированы механизмы защиты по контролю контекстно-зависимых операций, фильтрации операций и контроля тайминга их выполнения гипервизорными модулями и контроллерами памяти. Внедрен модуль с функциями верификатора команд на уровне привилегий IPL.

В рамках методики тестирования уровня защищенности введена новая характеристика для создания защищенных систем: коэффициент готовности к безопасной эксплуатации СК, реализован алгоритм идентификации угроз на основе набора маркеров. Исследования на макете позволили определить потенциальные возможности несанкционированных воздействий с использованием промышленных протоколов передачи данных, включая специализированные высокопроизводительные сети.

В рамках инженерных экспериментальных исследований был разработан многокомпонентный защищенный мультидоменный гипервизор в нескольких вариантах исполнений: программном и программно-аппаратном. Гипервизор состоит из проксирующих модулей, верификатора команд и монитора безопасности. Подсистема обеспечения информационной безопасности суперкомпьютеров серии «Ангара» ЕС1740 реализована на базе мультидоменного гипервизора «Альфа-монитор» и коммутаторов Ангара-МВС. Управление сервисными и вычислительными узлами, объединёнными высокоскоростными линиями сети «Ангара», осуществляется с помощью защищенного распределенного мультидоменного гипервизора. В программно-технических решениях, с точки зрения масштабируемости и кроссплатформенности, учтена поддержка как зарубежных кластерных ОС (Red Hat, Debian и др.), так и отечественных кластерных ОС.

Рассмотрим макет вычислительного устройства на основе данных из открытых патентов на изобретения [10; 11] (см. рисунок 5.7).

Аппаратная составляющая программно-технических решений представляет собой устройство в защищенном исполнении. Вычислительное устройство секционного типа.

Наряду с коммерческими общедоступными процессорами могут монтироваться и специализированные высокопроизводительные процессоры (например, ядра MT0 и MT1).

Один кристалл имеет два ядра: **MT0** и **MT1**. Количество ядер может быть увеличено до 512. К этому микропроцессору подключена основная память типа DRAM вычислительного узла, а также память типа NVRAM (энергонезависимая память, ее можно рассматривать как сверхбыстрый диск). Стрелками 1, 2 условно показаны каналы обмена между процессорными ядрами MT0, MT1 и блоком ММО.

Блок ММО – это функционально-логическое устройство, состоящее из **конвейера обработки запросов TCU** и **сетевого сопроцессора**. В отличие от традиционного конвейера обработки запросов, где передается только сигнатуры операций, здесь формируются генеративные таблицы на этапе приготовления задачи и хранятся в специальных тредовых регистрах `t_reg`.

Они представляют непроцедурную нефункциональную, а семантическую часть кодирования цепочки клиентских запросов. После чего передается стандартная процедурная составляющая – опкод инструкций (набор кода данных и кода команд, модификаторов и флагов условных переходов).

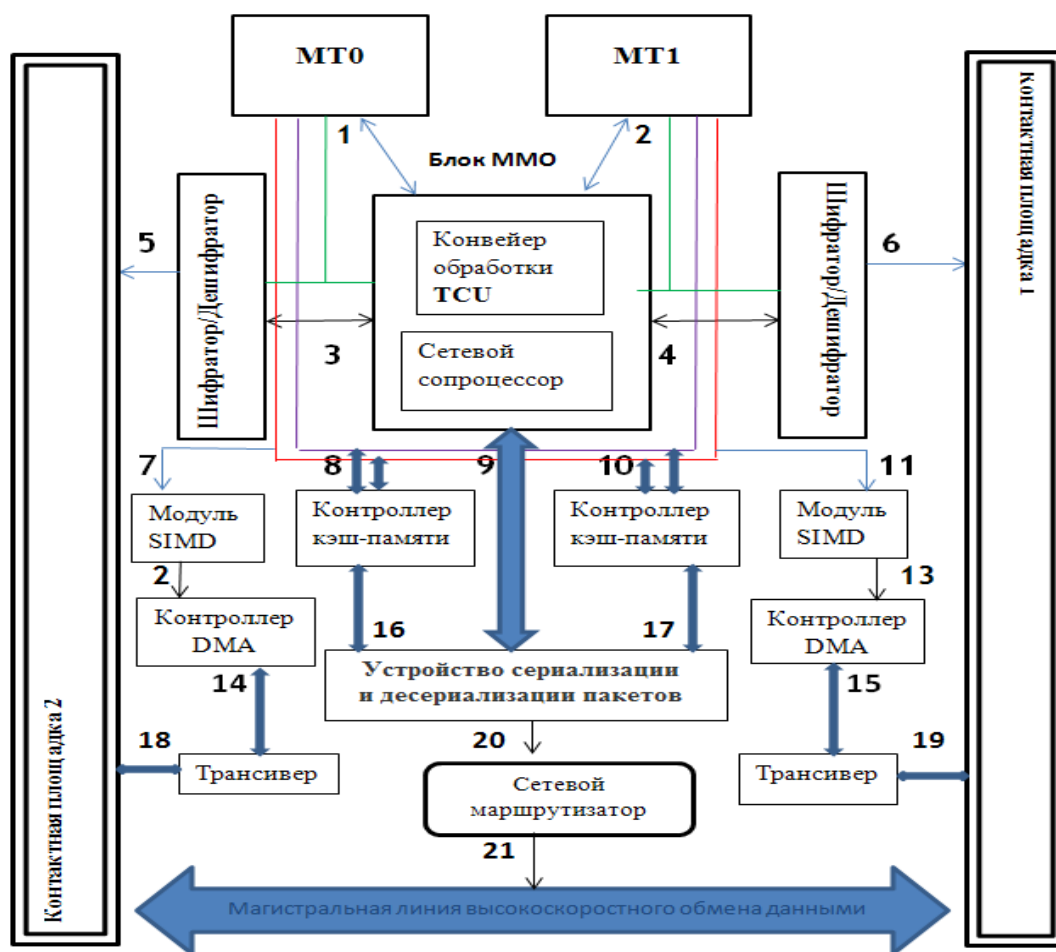


Рисунок 5.7 – Пример реализации защищенного вычислительного устройства

Цветными линиями показаны внутрикристальные шины: **красная** – шина данных, **зеленая** – шина выдачи команд (например, активных сообщений типа удаленного вызова процедур. Если таких парселов на задаче оказывается мало, то эти магистрали используются для передачи данных), **фиолетовая** – шина передачи пакетов сообщений.

Дешифратор/шифратор – устройство, занимающееся кодированием/декодированием команд процессора, имеет прямое канальное подключение к аппаратным ядрам процессора, сетевому сопроцессору и конвейеру обработки запросов (стрелки 3, 4) с возможностью обмена «на лету» с функциональными устройствами соседнего кристалла секционной сборки (стрелки 5, 6).

Сетевой сопроцессор позволяет выполнять математические операции с пакетами сообщений и передавать результаты на межмодульную сеть напрямую, не задействуя микропроцессор, благодаря чему получается сильнейшая экономия ресурсов ядер МТ0 и МТ1 с повышенной толерантностью к задержкам выполнения операций с памятью и сетью.

Кроме сетевых операций, выполняет следующие функции: выполняет трансляцию виртуальных адресов, после трансляции адреса разделяет обращения в локальную память узла от обращений в удаленные узлы, производит агрегирование обращений к удаленным узлам, а также распаковку поступающих сообщений от удаленных узлов, выполняет локализованные в непосредственной близости от памяти функции, которыми могут быть атомарные операции, транзакционные преобразования структур данных, сложные функции синхронизации типа действий специальных узлов синхронизации или преобразования данных (акторы). В силу этого он успешно справляется с обработкой в реальном времени множества интенсивных потоков данных и эффективно работает через единое адресное пространство с огромным объемом памяти в несколько десятков петабайт даже в режиме интенсивной нерегулярной работы с ней, обладать исключительной отказоустойчивостью и готовностью.

Кристаллы микропроцессора размещаются без корпусов на керамической подложке. На подложках имеются сверхскоростные соединения и контактные переходы между подложками. Это позволяет одновременно с масштабированием количества ядер масштабировать и интерфейсы с памятью, сетью, а также внутрикристальную сеть. Это возможно благодаря тому, что можно соединять секции сверхбыстрыми связями на трех уровнях (конвейеры команд, функциональные устройства, порты внутрикристальных коммутаторов со стороны конвейеров и функциональных устройств).

Контактная зона 1 и Контактная зона 2 – высокоскоростные линки подложки, выполненные с использованием нанотехнологий, для обеспечения прямой связи кристаллов микропроцессора и для реализации N-мерной сети модулей для организации сверхбыстрого обмена данными между разными секциями сборочных модулей вычислительного устройства. Представляют собой соединения бесштырькового типа с использованием карбоновых нанотрубок с металлическими лигандами. Процессорные модули вычислительного устройства могут быть разными по типу поддерживаемых в ядре процессов и эмулировать выполнения команд одного или множества процессов-тредов – сверхлёгкие, легкие, средней тяжести, тяжелые: сверхлегкие ядра графических процессоров (GPU), средние ядра RISC-микропроцессоров типа MIPS, тяжелые ядра суперскалярных микропроцессоров с архитектурой x86, легкие ядра векторных процессоров с поддержкой SIMD-режима.

7, 11, 12, 13 – каналы обработки легких и сверхлегких процессов-тредов: в схему добавлен модуль SIMD, который напрямую по линиям 14 и 15 получает доступ через DMA-интерфейс (на схеме контроллеры DMA) к WDM – трансиверам, которые служат для реализации передачи сообщений через оптоволокно одновременно на разных длинах волн (магистраль 18 и 19).

9 – канал обработки средних процессов-тредов: для работы реализована связь с устройством сериализации/десериализации пакетов сообщений с целью уплотнения (сжатия) передачи данных, обеспечивает распознавание большого количества форматов сообщений, реализации методов агрегирования коротких сообщений в длинные и обратное преобразование, затем они передаются через сетевой маршрутизатор во внешнюю высокоскоростную магистраль (стрелки 20, 21). При этом не используется «накачка» данных из кэш-памяти.

8, 10, 16, 17 – каналы обработки тяжелых процессов-тредов: требуется для выполнения инженерно-расчетных задач с «накачкой» данных из кэш-памяти. Могут использоваться не только 3D-сборка, но и трехмерные СБИС. В системе используются средства работы с очень большими данными (большими объемами данных из многоуровневой памяти системы), а также выделенные магистрали для работы с сетями передачи данных (датчиками, через которые поступают потоки данных). Возможна организация обработки данных с использованием синхронных и асинхронных тредов, а также с использованием векторов разной длины.

Проектируемое вычислительное устройство с несколькими уровнями иерархии, высоким параллелизмом и глобально адресуемой памятью предполагает наличие мощного программного обеспечения с динамическим распараллеливанием программ, средствами оптимизации, учитывающими неоднородность доступа к глобально адресуемой памяти.

Объем доступной памяти программе пользователя – **32 ПБайт**, физическая память – **64 ПБайт**, что сделано для горячего резервирования.

В качестве областей применения опытного образца вычислительного устройства: квантовая информатика (система команд содержит специальные команды ее поддержки), DIS-задачи (задачи с интенсивной нерегулярной работой с памятью), высокоскоростная обработка сигналов и изображений, метеорология, георазведка, работа с хранилищами разнородной информации огромного размера (RDF - базы данных).

В качестве средств разработки использовался программный пакет схемотехнической разработки VHDL, GNU ассемблер, отладчик Total View. На основе результатов проведения испытаний составлены в таблице 5.2. При этом количество успешных обнаружений:

- 1) Kaspersky Security Center – 2464.
- 2) NortonSecurity Cluster Center – 2340.
- 3) VIP NET Cluster Firewall – 2200.
- 4) Альфа-монитор – 2600.

Таблица 5.2 – Обнаружение вредоносного ПО

Название вредоносного ПО и интерфейсные уровни работы компонентов СЗИ	Hacker Defender, Hox(S1-S3)	Seven Pandora (S3-S5)	Storm (S5)	Croax, Legend (S6)	Ice Brute, Dragon, VICE, HIVE iL (S7)	Rustock1, Rustock2, Rustock3, Spectre Meltdown (S7)
Kaspersky Security Center	+	+	–	+	–	–
NortonSecurity Cluster Center	+	+	–	+	+	+
VIP NET Cluster Firewall	+	+	–	+	–	–
Альфа-монитор	+	+	+	+	+	+

«+» – успешное распознавание

«–» – ошибка

Разработанные программно-технические модули прошли опытное тестирование. Поддерживаемые платформы: Intel, AMD, Tiler, Cray XMT 1/2/3, Angara J7/J10, Loong Song 200/300, Fei-Teng 3000/4000, IBM Power 7/8, Комдив, Грифон, Эльбрус 4C/8C/16C, Байкал T1/T2. В планах на будущее реализовать поддержку процессоров Monarch.

Первый эксперимент заключался в модификации гостевых и хостовых ОС вредоносным ПО: Seven Pandora, Hox, Hacker Defender, Storm, Croax, Legend, Blue Pill, VICE Toolkit, DRM, Ice Brute, Rustock, Dragon, HVMiL, Spectre, Meltdown и др. Компоненты указанного вредоносного ПО загрузились и устанавливались на физические серверы и виртуальные

машины суперкомпьютера. Затем осуществлялось сканирование с применением разных средств антивирусной защиты и анализировалось число успешных распознаваний и ошибок. В качестве тестового набора ПО выбраны СЗИ с поддержкой кластерных технологий для развертывания в суперкомпьютерной инфраструктуре.

В качестве репрезентативной выборки рассматривался процесс обнаружения вредоносного ПО и оперативность восстановления системы в случае исключений и ошибок на платформе Windows и Linux. Результаты опытов представляют собой матрицы 1000x10, содержащие данные измерений длительности выполнения трассы (мкс) для случаев отсутствия и присутствия недоверенного ПО. Для обеспечения требований ISO 5725 к сходимости и воспроизводимости данных измерений опыты выполнялись сериями из 5-ти повторных опытов в течение 14-ти дней до стабилизации наблюдаемых эмпирических данных.

Таблица 5.2 наглядно иллюстрирует, что классический метод контроля, обеспечения защиты ПО с использованием 4 базовых уровней S1 – S4 показывает наихудший результат тестирования. На рисунке 5.8 проиллюстрирована общая структура многоуровневой “песочницы”, в которой запускаются и анализируются приложения СК.

- 1) Синим выделены разрешенные дуги переходов;
- 2) Красным выделены запрещенные дуги переходов;
- 3) Зеленым – дуги сборки команд непосредственного выполнения исходящих директив;
- 4) Вся «карта состояний» прохождения запросов поделена на восемь уровней иерархии и 8 этапов обработки с учетом режима работы процессора;
- 5) Существуют 3 зона (уровня – выделены синим) – для работы с памятью устройств гостевых и управляющих ОС и требующих повышения уровня привилегий;

- 6) Склейка тегов Dom_id, S 010 110 100;
- 7) Зоны (уровня – выделены рыжим)– для работы с приложениями, не требующими изменения бита системности 011 111 101;
- 8) Зона (S7 – выделен серым прямоугольником) – для работы гипервизора с оборудованием и не требующих изменения бита системности 001;
- 9) Дополнительный уровень (S8 – выделен бордовым прямоугольником)– IPL – уровень, для работы гипервизора в режиме верификации и сборки команд сценария выполнения 000;
- 10) Проверка на наличие непустых команд;
- 11) Формируется очередь запросов. Анализируется процесс прохождения запросов с первого уровня по восьмой. Если меняется фрейм запроса в процессе его обработки, то происходят изменения значащих битов тегированных полей: S, Context_type, Ord, tr;
- 12) Связка dom_id и бит S – ограничение на повышение уровня привилегий за счет появления значащего бита. Отслеживается изменение с 1 на 0. Всего 8 уровней привилегий;
- 13) Введены тегированные поля индикация обращения к привилегированным областям памяти (поле context_type);
- 14) Вычисляется уникальный id для каждого процесса в виде хеш-значения;
- 15) Анализируется «трассы выполнения команд» с учетом переключений всех возможных режимов работы ЦП в виде набора меток (маркеров).

Фактически работаем в виртуальной среде с разными режимами эмуляции и виртуализации, переключаясь между которыми, можно безопасно распараллеливать выполнение процессов, не боясь, что они будут конфликтовать.

Параллельные процессы не могут «расплываться» по всем участкам памяти. Они привязаны к определенному кольцу защиты, контролируется контекст выполнения операций запросов на всех уровнях иерархии. Операции выполняются в определенном режиме работы процессора и обращаются только к выделенным им сегментам памяти данных и программ, а не случайно перемещаются по всей оперативной памяти.

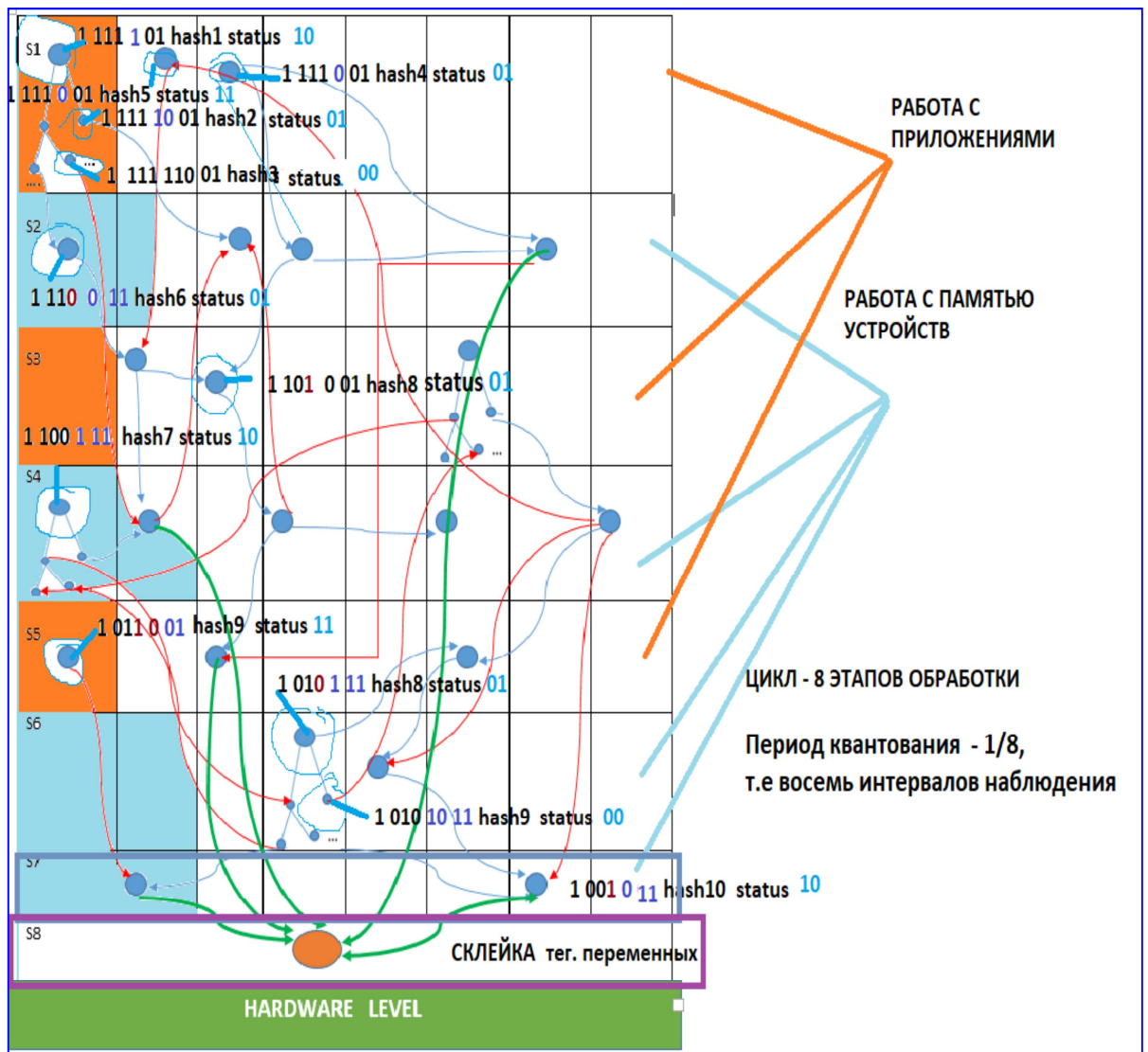


Рисунок 5.8 – карта состояний процессов в многоуровневой “песочнице”

Основные этапы верификационных испытаний:

- 1) Приходит запрос (в виде фрейма пакета запросов) при работе с аппаратной транзакционной памятью на выделение необходимых ресурсов приложению СК и начинается его обработка. Подробное описание полей фреймов запросов было приведено в Главе 3;
- 2) Передается опкод операции, которую надо выполнить;
- 3) Введено дополнительное тегированное поле, в которое записывается ID домена, в котором запускается процесс;
- 4) Введен дополнительный интерфейсный уровень контроля S8, на котором функционирует монитор безопасности гипервизора. При этом программные агенты и датчики сбора информации работают на всех уровнях иерархии;
- 5) При попытке изменить порядок обработки и контекста операций при трансляции гипервизором происходит изменение значащих битов в тегах S Context_type Ord Context_id и поле TR. CR – контр. биты сверточных кодов;
- 6) TCU_status, TR – значения полей, служащих для контроля статуса выполнения операций в гипервизоре;
- 7) TCU_B – Признак блокировки при обработке или передачи пакетов.

Второй эксперимент – тестирование, в ходе которого сравнивались выполнения разного класса задач на физическом кластере, на виртуализированных узлах под управлением «тяжелых» гипервизоров (Xen/kvm) и под управлением «легких» гипервизоров (LXC). Работа в суперкомпьютерной среде делает приложение очень восприимчивым к пространственной локализации обращений в память. В точке с плохой пространственной и временной локализацией (точка G) на одно обращение к памяти под управлением гипервизора Xen(KVM) тратит от 2 (1 процесс) до 3 (16 процессов) раз больше времени, чем под управлением LXC или ОС.

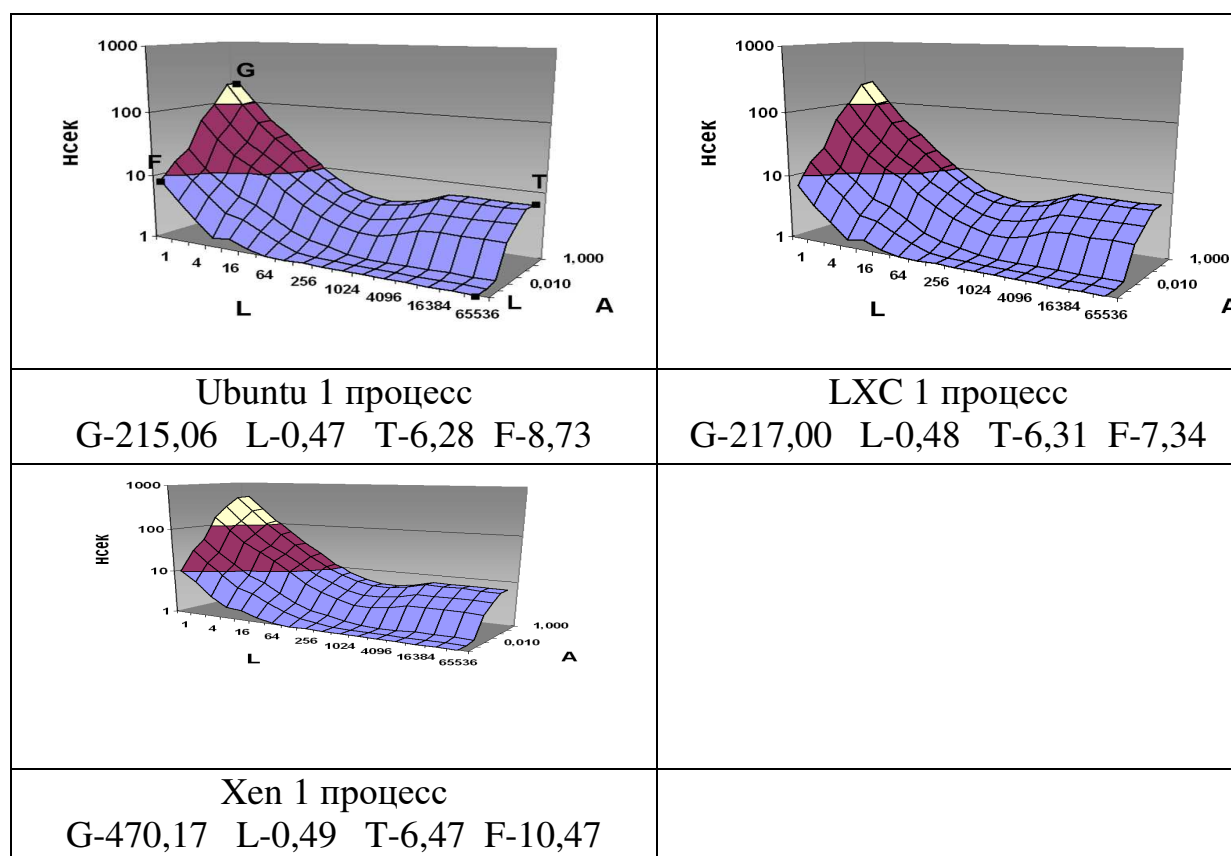


Рисунок 5.9 – Арех – поверхности, 1 процесс

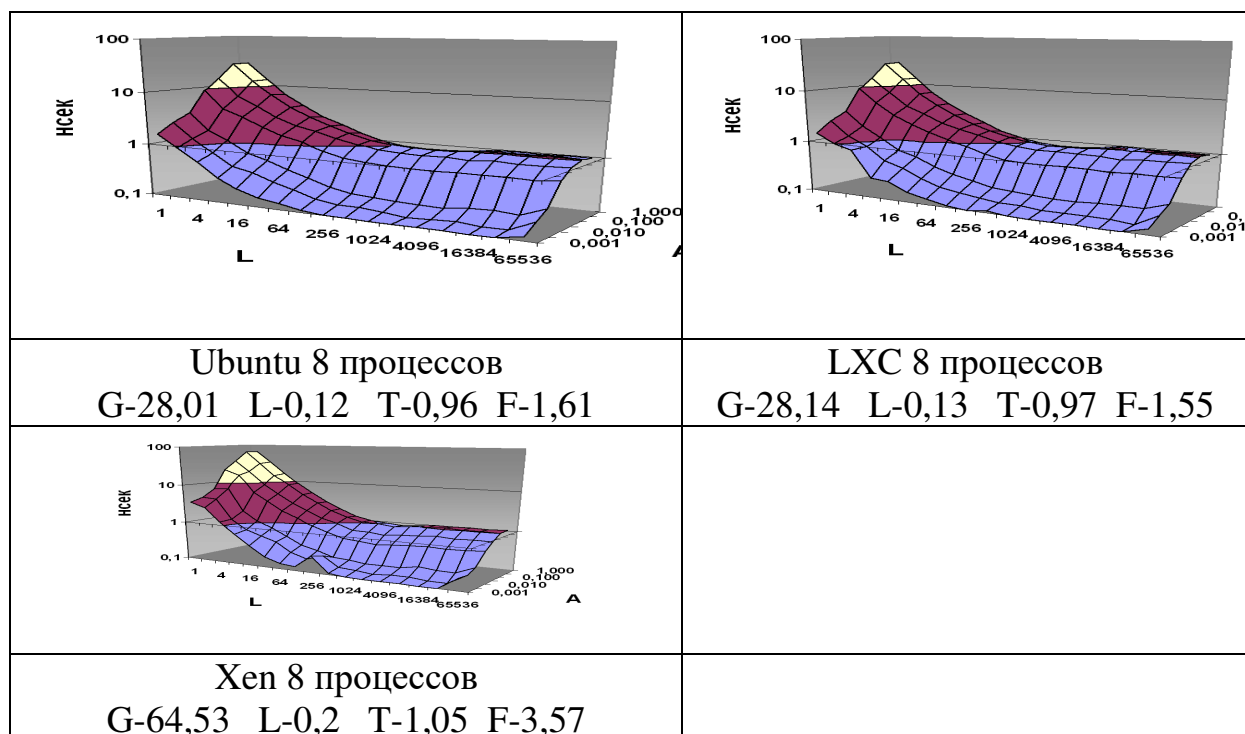


Рисунок 5.10 – Арех – поверхности, число процессов равное половине ядер узла – 8 процессов

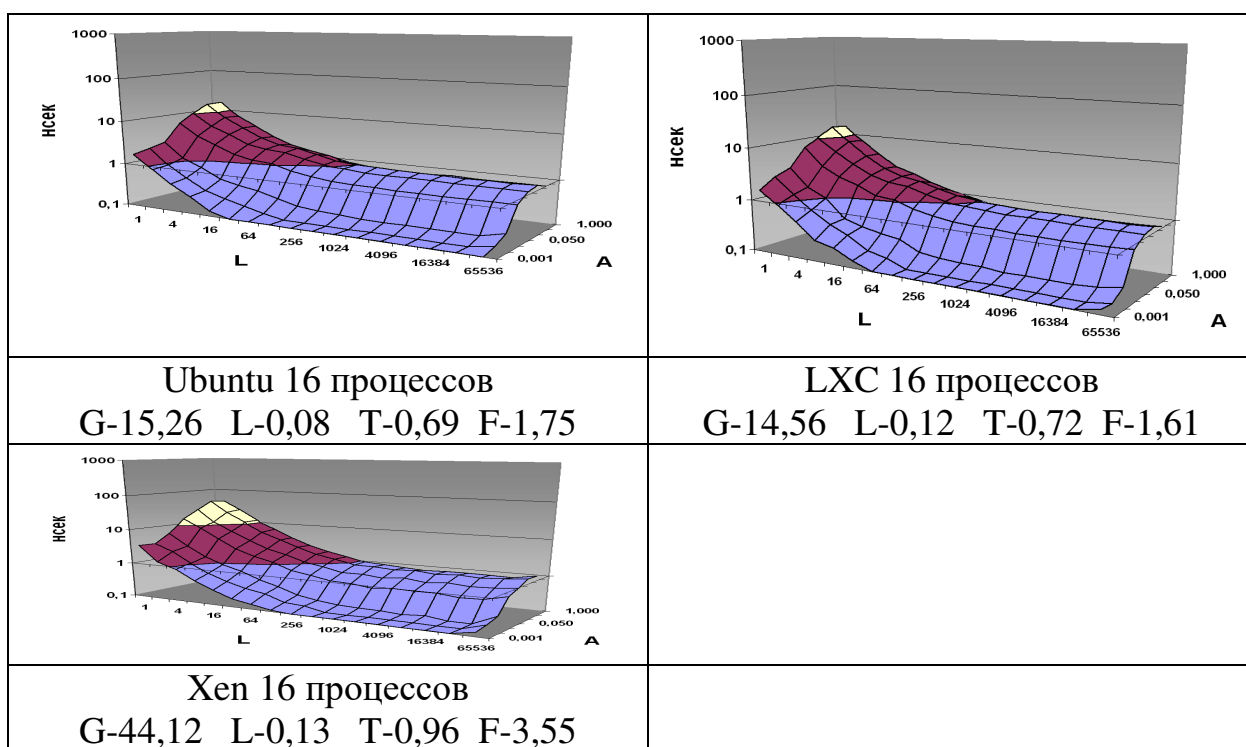


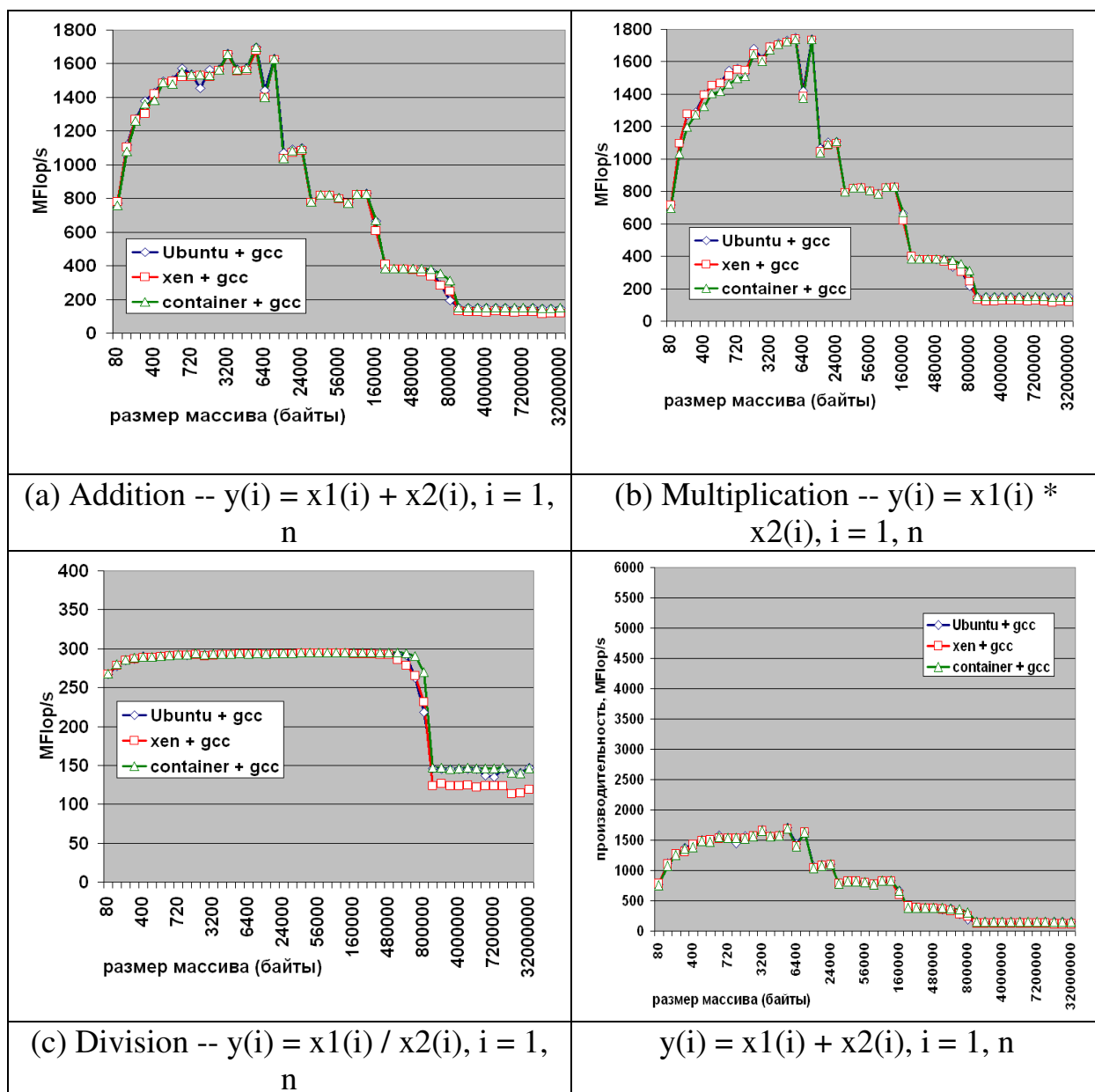
Рисунок 5.11 – Арех – поверхности, число процессов равно числу ядер - 16

В точке с хорошей временной локализацией и плохой временной локализацией (точка F) отставание Xen (KVM) меньше, но, все же, остается существенным: от 1,2 раза для одного процесса до 2 раз для 16 процессов. В области хорошей пространственной локализации (точки T и L) для одного процесса времена на выполнение обращения к памяти совпадают.

Для 8 процессов наблюдается проигрыш Xen (KVM) в 2 раза в точке хорошей временной и пространственной локализации (точка L) по сравнению с «чистой» ОС и LXC, при сравнимых результатах в точке плохой временной и хорошей пространственной локализации (точка T).

Для 16 процессов наблюдается проигрыш Xen (KVM) в 1,5 раза в точке плохой временной и хорошей пространственной локализации (точка T), при сравнимых временах обращения в точке хорошей временной и пространственной локализации (точка L).

Результаты тестирования на пакете APEx-test показали, что средняя потеря производительности меньше 7%. Результаты тестирования на пакете Euroben проиллюстрированы на графиках (см. рисунки 5.9, 5.10 и 5.11).



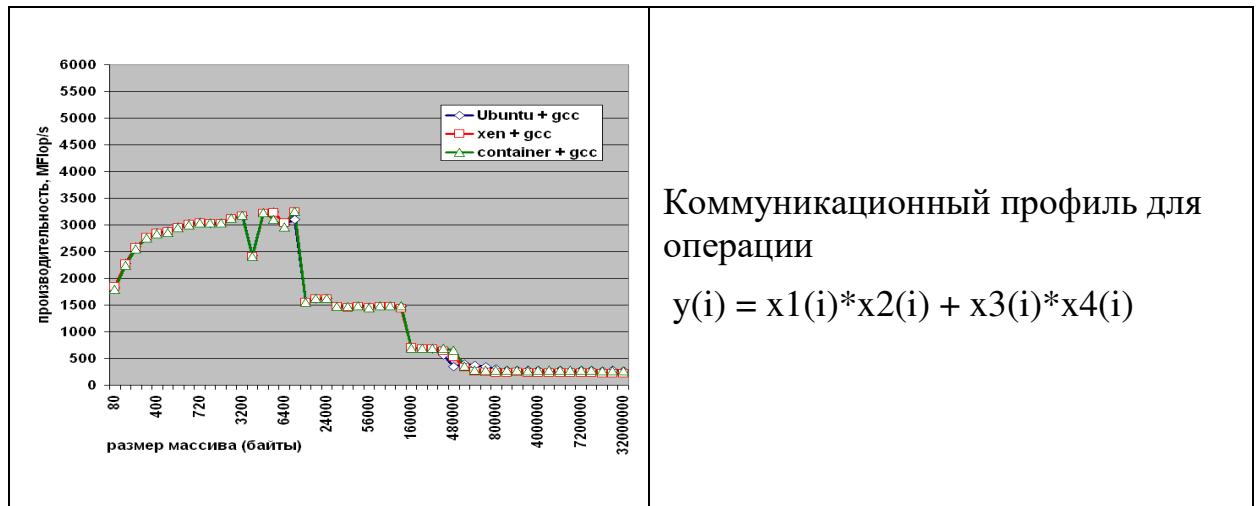


Рисунок 5.12 – Производительность МФлоп/с на операции сложения векторов

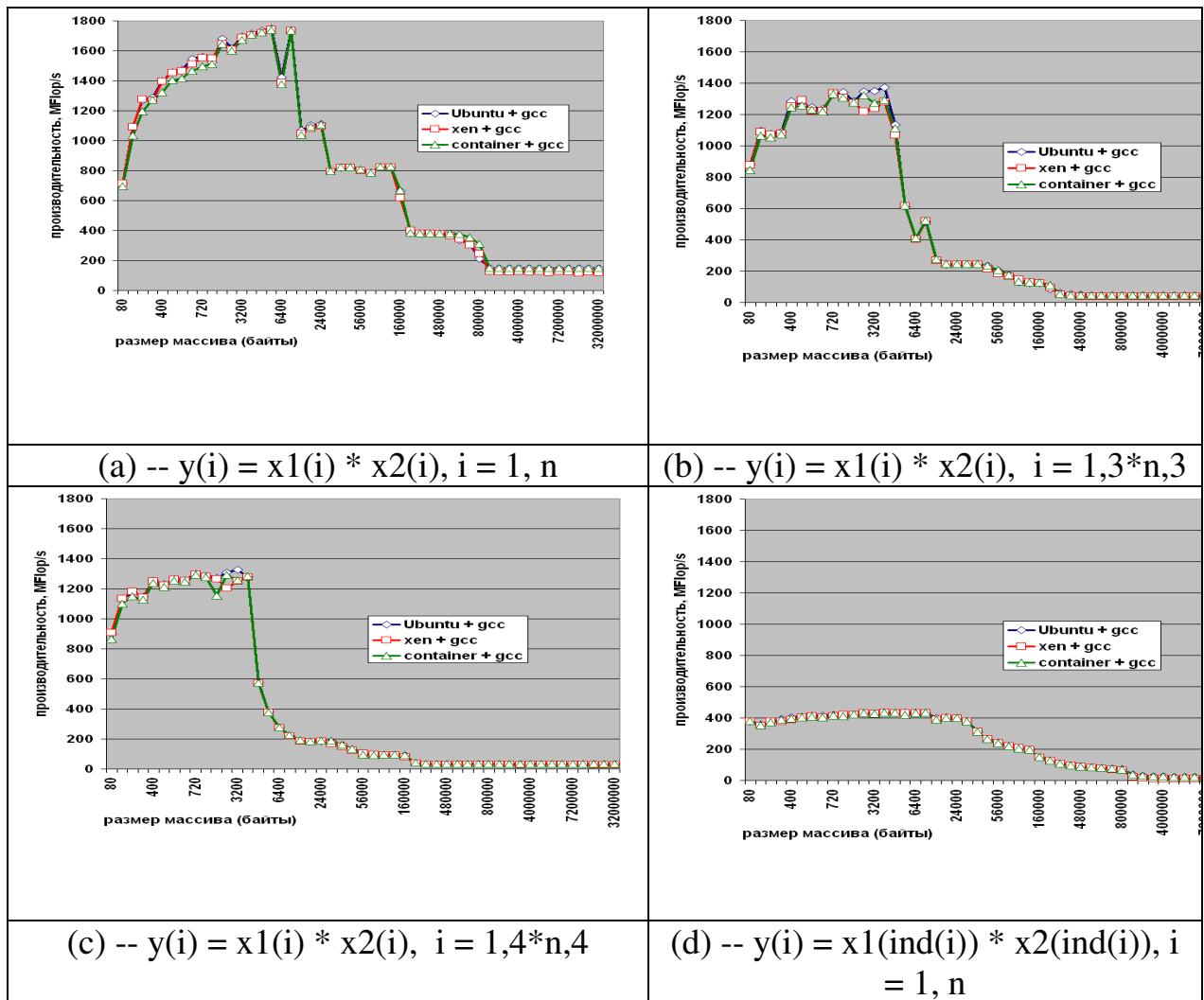


Рисунок 5.13 – Производительность МФлоп/с на операциях группы 1

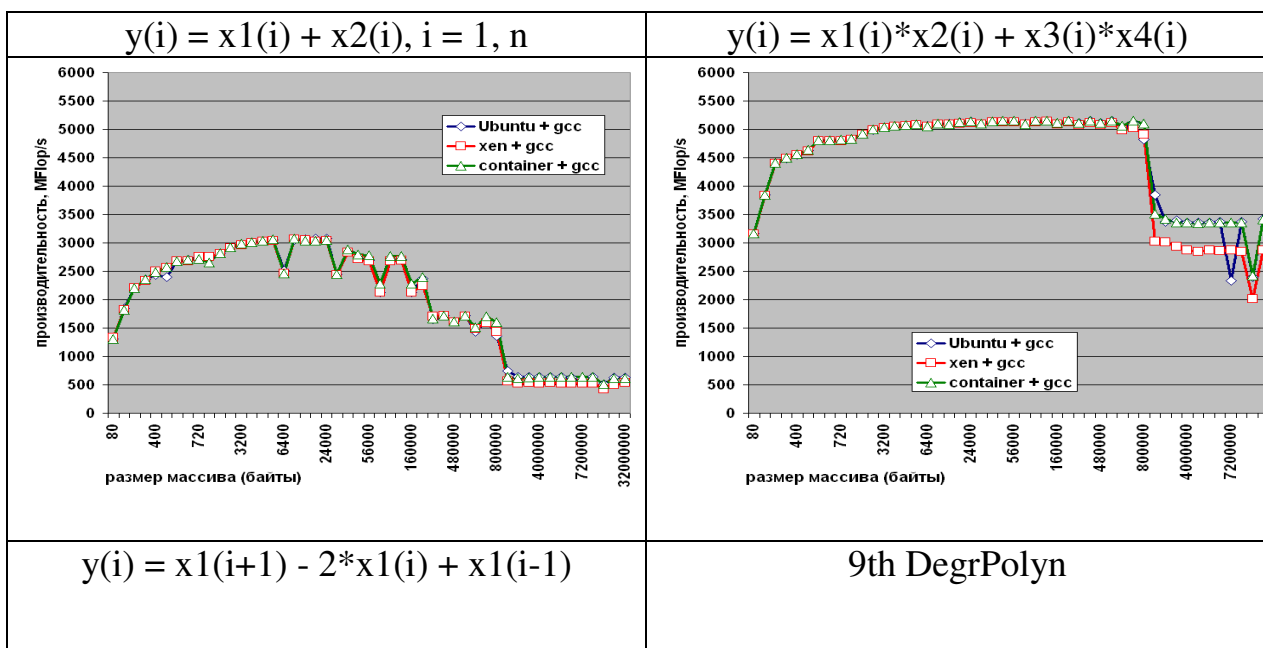
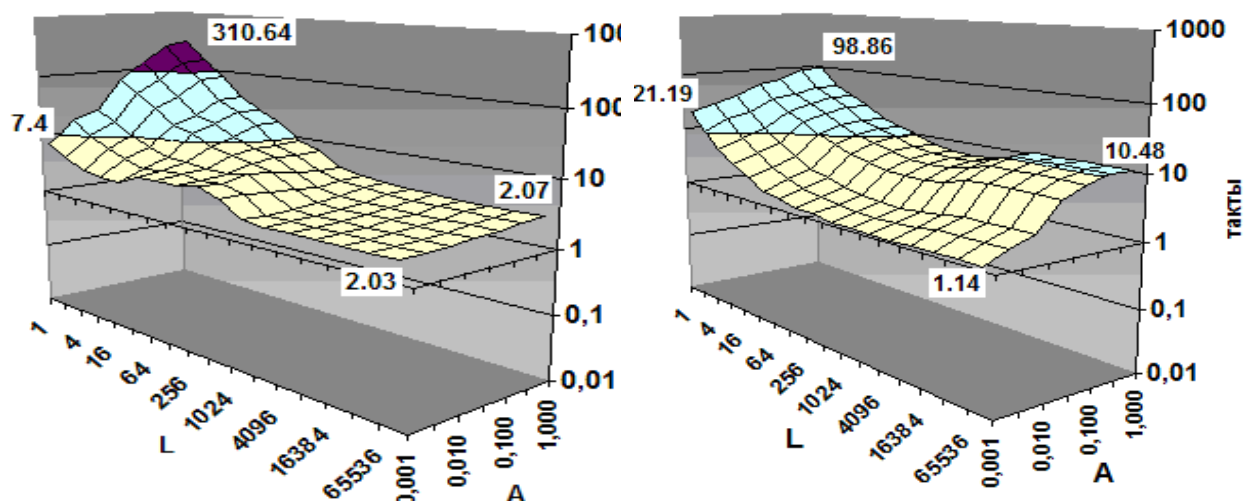


Рисунок 5.14 – Производительность МФлоп/с на операции группы 2

Каждая тестовая выборка имеет свою отличительную особенность с точки зрения использования вычислительных ресурсов (коммуникационный профиль, тип наиболее часто исполняемых команд, рабочее множество адресов обращений к оперативной памяти).

Результаты измерений на тестах пакета Euroben показывают, что производительность на рассмотренных операциях не зависит от использования «чистой» операционной системы или средств виртуализации, если размеры рабочих массивов помещаются в кэш памяти.

Для получения представления о профилях работы с памятью для задач с разной пространственно-временной локализацией рассмотрим рисунки 5.12 – 5.15.



Westmere, 1 ядро

MagnyCours, 1 ядро

Рисунок 5.15 – APEX-поверхности для одного ядра микропроцессоров Westmere и MagnyCours. L – Пространственная локализация, влияет на попадание в кэш-строку и правильность преднакачки данных в кэш-память. A – Временная локализация, влияет на попадание в кэш-память по одному и тому же адресу.

На рисунках 5.16 – 5.17 в координатах “адресное пространство - время” представлены профили задачи CF-класса (умножение плотно заполненных матриц, этот профиль похож на профиль теста Linpack рейтингового списка Top500), а также DIS-задач (преобразование Фурье, умножение разреженной матрицы на вектор, поиск вширь в графе – тест BFS (он используется для составления нового рейтингового списка Graph500)).

Каждая точка на этих рисунках имеет координату “адрес-время”, соответствует обращению к памяти, а цвет точки обозначает тип операции – считывание (красный), запись (зеленый), атомарная операция (голубой).

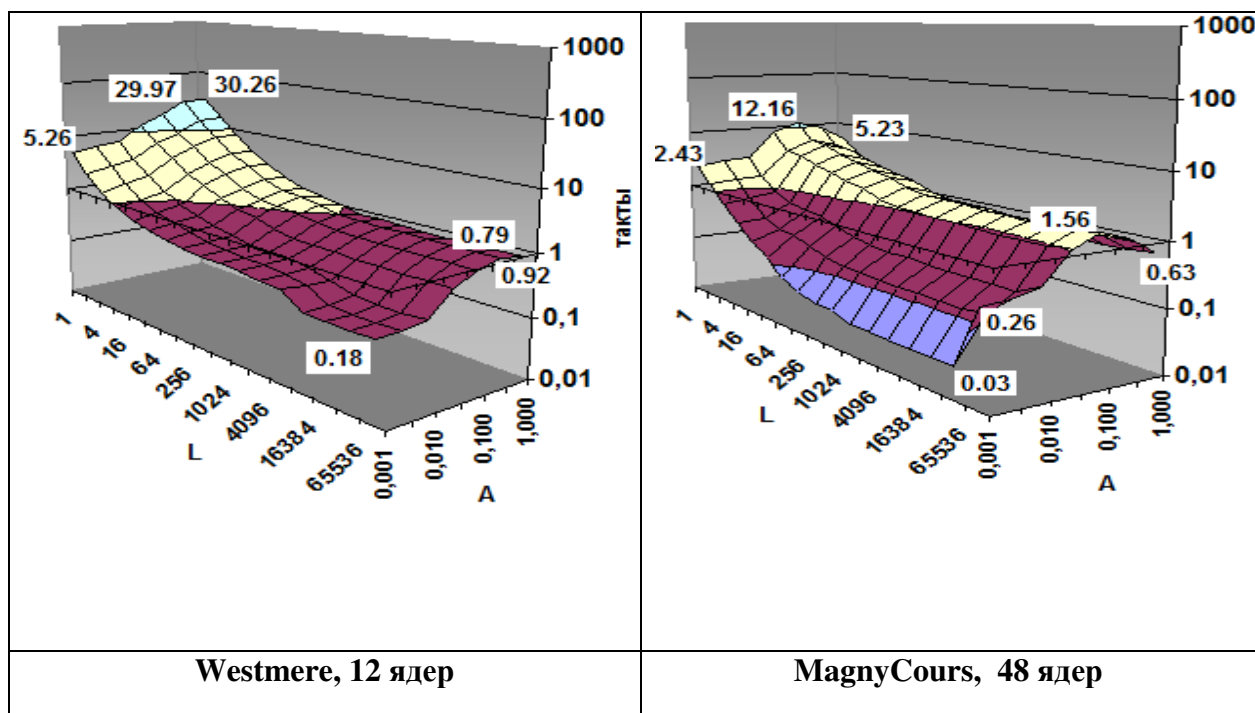


Рисунок 5.16. Демонстрация толерантности - результаты на тесте APEx-MAP для одного узла (максимальное количество ядер).

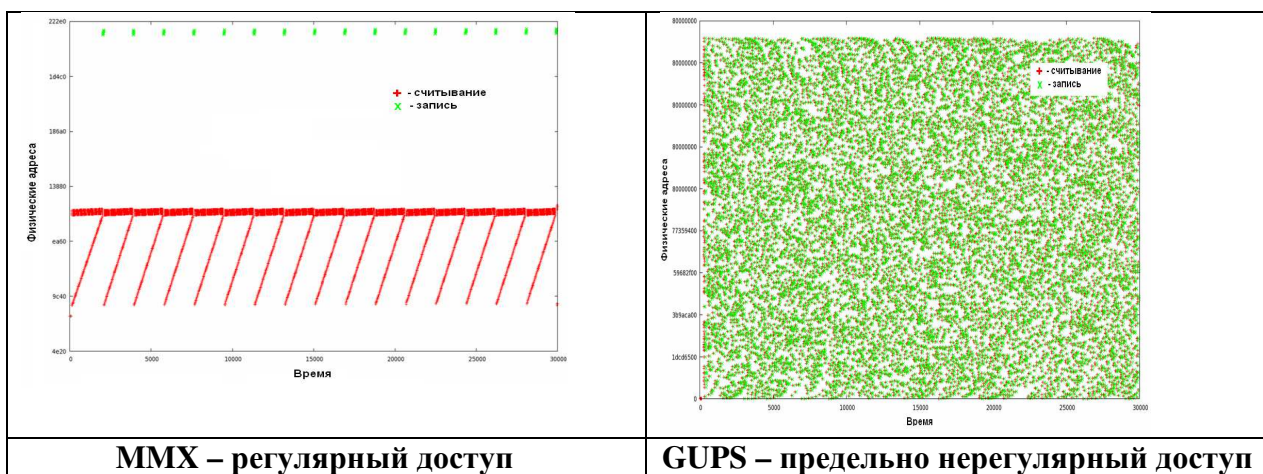


Рисунок 5.17 – Профиль обращений с регулярным доступом на задаче MMX умножения плотно заполненных матриц и предельно нерегулярный профиль обращений на специальном тесте RandomAccess (или другое название - GUPS).

Приведенные примеры разного по пространственно-временной работе с памятью задач и реальные характеристики работы оборудования в таких разных режимах приведены для того, чтобы проиллюстрировать, что на практике нужны разные типы суперкомпьютеров, например, для CF- и DIS-задач.

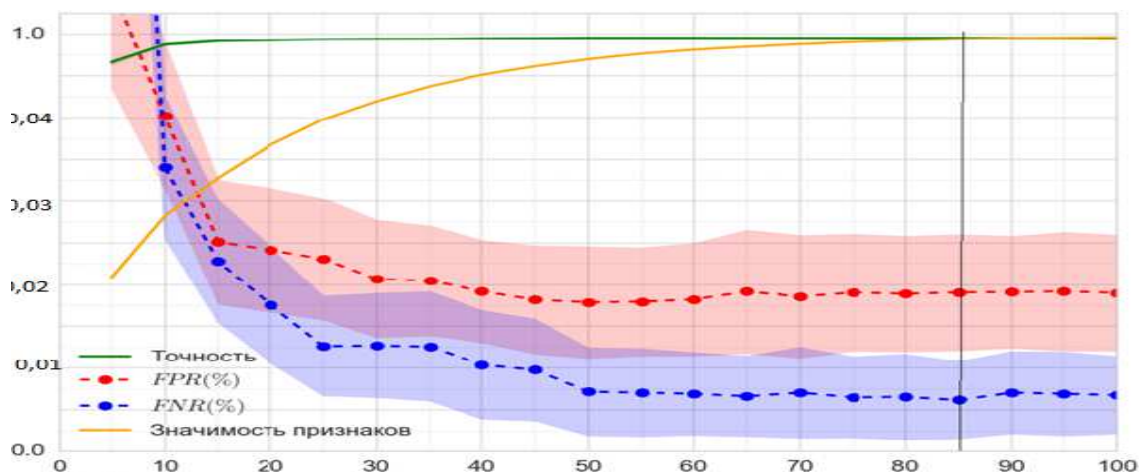


Рисунок 5.18 – Зависимость точности идентификации угроз от объема обучающей выборки

На рисунке 5.18 показано, что не менее чем в 86 из 100 обучающих выборок ошибки 1-го и 2-го рода не превышают 0,002. При увеличении выборок и интервала наблюдения по времени наблюдается устойчивое снижение: значение ложных срабатываний стремится к нулю.

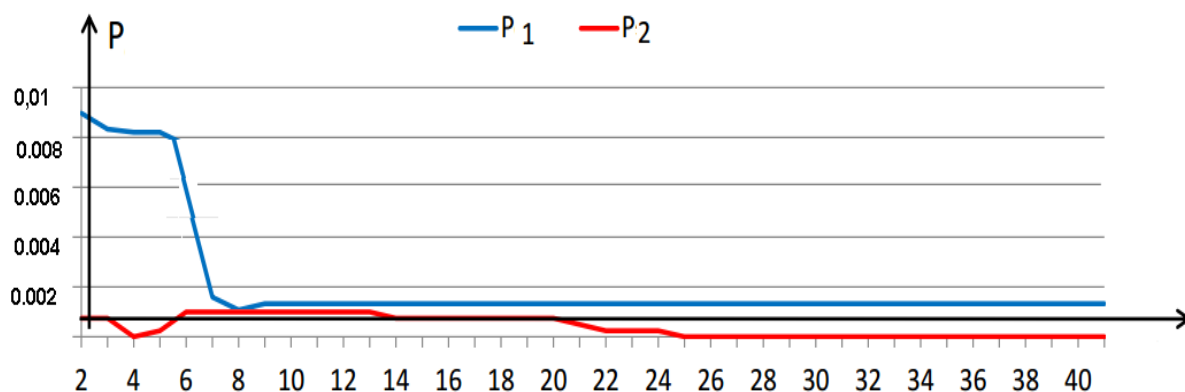


Рисунок 5.19 – Зависимость вероятности ошибок 1-го и 2-го рода от количества обучающих кластеров

На рисунке 5.19 показана зависимость ошибок 1-го и 2-го рода от количества обучающих кластеров. Для обучения и корректного распознавания требуется восемь обучающих доменов, поддерживаются любые форматы исполняемых файлов. Использование восьми кластеров для эффективного обнаружения вредоносного кода, что значительно сокращает расходы на реализации программно-технического решения.

Каждый кластер привязан к домену защиты. Их число тоже восьми. Таким образом, достигнуто однозначное соответствие число уровней иерархии числу доменов защиты и количеству счетных кластеров, привязанных к определенному уровню «песочницы».

В таблице 5.3 и 5.4 приведены результаты стендовых испытаний и сравнения функциональных возможностей четырех СЗИ: *Group-IB Polygon*, *Kaspersky KATA* и *PT Security Sandbox* (на основе марковских процессов с использованием алгоритма Баумана-Уэлча) и авторский продукт *Альфа-монитор* (с использованием алгоритма “маркерного” сканирования). Обобщая все сказанное выше, можно сделать вывод о том, что показатели защищенности суперкомпьютеров, на которых установлен защищенный мультидоменный гипервизор, наилучшие, а затраты на вычислительные ресурсы в 5-6 раз меньше, чем у существующих технологических решений на рынке СЗИ.

Таблица 5.3 – Результаты измерений

Класс суперкомпьютеров	Кол-во интервальных ограничений фильтрации М	Кол-во уровней иерархии выполнения запросов СК N	Вероятность ошибки идентификации угроз (взвешенные усредненные значения)	
			I рода	II рода
1. Ботовые супер-ЭВМ (сверхлегкие ядра)	3	4	0.01	0,012
2. Малый промышленный кластер (легкие ядра)	4	5	0.001	0, 003
3. Средний суперкластер (средние ядра)	6	7	0,0008	0,0007
4. Суперкластер стратегического класса (тяжелые ядра)	7	8	0,000035	0,000045

Таблица 5.4 – Результаты тестирования СЗИ

Название СЗИ	Число кластерн. зон для обучения	Макс. число доменов защиты	Число успешных распознаваний с мин. погрешн.	Погрешность	Эффективность (в процентах %)	Потери производ.
Group-IB Polygon	50	2	5 из 100	0,55	0,05	35-40 %
Kaspersky KATA	60	3	4 из 100	0,6	0,04	40-46 %
PT Security Sandbox	56	3	6 из 100	0,53	0,06	25-30 %
Альфа-монитор	8	8	86 из 100	0,0139	0,86	6-7 %

Существующие современные методики требуют не менее двадцати пяти обучающих доменов, при этом поддерживается только один РЕ-формат исполняемых файлов. Кроме того, во всех обучающих выборках (100 из 100) присутствуют ошибки 1-го и 2-го рода, которые превышают пороговое значение 0,5.

Следовательно, по сравнению с внедренными традиционными программно-техническими решениями требования к вычислительным ресурсам при реализации алгоритма “маркерного” сканирования в 5 – 6 раз меньше. Испытания на экспериментальном стенде показали минимальные потери производительности – менее 6-7 процентов.

5.5 Выводы по разделу:

1. Потеря производительности при реализации предложенной модели операций составляет менее 6-7 %;
2. Достаточно всего восьми кластеров для эффективного обнаружения вредоносного кода, что значительно сокращает расходы на реализации программно-технического решения;
3. Экспериментальные исследования показали, что не менее чем в 86 из 100 обучающих выборок ошибки 1-го и 2-го рода не превышают 0,01.
4. При увеличении выборок и интервала наблюдения по времени наблюдается устойчивое снижение: значение ложных срабатываний стремится к нулю;
5. В результате экспериментальных исследований получены новые научные результаты, подтверждающие эффективность и минимальную потерю производительности применения технологии аппаратной виртуализации в виде многоуровневой “песочницы” для перспективных суперкомпьютеров по сравнению с использованием традиционных кластеров;

6. Структура современных вычислительных центров стала гибридной, они стали состоять из суперкомпьютерных сегментов разного типа, но объединенные в единую инфраструктуру. Для их эффективного применения стали важными средства организации облачных вычислений и виртуальная инфраструктура, особенно с интеллектуальной организацией и загрузкой;
7. По сравнению с внедренными промышленными решениями требования к вычислительным ресурсам при реализации алгоритма “маркерного” сканирования в 5 – 6 раз меньше;
8. Эффективность обнаружения авторской методики на 86 % выше по сравнению с традиционными СЗИ.

ЗАКЛЮЧЕНИЕ

В результате выполнения исследования все поставленные задачи, связанные с разработкой моделей и методов успешно решены. Основные результаты и выводы следующие:

1. Впервые разработаны теоретические и научно-методические принципы защиты суперкомпьютеров с использованием средств виртуализации, основанные на построении и формализации универсальных и равномерных функций оценки безопасных состояний СК для заданных классов операций, что, в отличие существующих аксиоматик в классической теории информационной безопасности, позволяющие решить научную проблему супервентности. При этом получены важные теоретические выкладки. Доказана теорема о равномерной перечислимости функций СК.

2. Разработана модель угроз целостности среды выполнения процессов, основанная на новой парадигме написания правил политики безопасности, что, в отличие от существующих методик ФСТЭК и Росатома, позволяет учесть специфику работы стационарных и бортовых суперкомпьютеров. Вместо классической связки «субъект, объект, предикат» реализована новая парадигма написания правил политики безопасности и заданы новые сущности – «субъект», «объект» и «дескриптор оценки безопасности состояний»: для каждой i -ой угрозы вычисляется значение функции оценки безопасности состояний, аргументы которой заданы в виде конъюнкции предикатов из восьми логических переменных, субъектами являются гостевые и управляющие ОС, а объектами – компоненты гипервизора и контроллера транзакционной памяти.

3. Разработана модель безопасных операций, основанная на декомпозиции информационных процессов в виде 8-уровневой иерархической структуры, что, в отличие от классических моделей

безопасности дискреционного, мандатного, ролевого управления доступом, логического управления доступом и информационными потоками (ДП-моделей), в том числе иерархическим представлением мандатной сущностно-ролевой ДП-модели, позволяет идентифицировать и блокировать новый класс угроз, связанных с неоднозначностью состояний выполнения прикладных и системных программ на стационарных и бортовых суперкомпьютерах. Сформулировано условие разрешимости, как следствие из теоремы о равномерной перечислимости функций, проблемы установки логического соответствия (супервентности) процесса выполнения программы на суперкомпьютере, изменением состояний компонентов гипервизора и аппаратуры: при числе уровней иерархии меньше 8 процессы на всех уровнях в СК не наблюдаемы, что затрудняет обнаружение и идентификацию угроз со стороны СЗИ.

4. Разработан и исследован метод реактивной защиты суперкомпьютеров, основанный на виртуализации среды выполнения процессов, что, в отличие существующих классических методов управления доступом в изолированной программной среде, позволяет вычислять «свои» значения дескрипторов и задавать методы их обработки – get, set, delete и т.п.: вид характеристических функций при анализе траектории вычислений на структурах Крипке зависит от контекста входных данных и типа конфигурации СК. Ошибки 1-го и 2-го рода могут возникать на локальном i -ом уровне, но они детектируются и исправляются на уровне системной задача анализа контроля целостности поступающей информации с датчиков со всех уровней иерархии обработки запросов.

5. Разработан и исследован метод реконфигурации среды выполнения программ проактивной защиты суперкомпьютеров, основанный на парадигме темпоральной логики для проверки истинности принимаемых решений на структурах Крипке, что, в отличие от существующих классических методов мониторинга событий безопасности, позволяет

адаптировать число интерфейсных уровней к типу конфигурации СК и потоку обрабатываемых данных в режиме реального времени. Определены два направления реализации метода:

- для стационарных суперкомпьютеров заданы три конфигурации: число уровней защиты N от пяти до восьми. Предельное значение доверительной вероятности (при $N = 8$) приближается к 0,9. Показатель защищенности компонентов среды выполнения программ достигает максимума, когда система перестает быть уязвимой к любым видам воздействий и типам атак;
- для бортовых Супер-ЭВМ задана конфигурация минимальной защиты, в которой число уровней N равно четырем. Доверительная вероятность равна 0,8. Таким образом, при уменьшении числа доменов защиты, показатель защищенности компонентов среды выполнения программ по-прежнему остается высоким;
- показатель защищенности для традиционных СЗИ равен 0,5-0,6.

6. Разработана методика тестирования уровня защищенности суперкомпьютеров, основанная на алгоритме “маркерного” сканирования, что, в отличие от современных методик позволяет снизить затраты на использование вычислительных ресурсов (для обучения и корректного распознавания при использовании требуется восемь обучающих доменов, поддерживаются любые форматы исполняемых файлов), снизить кол-во ошибок и повысить эффективность идентификации атак и угроз. Традиционные методики и алгоритмы требуют не менее пятидесяти обучающих доменов, при этом поддерживается только один PE-формат исполняемых файлов. Следовательно, по сравнению с внедренными промышленными решениями требования к вычислительным ресурсам при реализации алгоритма “маркерного” сканирования в 5 – 6 раз меньше.

Экспериментальные исследования показали, что не менее чем в 86 из 100 обучающих выборок при использовании алгоритма “маркерного” сканирования ошибки 1-го и 2-го рода не превышают 0,012. При увеличении выборок и интервала наблюдения по времени наблюдается устойчивое снижение: значение ложных срабатываний стремится к нулю. Во всех обучающих выборках (100 из 100) существующих методик присутствуют ошибки 1-го и 2-го рода, которые превышают пороговое значение 0,5. Эффективность обнаружения авторской методики на 86 % выше по сравнению с традиционными СЗИ.

7. Разработаны и запатентованы программно-технические решения в виде многомодульного защищенного мультидоменного гипервизора, основанные на разработанных моделях и методах, что, в отличие существующих сертифицированных гипервизорных решений, позволяют получить наилучшие показатели производительности, показатели защищенности и не допускают аварийных завершений критических процессов из-за компьютерных атак. Порог потери производительности при использовании комплекса верификации менее 6-7 %. При количестве измерений равным восьми, получаемых с датчиков измерений на каждом уровне “песочницы”, показатель погрешности вычислений $\epsilon = 0,0139$. Подсистема обеспечения информационной безопасности реализована для российских суперкомпьютерных платформ семейства Ангара ЕС1740, для китайских суперкомпьютеров серии Tian-he/СТ-2, японских суперкомпьютеров серии Tsubame 3.0.

Рекомендации по применению включают в себя указания по использованию методов и методики тестирования уровня защищенности СК, а также разработку подходов, направленных на повышение эффективности процесса выявления угроз безопасности.

Научные и практические результаты могут быть использованы для широкого класса критически важных информационных и программных систем с гибкой конфигурацией окружения и контекстом вызовов сервисов, сбором метрики приложений на основе мультитредовой архитектуры с управлением потока данных.

В качестве перспектив дальнейшей разработки тематики следует указать исследования, связанные с развитием моделей описания информационных процессов и улучшения показателей идентификации вредоносного ПО. Также возможно использование описанных в работе моделей и методов в целевых программах и опытно-конструкторских работах силовых ведомств при разработке перспективных АС, ввод в промышленную эксплуатацию суперкомпьютера «Ангара» на базе отечественного процессора J7 в рамках национального проекта по импортозамещению.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

«Руткиты» – набор утилит или специальный модуль ядра, которые злоумышленник использует для скрытого встраивания в операционные системы пользователей вредоносного программного обеспечения

АНБ – Агентство национальной безопасности

АС – автоматизированная система

ВМ – виртуальная машина

ВПО – вредоносное программное обеспечение

ЗГОС – защищенная гипервизорная операционная система

ЗОС – защищенная операционная система

ИБ – информационная безопасность

КИИ – критическая информационная инфраструктура

МВД – Министерство внутренних дел

НИР – научно-исследовательская работа

НСД – несанкционированный доступ

ОБС – обманная система

ОКР – опытно-конструкторская работа

ООН – Организация Объединённых Наций

ОС – операционная система

ПБ – политика безопасности

ПЗУ – постоянное запоминающее устройство

ПЛИС – программируемые логические интегральные схемы

СВТ – средство вычислительной техники

СЗИ – система защиты информации

СК – суперкомпьютер

СУБД - системы управления базами данных

ТТХ – тактико-технические характеристики

ФСБ – Федеральная служба безопасности

ЦПУ – центральное процессорное устройство

ЦРУ – Центральное разведывательное управление

ЭВМ – электронно-вычислительная машина

ЮНЕКО – Организации Объединённых Наций по вопросам образования, науки и культуры.

CF-задачи – кеш-когерентные задачи

DIS -задачи – задачи с интенсивной нерегулярной работой с памятью

DTK –The Deception Toolkit (комплект инструментов обмана)

DKOM – технология манипулирования объектами ядра Windows NT без использования интерфейсных функций

GDT – глобальная дескрипторная таблица операционной системы

MITRE ATT&CK – Тактики, техники и общедоступные знания о злоумышленниках

SMM – System monitor mode (режим «прозрачного» контроля информационных потоков на уровне процессорного модуля с поддержкой виртуализации)

TSS – Task State Segment, сегмент состояния задачи

VICE – технология перехвата управления в отладочной сессии на реальных аппаратных модулях или на виртуальной машине

СПИСОК ЛИТЕРАТУРЫ

1. Критически важные объекты и кибертерроризм. Часть 1. Системный подход к организации противодействия // О.О. Андреев [и др.]. Под ред. В.А. Васенина. – М.: МЦНМО, 2008. – 198 с .
2. ГОСТ Р 51624–00. Защита информации. Автоматизированные системы в защищенном исполнении. Общие требования. – Введ. 30.06.00. Действ. – М.: Изд-во стандартов, 2000. – 12 с.
3. О безопасности критической информационной инфраструктуры Российской Федерации: Федеральный закон от 26 июля 2017 № 187-ФЗ // Собрание законодательства. – 2017. – № 31 (31.07). – ст. 4736.
4. Безкоровайный, М.М. Кибербезопасность – подходы к определению понятия / М.М. Безкоровайный, А.Л. Татузов // Вопросы кибербезопасности. – 2014. – №1. – С. 22-27.
5. Добродеев, А. Ю. Кибербезопасность в российской федерации. Модный термин или приоритетное технологическое направление обеспечения национальной и международной безопасности ххi века / А. Ю. Добродеев // Вопросы кибербезопасности. – 2021. – №4. – С. 61-70.
6. Зегжда, Д.П. Построение защищенных операционных систем на основе технологии виртуализации / Д.П. Зегжда. – Санкт-Петербург, 2018. – 33 с.
7. Зубков, Е.В. Методы интеллектуального анализа данных и обнаружение вторжений / Е. В. Зубков, В. М. Белов // Вестник СибГУТИ. – Новосибирск, 2016. – № 1. – С. 118-133.
8. Колесникова, С.И. Методы анализа информативности разнотипных признаков / С.И. Колесникова // Вестник Томского государственного университета: Управление, вычислительная техника и информатика. – 2009. – № 1 (6). – С. 69–80.
9. Кондратьев, А. А. Методологическое обеспечение интеллектуальных систем защиты от сетевых атак / А. А. Кондратьев, А. А. Талалаев, И. П. Тищенко,

- В. П. Фраленко, В. М. Хачумов // Современные проблемы науки и образования.– 2014. – № 2.
10. Патент № 2618367 С1 Российская Федерация, МПК G06F 12/02. вычислительное устройство программно-аппаратного комплекса : № 2016110564 : заявл. 23.03.2016 : опубл. 03.05.2017, Бюл. 13 / А. С. Моляков; заявитель и патентообладатель Моляков А.С. – 2 с. 1 ил.
 11. Патент № 2626350 С1 Российская Федерация, МПК G06F 12/02. Способ функционирования операционной системы вычислительного устройства программно-аппаратного комплекса : № 2016113545 : заявл. 11.04.2016 : опубл. 26.07.2017, Бюл. 21 / А. С. Моляков; заявитель и патентообладатель Моляков А.С. – 2 с. 1 ил.
 12. Сычев, А.М. Обоснование требований к межсетевым экранам и системам управления безопасностью в распределенных информационных систем: диссертация на соискание ученой степени кандидата технических наук: 05.13.19 / Сычев Артем Михайлович – М: МИФИ, 2002. – 149 с.
 13. Скиба, В.Ю. Модель адаптивного управления процессами обеспечения безопасности информации/ В.Ю. Скриба, Л.М. Ухлимов // Тезисы докладов научно–технической конференции в ВИККА им. А.Ф. Можайского.– СПб: Изд–во ВИККА им. А.Ф.Можайского, 1995. – С. 12-13.
 14. Хоффман, Л.Дж. Современные методы защиты информации: пер. с англ. / Л.Дж. Хоффман. – М.: Изд–во Сов. радио, 1989. – С. 50-52.
 15. Воробьев, А.А.. Анализ методов синтеза систем обнаружения вторжений / А.А. Воробьев, А.В. Непомнящих // Сб. материалов X республиканской научно – технической конференции Методы и технические средства. – СПб.: Изд–во Политех. Ун–та.– 2007. – С. 135-138.
 16. Воробьев, А.А. Методы адаптивного управления защищенностью автоматизированных систем / А.А. Воробьев // Техника – машиностроение. – 2001. – №1. – С. 156-157.

17. Molyakov, A. S. Model of hidden IT security threats in the cloud computing environment / A. S. Molyakov, V. S. Zaborovsky, A. A. Lukashin // Automatic Control and Computer Sciences. – 2015. – Vol. 49, No. 8. – P. 741-744. – DOI 10.3103/S0146411615080295.
18. Бородакий, Ю.В.. Кибербезопасность как основной фактор национальной и международной безопасности XXI века (Часть 1) / Ю.В. Бородакий, А.Ю. Добродеев, И.В. Бутусов // Вопросы кибербезопасности. 2013. № 1(1). С.2-9.
19. Приказ ФСТЭК России от 21 декабря 2017 г. N 235 «Об утверждении Требований к созданию систем безопасности значимых объектов критической информационной инфраструктуры РФ и обеспечению их функционирования».
20. Продуктовые линейки ViPNet [Электронный ресурс]. – Режим доступа: <http://www.infotecs.ru/products/line>, свободный. – Загл. с экрана (дата обращения: 25.11.2018).
21. Моляков, А.С. Модель угроз и теоретические основы метода реактивной защиты суперкомпьютеров / А.С. Моляков // Естественные и технические науки. - М.: Изд-во Компания Спутник +, 2019. – №7. – С. 197-201.
22. Моляков, А.С. Аксиоматика и принципы информационной безопасности суперкомпьютеров: учебно-методическое пособие. – М.: Изд-во Компания Спутник +, 2019. – 134 с.
23. Моляков, А. С. Маркерное сканирование как новый научный подход в области создания защищенных информационных систем на примере ОС нового поколения Microtek / А. С. Моляков // Проблемы информационной безопасности. Компьютерные системы. – 2016. – № 1. – С. 29-36.
24. Моляков, А.С. Модель безопасных операций и метод реконфигурации среды выполнения программ суперкомпьютеров / А.С. Моляков // Естественные и технические науки. – М.: Изд-во Компания Спутник +, 2019. – №7. – С. 193 – 196.
25. Моляков, А.С. Исследование ядра Windows NT на платформе Intel 3000: систематический поиск недеklarированных возможностей: монография / А.С. Моляков. – М: Изд-во Спутник Плюс, 2007. – 129 с.

26. Моляков, А. С. Модель скрытых угроз информационной безопасности в среде облачных вычислений / А. С. Моляков, В. С. Заборовский, А. А. Лукашин // Проблемы информационной безопасности. Компьютерные системы. – 2014. – № 2. – С. 41-46.
27. Моляков, А. С. KPROCESSOR_CID_TABLE -факторинг – новое направление в теории компьютерного анализа вирусного кода и программных закладок / А. С. Моляков // Проблемы информационной безопасности. Компьютерные системы. – 2009. – № 1. – С. 7-18.
28. Моляков, А. С. KPROCESSOR-CID-table-факторинг в области создания защищенных ОС нового поколения / А. С. Моляков // Безопасность информационных технологий. – 2009. – Т. 16, № 4. – С. 125-129
29. Моляков, А. С. Мультиграфовая модель операций для защиты среды облачных вычислений от скрытых угроз информационной безопасности / А. С. Моляков, В. С. Заборовский, А. А. Лукашин // Проблемы информационной безопасности. Компьютерные системы. – 2014. – № 2. – С. 37-40.
30. Горбунов, В. Тихоокеанско-азиатские петафлопсы / В. Горбунов, П. Забеднов, А. Моляков // Открытые системы. СУБД. – 2011. – № 7. – С. 26.
31. Архитектура фон Неймана [Электронный ресурс]. – Режим доступа: [http://ru.wikipedia.org/wiki/Архитектура фон Неймана](http://ru.wikipedia.org/wiki/Архитектура_фон_Неймана), свободный. – Загл. с экрана (дата обращения: 22.01.2019).
32. Моляков, А. С. Разработка прототипа вычислительного устройства с не фон-Неймановской архитектурой на базе отечественного микропроцессора стратегического назначения J7 / А. С. Моляков // Проблемы информационной безопасности. Компьютерные системы. – 2016. – № 1. – С. 22-28.
33. ICITE doing in common [Электронный ресурс]. – Режим доступа: <http://www.insaonline.org/CMDownload.aspx?ContentKey=b0443984-3e2a-4456-bbb9-cf570d61b7be&ContentItemKey=bdd62985-393b-4662-b422-8f79a840d67e>, свободный. – Загл. с экрана (дата обращения: 22.10.2019).

34. DARPA UHPC – дорога к эксафлопсам [Электронный ресурс]. – Режим доступа: <https://www.osp.ru/text/print/article/13005724.html>, свободный. – Загл. с экрана (дата обращения: 27.11.2019).
35. IDA Pro – интерактивный дизассемблер и отладчик [Электронный ресурс]. – Режим доступа: <http://www.idapro.ru/description>, свободный. – Загл. с экрана (дата обращения: 27.11.2020).
36. Свидетельство о государственной регистрации программы для ЭВМ № 2014616744. Альфа-монитор : № 2014613359 : заявл. 15.04.2014 / А. С. Моляков ; заявитель Моляков Андрей Сергеевич.
37. Слуцкий, А. Российский суперкомпьютер с глобально адресуемой памятью / А.И. Слуцкий, Л.К. Эйсымонт. – «Открытые системы», №9. – 2007. – С. 42-51.
38. Митрофанов В., Слуцкий А., Эйсымонт Л. Суперкомпьютерные технологии для стратегически важных задач. ЭЛЕКТРОНИКА: НТБ, 2008, №7, С.66-79.
39. Семенов А., Соколов А., Эйсымонт Л. Архитектура глобально адресуемой памяти мультитредово-поточкового суперкомпьютера. ЭЛЕКТРОНИКА: НТБ, №1, 2009, С.50-59.
40. Котенко, И.В. Новое поколение систем мониторинга и управления инцидентами безопасности / И.В. Котенко, И.Б. Саенко, Р.М. Юсупов // Научно-технические ведомости Санкт-Петербургского государственного политехнического университета. Информатика. Телекоммуникации. Управление.– 2014. – №. 3 (198).
41. Котенко, И.В. Построение модели данных для системы моделирования сетевых атак на основе онтологического подхода / И.В. Котенко, О.В. Полубелова, А.А. Чечулин // Труды СПИИРАН. – 2013. – Т. 3. – №. 26. – С. 26-39.
42. Казачок, А.В: Распознавание вредоносного программного обеспечения на основе скрытых марковских моделей [Электронный ресурс] – Режим доступа: <https://docplayer.ru/91734549-Raspoznavanie-vredonosnogo-programmnogo-obespecheniya-na-osnove-skrytyh-markovskih-modeley-k-t-n->

- kozachok-aleksandr-vasilevich.html, свободный. – Загл. с экрана (дата обращения: 02.09.2020).
43. Кульнев, Д. Защищенная ОС / Д.Кульнев, Р.Мадянов, А.Петрик // Открытые системы. СУБД. — 2015. — № 4. — С. 12-13
44. Molyakov, A.S. New security descriptor computing algorithm of Supercomputers / A. Molyakov // Proceedings of the 3rd World Conference on Smart Trends in Systems, Security and Sustainability, WorldS4 2019. – 2019. – P. 349-350. – Art. No 8903965. – DOI 10.1109/WorldS4.2019.8903965.
45. Моляков, А. С. Метод контроля отображения защищенных сегментов памяти при трансляции виртуальных адресов процессов ОС Windows / А. С. Моляков // Вопросы защиты информации. – 2008. – № 2(81). – С. 32-35.
46. Сайт Citiforum [Электронный ресурс]. – Режим доступа: http://citforum.ru/programming/digest/transactional_memory, свободный. – Загл. с экрана (дата обращения: 05.01.2020).
47. Моляков, А. С. Исследование скрытых механизмов управления задачами ядра Windows NT 5.1 / А. С. Моляков // Известия ЮФУ. Технические науки. – 2007. – № 4(76). – С. 139-147.
48. Щеглов К. А., Щеглов А. Ю. Марковские модели угрозы безопасности информационной системы // Изв. вузов. Приборостроение. 2015. Т. 58, № 12. С. 957—965.
49. Welcome to the DTK Software web site [Электронный ресурс]. – Режим доступа: <http://www.dtksoft.com>, свободный. – Загл. с экрана (дата обращения: 05.01.2020).
50. Rutkowska, J. Following the White Rabbit: Software Attacks against Intel VT-d technology [Электронный ресурс] / J. Rutkowska, R. Wojtczuk. – 2011. – Режим доступа: <http://www.invisiblethingslab.com/resources/2011/Software%20Attacks%20on%20Intel%20VT-d.pdf>, свободный. – Загл. с экрана (дата обращения: 18.08.2020).
51. Моляков, А. С. Новый метод систематического поиска недеklarированных возможностей ядра Windows NT 5.1 с введением контроля ContextHooking и

- PspCidHooking / A. С. Моляков // Вопросы защиты информации. – 2008. – № 1(80). – С. 49-56.
52. Molyakov, A. S. Model of hidden IT security threats in the cloud computing environment / A. S. Molyakov, V. S. Zaborovsky, A. A. Lukashin // Automatic Control and Computer Sciences. – 2015. – Vol. 49, No. 8. – P. 741-744. – DOI 10.3103/S0146411615080295.
53. Akyildiz I. F. et al. Wireless sensor networks: a survey // Computer networks. – 2002. – T. 38. – №. 4. – С. 393-422.
54. Almomani, I. WSN-DS: A Dataset for Intrusion Detection Systems in Wireless Sensor Networks / I. Almomani, B. Al-Kasasbeh, M. AL-Akhras // Journal of Sensors. – 2016. – 16 p.
55. Alrajeh, N. A. Intrusion Detection Systems in Wireless Sensor Networks: A Review / N.A. Alrajeh, S. Khan, B. Shams // International Journal of Distributed Sensor Networks. – 2013. – Vol. 9, no. 5. – 7 p.
56. Amini, M. Effective intrusion detection with a neural network ensemble using fuzzy clustering and stacking combination method / M. Amini, J. Rezaeenour, E. Hadavandi // Journal of Computing and Security. – 2014. – Vol. 1, no. 4. – P. 293-305.
57. Amudha, P. Classifier Model for Intrusion Detection Using Bio-inspired Metaheuristic Approach / P. Amudha, S. Karthik, S. Sivakumari // International Journal of Computer Science and Information Technologies. – Tamilnadu, India, 2014. – P. 7637-7642.
58. Anderson, J. P. [et al.] Computer security threat monitoring and surveillance / J. P. Anderson: tech. rep. – 1980.
59. Atkinson, D.C. Effective whole-program analysis in the presence of pointers. / D. C. Atkinson, W. G. Griswold // ACM SIGSOFT 1998 Symposium on the Foundations of Software Engineering. – 1998. – P. 46-55.
60. Baheti, R. Cyber-physical systems / R. Baheti, H. Gill // The impact of control technology. – 2011. – T. 12. – №. 1. – С. 161-166.

61. Bandy, M. T. Security in Context of the Internet of Things: A Study // Cryptographic Security Solutions for the Internet of Things. – IGI Global, 2019. – C. 1-40.
62. Barati M. et al. Distributed Denial of Service detection using hybrid machine learning technique // 2014 International Symposium on Biometrics and Security Technologies (ISBAST). – IEEE, 2014. – C. 268-273.
63. Barbara, D. Detecting novel network intrusions using bayes estimators / D. Barbara, N. Wu, S. Jajodia // In Proceedings of the 2001 SIAM International Conference on Data Mining. – SIAM. 2001. – P. 1-17.
64. Barford, P. Characteristics of Network Traffic Flow Anomalies / P. Barford, D. Plonka // In Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement. – ACM. 2001. – P. 66-73.
65. Baronti, P. Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards / P. Pillai, V. W. C. Chook, S. Chessa, A. Gotta, Y. Fun Hu // Computer Communications. – 2007. – P. 1655-1695.
66. Bella, G. Managing reputation over manets / G. Bella, G. Costantino, S. Riccobene // 2008 The Fourth International Conference on Information Assurance and Security. – IEEE, 2008. – C. 255-260.
67. Bhojannawar, S. S. Anomaly Detection Techniques for Wireless Sensor Networks – A Survey / S. S. Bhojannawar, C. M. Bulla, V. M. Danawade // International Journal of Advanced Research in Computer and Communication Engineering. – IEEE. 2010. – Vol. 2. – P. 3582-3857.
68. Boyce, C. A. P. A Comparison of Four Intrusion Detection Systems for Secure E-Business / C. A. P. Boyce, A. N. Zincir-Heywood // In Proceedings of the International Conference on Electronic Commerce Research. – 2003. – P. 302-314.
69. Branitskiy, A. Network attack detection based on combination of neural, immune and neuro-fuzzy classifiers / A. Branitskiy, I. Kotenko // In Proceedings of the 18th IEEE International Conference on Computational Science and Engineering (IEEE CSE2015). – IEEE. Oct. 2015. – P. 152-159.

70. Breiman, L. Random Forests / L. Breiman // Machine Learning. – 2001. Vol. 45, no. 1. – P. 5-32.
71. Brindasri, S. Evaluation of Network Intrusion Detection Using Markov Chain / S. Brindasri, K. Saravanan // International Journal on Cybernetics & Informatics (IJCI). – 2014. – Vol. 3, no. 2. – P. 11-20.
72. Burke, M. Flow-insensitive interprocedural alias analysis in the presence of pointers M. Burke, P. Carini, L. Choi, M. Hind // Lecture Notes in Computer Science. Springer –Verlag. – 1995. – P. 234-250.
73. Choi, J. Efficient flow-sensitive interprocedural computation of pointer induced aliases and side effects / J. Choi, M. Burke, P. Carini // In 20th Annual ACM SIGACT-SIGPLAN Symposium on the Principles of Programming Languages. – 1993. – V. 21. – P. 232-245.
74. Chae, Y. Trust Management for Defending On-off Attacks / Y. Chae, L.C. DiPippo, Y.L. Sun // Transactions on Parallel and Distributed Systems. – IEEE. 2015. – Vol. 26, no 4. – P. 1178-1191.
75. Computer Network Security and Privacy Protection, Home Land Security Report, Feb. 19. – 2010. – 20 p.
76. Chandrasekhar, A. M. Intrusion detection technique by using k-means, fuzzy neural network and SVM classifiers / A. M. Chandrasekhar, K. Raghuveer // In Proceedings of International Conference on Computer Communication and Informatics (ICCCI). – IEEE. 2013. – P. 1-7.
77. Chaudhari, H. Wireless Sensor Networks: Security, Attacks and Challenges / H. Chaudhari, L. Kadam // International Journal of Networking. – 2011. – Vol. 1, no 1. – P. 4-16.
78. Моляков, А. С. Супер-ЭВМ и операционные системы нового поколения / А. С. Моляков // Информационная безопасность: вчера, сегодня, завтра: Сборник статей по материалам Международной научно-практической конференции, Москва, 23 апреля 2019 года / Отв. ред. Н.В. Гришина. – Москва: Российский государственный гуманитарный университет, 2019. – С. 196-200..

79. Моляков, А. С. Наиболее распространённые угрозы безопасности АС / А. С. Моляков // Естественные и технические науки. – 2006. – № 6(26). – С. 254-256.
80. Моляков, А. С. Обнаружение скрытых процессов в 3 кольце защиты ОС Windows / А. С. Моляков // Естественные и технические науки. – 2006. – № 6(26). – С. 257-264..
81. Моляков, А. С. Новые методы поиска скрытых процессов ядра Windows NT 5.1 / А. С. Моляков // Программные продукты и системы. – 2007. – № 4. – С. 10.
82. Новый метод Branch History Injection (BHI) [Электронный ресурс]. – Режим доступа: <https://www.vusec.net/projects/bhi-spectre-bhb/>, свободный. – Загл. с экрана (дата обращения: 05.01.2020).
83. Центральный процессор «Эльбрус» (ТВГИ.431281.006) [Электронный ресурс]. – Режим доступа: <http://www.mcst.ru/elbrus>, свободный. – Загл. с экрана (дата обращения: 05.01.2020).
84. Грушо, А. А. Исследование проектов верхнего и нижнего уровней политики безопасности ОС Windows / А. А. Грушо, А. С. Моляков // Программные продукты и системы. – 2007. – № 4. – С. 45-47.
85. «Ростех» совместно с «Каскадом» и МТУСИ создают отечественные защищенные криптопроцессоры [Электронный ресурс]. – Режим доступа: https://www.cnews.ru/news/line/2022-03-09_rosteh_sovmestno_s_kaskadom, свободный. – Загл. с экрана (дата обращения: 05.01.2020).
86. Molyakov A. Japanese Strategic Supercomputer “Arrow of Time” /A.S. Molyakov // Biomedical Journal of Scientific & Technical Research 45(2), 36348 - 36351, 2022. – DOI: 10.26717/BJSTR.2022.45.007183.
87. Molyakov, A. Mathematical Modeling of Neurodynamic Systems: Solving DIS-Tasks Using Massive-Multithread Supercomputers / A. Molyakov // Biomedical Journal of Scientific and Technical Research. – 2019. – Vol. 21, No. 5. – P. 16159-16162. – DOI 10.26717/BJSTR.2019.21.003657.

88. Molyakov, A. Revolutionary Direction of Secured Supercomputer Technologies Development in Russia. In Proceedings of the International Scientific and Practical Conference on Computer and Information Security (INFSEC). – ISBN 978-989-758-531-9, 5-11, 2022. – DOI: 10.5220/0010616500003170.
89. Molyakov, A. China Net: Military and Special Supercomputer Centers / A.S. Molyakov // Journal of Electrical and Electronic Engineering 7 (4), 95-100, 2019. – DOI: 10.11648/j.jeee.20190704.12.
90. Molyakov, A. Age of Great Chinese Dragon: Supercomputer Centers and High Performance Computing / A.S. Molyakov // Journal of Electrical and Electronic Engineering 7 (4), 87-94, 2019. – DOI: 10.11648/j.jeee.20190704.11.
91. Molyakov, A. Specialized Hybrid Chinese Microprocessors for Solving Important Problems of Fundamental Medicine and National Defense / A.S. Molyakov // Biomedical Journal of Scientific & Technical Research 34(4), 26876-26881, 2021. – DOI: 10.26717/BJSTR.2021.34.005573.
92. Molyakov, A. The Information and PSY Wars of the Future: Chinese Cyber Troops / A.S. Molyakov // Biomedical Journal of Scientific & Technical Research 33(2), 25603 -25608, 2021. – DOI: 10.26717/BJSTR.2021.33.005362.
93. Molyakov, A. S. A prototype computer with non-von Neumann architecture based on strategic domestic J7 microprocessor / A. S. Molyakov // Automatic Control and Computer Sciences. – 2016. – Vol. 50, No. 8. – P. 682-686. – DOI 10.3103/S0146411616080137.
94. Molyakov, A. S. Token scanning as a new scientific approach in the creation of protected systems: A new generation OS MICROTEK / A. S. Molyakov // Automatic Control and Computer Sciences. – 2016. – Vol. 50, No. 8. – P. 687-692. – DOI 10.3103/S0146411616080149.
95. Molyakov, A. S. New Multilevel Architecture of Secured Supercomputers / A. S. Molyakov // Current Trends in Computer Sciences & Applications. – 2019. – Vol. 1, No. 3. – P. 57-59. – DOI 10.32474/CTCSA.2019.01.000112.
96. Molyakov, A. Based on Reconfiguring the Supercomputers Runtime Environment New Security Methods [Электронный ресурс] / A.S. Molyakov // Advances in

- Science Technology and Engineering Systems Journal, vol. 5, no. 3, 291-298, 2020.
97. Моляков, А. С. Инновационный вариант развития защищенных супер-ЭВМ для решения важных задач фундаментальной медицины и инженерии в России / А. С. Моляков // Современная наука: актуальные проблемы теории и практики. Серия: Естественные и технические науки. – 2022. – № 4-2. – С. 88-93. – DOI 10.37882/2223-2966.2022.04-2.26.
 98. Molyakov, A.S. Secured Supercomputer Technologies in Russia: Functional Computing Units Based on Multithread-Stream Cores with Specialized Accelerators. / A.S. Molyakov // Proceedings of Seventh International Congress on Information and Communication Technology. Lecture Notes in Networks and Systems. – 2023. – vol 448. – P. 559 - 566. – DOI:10.1007/978-981-19-1610-6_49.
 99. Molyakov, A. Science Innovation Review: Developers of Supercomputers in Russia / A. Molyakov // Journal of Electrical and Electronic Engineering. – 2019. – Vol. 7, No. 5. – P. 107-112. – DOI 10.11648/j.jeee.20190705.12.
 100. Molyakov, A. New-age Supercomputers: Hi-Speed Networks and Information Security / A. Molyakov // Journal of Electrical and Electronic Engineering. – 2019. – Vol. 7, No. 3. – P. 82-86. – DOI 10.11648/j.jeee.20190703.12
 101. Garfinkel, T. Ostia: A delegating architecture for secure system call interposition [Электронный ресурс] / Т. Garfinkel. – Режим доступа: <http://benpfaff.org/papers/ostia.pdf>, свободный. – Загл. с экрана (дата обращения: 09.07.2020).
 102. Garfinkel, T. Traps and Pitfalls: Practical Problems in System Call Interposition Based Security Tools [Электронный ресурс] / Т. Garfinkel. – Режим доступа: <https://crypto.stanford.edu/cs155/papers/traps.pdf>, свободный. – Загл. с экрана (дата обращения: 15.03.2019).
 103. Viega, J. ITS4: Vulnerability Scanner for C and C++ Code / J. Viega, J. Bloch, T. Kohno // In Proceedings of the 16th Annual Computer Security Applications Conference . –2000. – P. 257 – 267.

104. Howard, J.D. A Common Language for Computer Security / J.D.Howard, T.Longstaff. Incidents // SANDIA Report. – 2008. – 120 p.
105. Computer Network Security and Privacy Protection, Home Land Security Report, Feb. 19. – 2010. – 20 p.
106. Defending Computer Networks against Attack, S&TR, Cyber Security, Feb. – 2011. – 17 p.
107. Chen, H. et al. A Novel Low-Rate Denial of Service Attack Detection Approach in ZigBee Wireless Sensor Network by Combining Hilbert-Huang Transformation and Trust Evaluation //IEEE Access. – 2019. – T. 7. – C. 32853-32866.
108. Hodo, A. Shallow and Deep Networks Intrusion Detection System: A Taxonomy and Survey / Bellekens, A. Hamilton, C. Tachtatzis, R. Atkinson [Электронный ресурс]. – 2017. – URL: <https://www.semanticscholar.org/paper/Shallow-and-Deep-Networks-Intrusion-Detection-A-and-Hodo-ellekens/193a4f04b007840571c96ac9b84b09f215063c97> (visited on 26.10.2020).
109. Hu, H. Security Metric Methods for Network Multistep Attacks Using AMC and Big Data Correlation Analysis / H. Hu, Y. Liu, H. Zhang, and Y. Zhang // Security and Communication Networks. – 2018. – Vol. 2018. – 14 p.
110. Gupta, A. Performing an IoT Pentest //The IoT Hacker's Handbook. – Apress, Berkeley, CA, 2019. – P. 17-37.
111. Popek G. J., Goldberg R. P. Formal Requirements for Virtualizable Third Generation Architectures // Communications of the ACM. — 1974.
112. Intel® Virtualization Technology / F. Leung, G. Neiger, D. Rodgers et al. // Intel Technology Journal. — 2006. — August. — Vol. 10.
113. AMD Corporation. — AMD64 Architecture Programmer's Manual Vol-ume 2: System Programming, 2013.
114. Implementing AMD's Advanced Synchronization Facility in an out-of-order x86 core / Stephan Diestelhorst, Martin Pohlack, Michael Hohmuth et al. // 5th ACM SIGPLAN Workshop on Transactional Computing. — 2010.

115. Intel® Software Development Emulator. — 2012. — Режим доступа: <https://software.intel.com/en-us/articles/intel-software-development-emulator>, свободный. — Загл. с экрана (дата обращения: 15.03.2019).
116. Rechistov Grigory, Plotkin Arnold. Implementation of Intel Restricted Transactional Memory ISA Extension in Simics // *Procedia Computer Science*. — 2013. — Vol. 18, no. 0. — P. 1804 - 1813.
117. Herlihy Maurice, Moss J. Eliot B. Transactional memory: architectural support for lock-free data structures // *Proceedings of the 20th annual international symposium on computer architecture*. — ISCA '93. — New York, NY, USA: ACM, 1993. — P. 289-300.
118. Evaluation of Blue Gene/Q Hardware Support for Transactional Memories / Amy Wang, Matthew Gaudet, Peng Wu et al. // *Proceedings of the 21st International Conference on Parallel Architectures and Compilation Techniques*. — PACT '12. — 2012. — P. 127-136.
119. Jacobi Christian, Slegel Timothy, Greiner Dan. Transactional Memory Architecture and Implementation for IBM System Z // *Proceedings of the 2012 45th Annual IEEE/ACM International Symposium on Microarchitecture*. — 2012. — 12. — P. 25-36.
120. ASF: AMD64 Extension for Lock-free Data Structures and Transactional Memory / Jaewoong Chung, Stephan Diestelhorst, Martin Pohlack et al. // *Proceedings of the 2010 43rd Annual IEEE/ACM International Symposium on Microarchitecture*. — 2010.
121. Moir Mark, Moore Kevin, Nussbaum Dan. The adaptive transactional memory test platform: a tool for experimenting with transactional code for ROCK (poster) // *Proceedings of the twentieth annual symposium on Parallelism in algorithms and architectures*. — SPAA '08. — New York, NY, USA: ACM, 2008. — P. 362-362.
122. Software Transactional Memory for Dynamic-sized Data Structures / Maurice Herlihy, Victor Luchangco, Mark Moir, William N. Scherer, III // *Proceedings of*

- the Twenty-second Annual Symposium on Principles of Distributed Computing. — PODC '03. — New York, NY, USA: ACM, 2003. — P. 92-101.
123. Dice Dave, Shalev Ori, Shavit Nir. Transactional Locking II // Proceedings of the 20th International Conference on Distributed Computing. — DISC'06. — Berlin, Heidelberg: Springer-Verlag, 2006. — P. 194-208.
 124. Multifacet's general execution-driven multiprocessor simulator (GEMS) toolset / Milo M. K. Martin, Daniel J. Sorin, Bradford M. Beckmann et al. // SIGARCH Comput. Archit. News. — 2005. — Nov. — Vol. 33, no. 4. — P. 92-99.
 125. Efficient Transaction Nesting in Hardware Transactional Memory / Yi Liu, Yangming Su, Cui Zhang et al. // Architecture of Computing Systems - ARCS 2010. — Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. — P. 138-149.
 126. Berkeley Lab APEX – Map V3.0 [Электронный ресурс]. – Режим доступа: <http://crd.lbl.gov/groups-depts/ftg/projects/previous-projects/apex/apex-map-v3-0>, свободный. – Загл. с экрана (дата обращения: 25.03.2020).
 127. Приказ ФСТЭК от 25 декабря 2017 г. N 239 “Об утверждении требований по обеспечению безопасности значимых объектов критической информационной инфраструктуры российской федерации” (в ред. приказов фстэк россии от 9 августа 2018 г. n 138, от 26 марта 2019 г. n 60, от 20 февраля 2020 г. n 35). – Режим доступа: <https://fstec.ru/tekhnicheskaya-zashchita-informatsii/obespechenie-bezopasnosti-kii/288-prikazy/1592-prikaz-fstek-rossii-ot-25-dekabrya-2017-g-n-239> . – Загл. с экрана (дата обращения: 25.03.2022).
 128. Методический документ. Методика оценки угроз безопасности информации" (утв. ФСТЭК России 05.02.2021). – Режим доступа: <https://fstec.ru/tekhnicheskaya-zashchita-informatsii/dokumenty/114-spetsialnye-normativnye-dokumenty/2170-metodicheskij-dokument-utverzhdenn-fstek-rossii-5-fevralya-2021> – Загл. с экрана (дата обращения: 25.03.2022).

ПРИЛОЖЕНИЕ 1

СВИДЕТЕЛЬСТВО О РЕГИСТРАЦИИ ПРОГРАММЫ ЭВМ

РОССИЙСКАЯ ФЕДЕРАЦИЯ



СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

№ 2014616744

Альфа-монитор

Правообладатель: *Моляков Андрей Сергеевич (RU)*Автор: *Моляков Андрей Сергеевич (RU)*

Заявка № 2014613359

Дата поступления 15 апреля 2014 г.

Дата государственной регистрации

в Реестре программ для ЭВМ 03 июля 2014 г.

Руководитель Федеральной службы
по интеллектуальной собственности

Б.П. Симонов

ПРИЛОЖЕНИЕ 2
ПАТЕНТЫ НА ИЗОБРЕТЕНИЯ

РОССИЙСКАЯ ФЕДЕРАЦИЯ



ПАТЕНТ

НА ИЗОБРЕТЕНИЕ

№ 2618367

**ВЫЧИСЛИТЕЛЬНОЕ УСТРОЙСТВО ПРОГРАММНО-
АППАРАТНОГО КОМПЛЕКСА**

Патентообладатель: *Моляков Андрей Сергеевич (RU)*

Автор: *Моляков Андрей Сергеевич (RU)*

Заявка № 2016110564

Приоритет изобретения 23 марта 2016 г.

Дата государственной регистрации в

Государственном реестре изобретений

Российской Федерации 03 мая 2017 г.

Срок действия исключительного права
на изобретение истекает 23 марта 2036 г.

Руководитель Федеральной службы
по интеллектуальной собственности

Г.П. Ивлиев



РОССИЙСКАЯ ФЕДЕРАЦИЯ



ПАТЕНТ

НА ИЗОБРЕТЕНИЕ

№ 2626350

СПОСОБ ФУНКЦИОНИРОВАНИЯ ОПЕРАЦИОННОЙ
СИСТЕМЫ ВЫЧИСЛИТЕЛЬНОГО УСТРОЙСТВА
ПРОГРАММНО-АППАРАТНОГО КОМПЛЕКСАПатентообладатель: *Моляков Андрей Сергеевич (RU)*Автор: *Моляков Андрей Сергеевич (RU)*

Заявка № 2016113545

Приоритет изобретения 11 апреля 2016 г.

Дата государственной регистрации в

Государственном реестре изобретений

Российской Федерации 26 июля 2017 г.

Срок действия исключительного права

на изобретение истекает 11 апреля 2036 г.

Руководитель Федеральной службы
по интеллектуальной собственности

Г.П. Ивлиев



ПРИЛОЖЕНИЕ 3

СХЕМА И НАБОР ИНСТРУМЕНТАЛЬНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ЭКСПЕРИМЕНТАЛЬНОГО СТЕНДА

В состав экспериментального стенда входят:

- 1) 50 сервисных узлов и узлов подсистемы хранения данных суперкомпьютера;
- 2) сервера управления суперкомпьютером для работы администраторов;
- 3) 30 вычислительных узлов суперкомпьютера;
- 4) 20 рабочих станций для операторов (клиентов) и обслуживающего технического персонала;
- 5) ПО Qemu, обеспечивающее виртуализацию ресурсов компьютера;
- 6) Модули виртуализации (контейнирования) KVM, XEN и LXC;
- 7) сетевая отказоустойчивая файловая система Ceph для создания распределенных хранилищ;
- 8) служба Нарпоху, предназначенная для обеспечения высокой доступности и отказоустойчивости TCP- и HTTP-соединений посредством перераспределения входящих запросов на обслуживание нескольких серверов;
- 9) Kaspersky Security Center;
- 10) NortonSecurity Cluster Center;
- 11) VIP NET Cluster Firewall;
- 12) Альфа-монитор.

В составе комплекса используются система резервного копирования и восстановления данных Acronis Backup & Recovery 11 и отечественное сертифицированное ПО RuBackup версии 1.4.

Базовый блок . Вычислительные узлы суперкомпьютера

Вычислительный блок с графическими процессорами с четырьмя многосокетными платами с разными типами микропроцессоров (гибридная конфигурация). На стенде используются как зарубежные процессоры, так и отечественные вычислители (ускорители) на базе процессоров Эльбрус, Байкал, Комдив и блоки (юниты) для проведения имитационных и эмуляционных исследований на базе Ангара-J7.

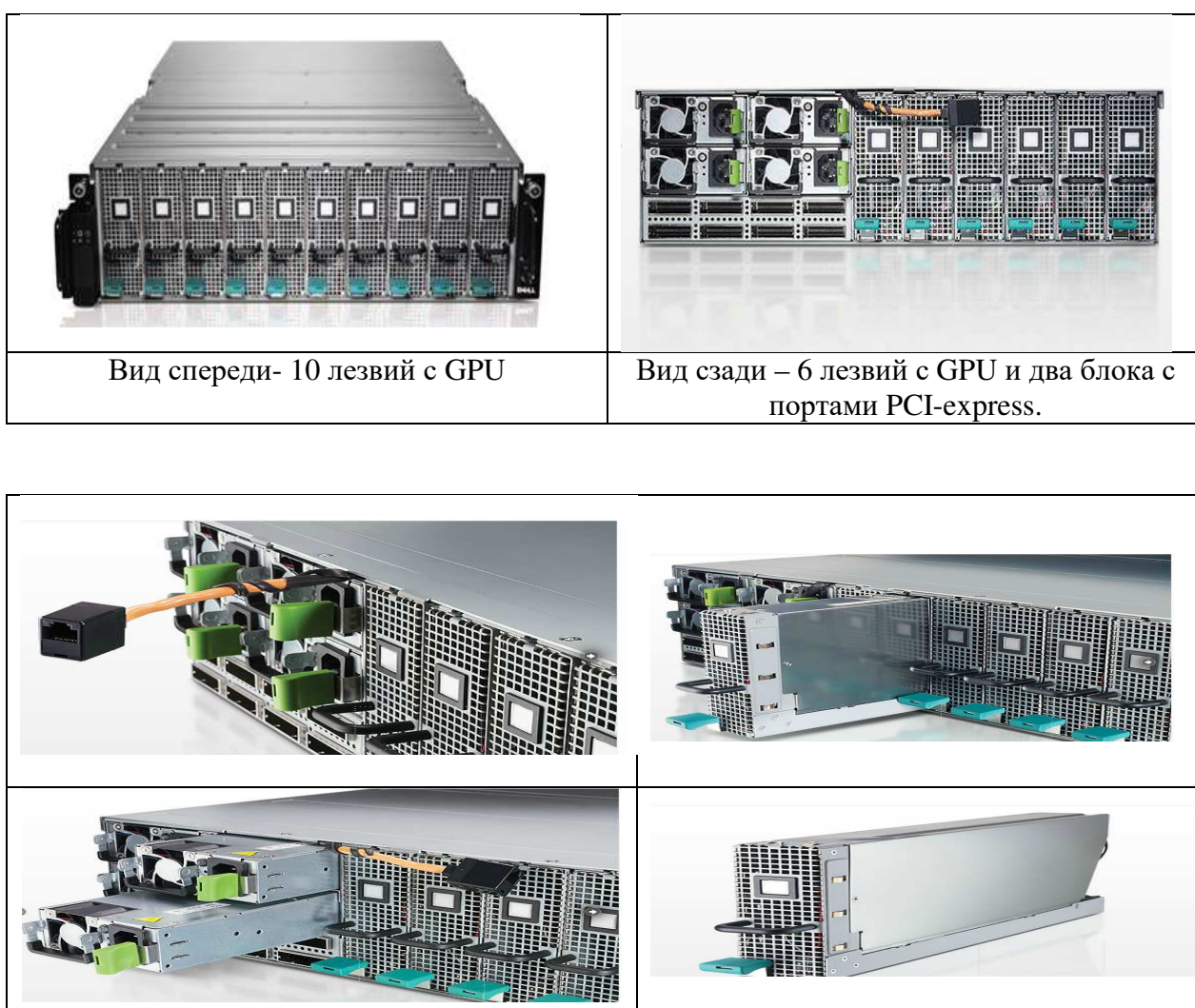


Рисунок 1 – Вид ускорительного блока вычислительного узла с 16 блоками с GPU и двумя блоками с 8-ю портами PCI-express.

Сервисные узлы суперкомпьютера



Рис. 2. Блейд-сервер управляющего узла

- Компактный двухсокетный (четырехсокетный) сервер.
- Подходит как для высокопроизводительных вычислений, так и в качестве сервера для ЦОД.
- Содержит контроллер управления
- Устанавливается в стойки и шасси РСК Торнадо, занимает 1 установочное место.
- Избыточность по питанию обеспечивается установкой модуля резервного питания

Подсистема мониторинга (на основе Zabbix 4.2) – набор программно-аппаратных средств, выполняющих следующие функции:

Контроль и отображение загрузки вычислительных ресурсов, сети, подсистемы хранения в реальном времени;

Контроль и отображение климатических (температура, влажность и т.д.) режимов работы оборудования;

Контроль и отображение параметров системы первичного питания и внутренних блоков питания, управление вкл./выкл. компонентов системы.

Ведение истории собранной статистики;

Корректное завершение работы и выключение контролируемых технических средств при обнаружении аварий в системе охлаждения или электропитания.

Общее программное обеспечение.

- Для открытого сегмента операционная система – CentOS / Debian/ Microsoft 2016 Server, для сегмента по работе с объектами КИИ – специализированные сборки ОС для оборонных предприятий.
 - Набор драйверов;
 - Инструментальное ПО;
 - Компиляторы C/C++ – GCC, ICC, Open64;
 - Библиотеки MPI – intelMPI, openMPI, mvapich2;
 - Средства отладки и профилирования Total View.

Система управления прохождением задач – предоставляет возможность запуска пользовательских однопоточных и параллельных задач (MPI, SHMEM, OpenMP) из очереди по настраиваемым критериям приоритетности. Обеспечивает настройку распределения машинных ресурсов между вычислительными процессами и задачами. Выполняет сбор статистики и сохранение истории запусков задач и использованных ресурсов.

Тестовое ПО – набор тестовых программ, применяемых для оценки производительности вычислительной системы и отдельных компонентов на различных классах задач, а также измерения характеристик сети и подсистемы памяти.

ПО виртуализации – набор программного обеспечения, обеспечивающий сетевой доступ к вычислительным ресурсам. Система виртуализации позволяет выполнять распределение нагрузки посредством выделения и программного распределения виртуальных серверов по реальным вычислительным ресурсам.

Совместимость: системы на основе универсальных микропроцессоров x86, x86-64 (Intel, AMD), Эльбрус, Байкал, GPU NVIDIA/AMD 2. Специализированные системы на основе FPGA.

Коммуникационная сеть суперкомпьютера:

Топология от «1D-mesh» до «4D-top».

Предусмотрено объединение всех установленных в стойку серверов управления и вычислительных узлов в две независимые сети — сеть управления и сеть приложений. Для этого в задней части стойки размещаются 6 коммутаторов Ethernet до 52 портов каждый, скорость сети — до 10 гбит/с.

Конструкция стойки спроектирована для максимально плотного размещения кабелей для подключения всех серверов к сети управления, сети приложений и высокопроизводительной высокоскоростной сети Intel Omni-Path или Mellanox InfiniBandEDR/HDR100.

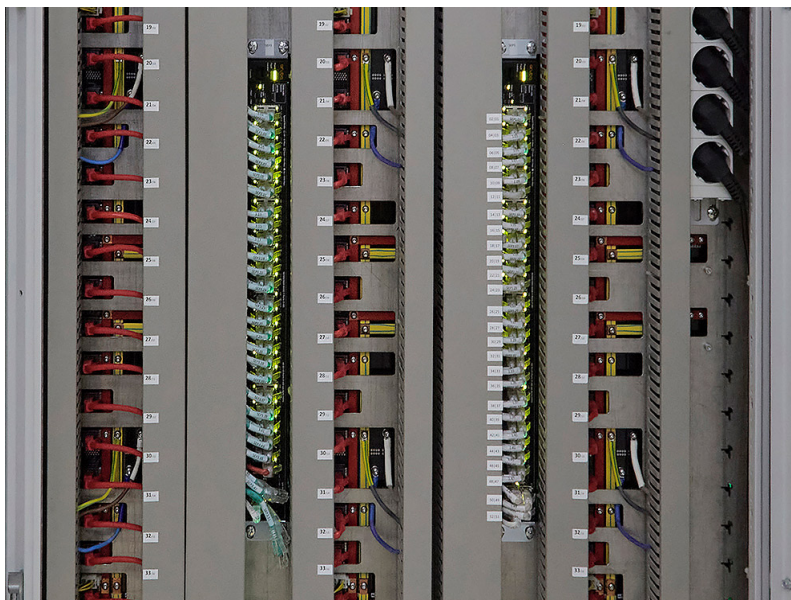


Рисунок 3 — Ethernet коммутаторы внутри стойки в машинном зале

ПРИЛОЖЕНИЕ 4

СПРАВКИ И АКТЫ О ВНЕДРЕНИИ

**Реализация и испытания программно-технических решений для
российских суперкомпьютерных платформ семейства Ангара ЕС 1740**

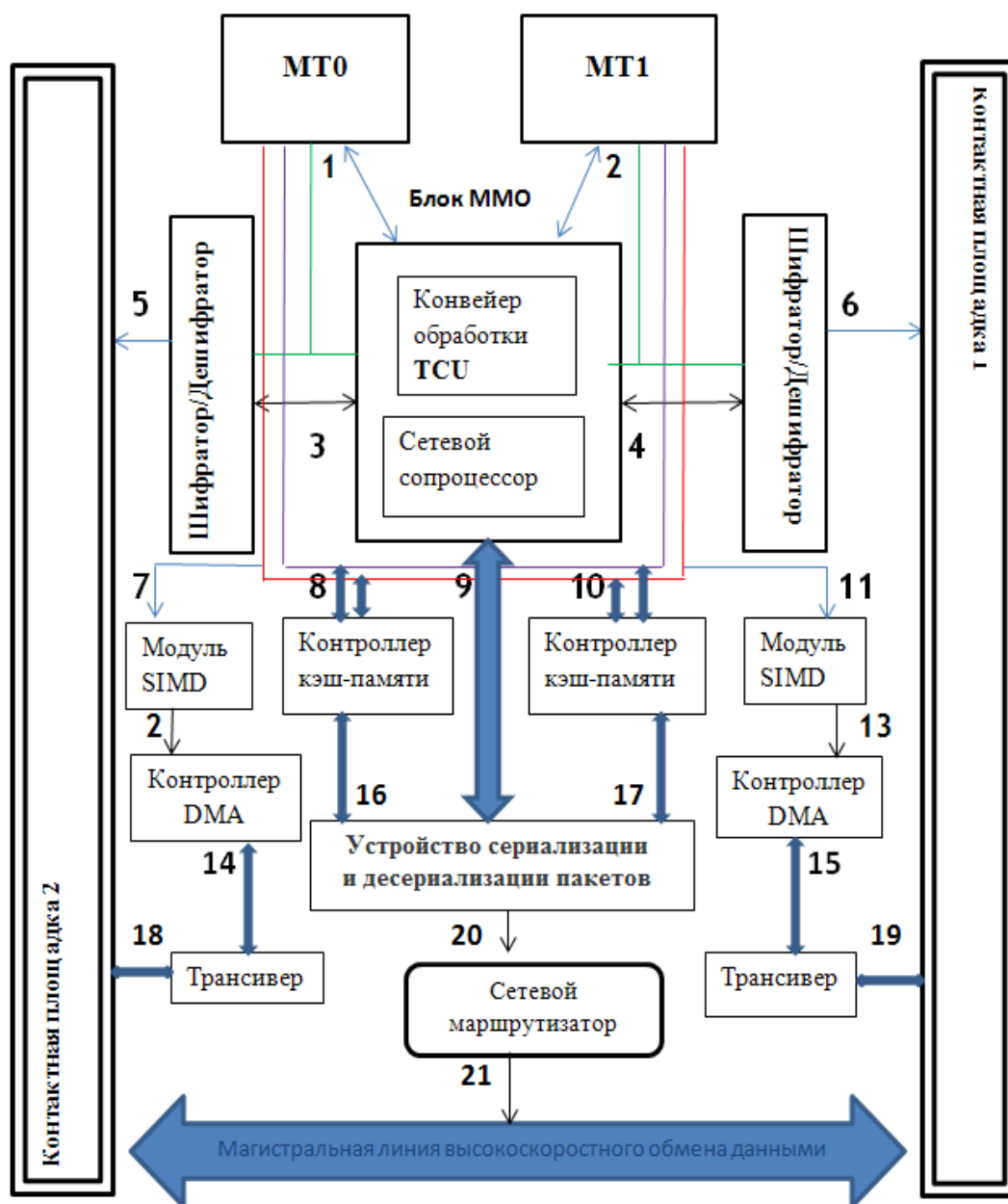


Рисунок 4 – Макет устройства в защищенном исполнении



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Российский государственный гуманитарный университет»
(РГГУ)

Минусская пл., д. 6, Москва, ГСП-3, 125993. Тел. (495) 250-61-18. Факс (495) 250-51-09.
E-mail: rsuh@rsuh.ru; <http://www.rggu.ru>. ОКПО 02068746. ОГРН 1037700067118. ИНН/КПП 7707033405/770701001.

УТВЕРЖДАЮ

Проректор по научной работе
канд. истор. наук., проф.
О.В. Павленко



2018 г.

СПРАВКА

О внедрении результатов диссертационной работы
на соискание ученой степени
доктора технических наук
Молякова Андрея Сергеевича

Результаты диссертационной работы Молякова Андрея Сергеевича «Модели и методы обеспечения информационной безопасности стационарных и бортовых суперкомпьютерных вычислительных систем», представленной на соискание докторской степени технических наук, успешно использованы при разработке вычислительного модуля программно-аппаратного комплекса «Альфа-монитор» (свидетельство о регистрации Роспатента № 2014616744, Патент на изобретение

«Вычислительное устройство программно-аппаратного комплекса» № 2618367 от 03.05.2017, Патент на изобретение «Способ функционирования операционной системы вычислительного устройства программно-аппаратного комплекса» № 2626350 от 26.06.2017).

В прототипе вычислительного устройства используются средства работы с очень большими данными.

Переходя на уровень инженерных решений, можно сказать, что на задачах с хорошей пространственно-временной работой с памятью могут эффективно использоваться кэш-памяти и схемы автоматической преднагрузки данных в кэш-память. Это позволяет значительно сократить среднее время обращения к памяти нескольких сотен тактов процессора до долей такта процессора

В рамках инженерных экспериментальных исследований был разработан **многокомпонентный защищенный мультидоменный гипервизор** в нескольких вариантах исполнений: программном и программно-аппаратном.

Подсистема обеспечения информационной безопасности (ПОИБ) суперкомпьютеров (СК) серии «Ангара» ЕС1740 реализована с поддержкой массово-мультитредовых гибридных реконфигурируемых вычислителей Ангара-J7 секционного типа на базе технологических модулей «Альфа-монитор», которые состоят из монитора безопасности и гипервизорных модулей (проксирующих модулей и верификатора команд). Она реализована на базе мультидоменного гипервизора и коммутаторов Ангара-МВС. Управление сервисными и вычислительными узлами, объединёнными высокоскоростными линиями сети «Ангара», осуществляется с помощью защищенного распределенного мультидоменного гипервизора. Суперкомпьютерные вычислительные платформы серии ЕС1740 с поддержкой сети коммутаторов 2-го поколения и масштабируемых высокопроизводительных кластеров реализованы АО «НИЦЭВТ» в рамках ОКР (шифр «Ангара-СК») и запущены в серийное производство в 2018 году.

Суперкомпьютерные вычислительные платформы серии «Ангара» ЕС1740 представляют собой множество узлов разного типа, объединённых несколькими коммуникационными сетями, одна из которых обладает свойством передачи с высокой пропускной способностью больших потоков коротких пакетов. Эта сеть необходима для реализации работы с глобально адресуемой памятью. Узлы могут быть вычислительными и сервисными, они подключаются к базовой рабочей сети.

Вычислительные узлы строятся на специальных многоядерных мультитредово-поточковых микропроцессорах, объединяются в модули в виде многосокетных плат и могут работать над логически единой адресуемой памятью (глобально адресуемой памятью).

Сервисные узлы строятся на обычных суперскалярных микропроцессорах, выполняют функции ввода-вывода, подключения пользователей, интерфейса с глобальной сетью, а также могут выполнять и вычисления, если они хорошо локализуются и эффективно выполнимы на этих узлах. Вычислительные и сервисные узлы подключены ещё к одной сети, являющейся компонентом подсистемы обеспечения надежности, готовности и сервиса, RAS-подсистемы.

Ключевые особенности/преимущества способа функционирования ПОИБ:

- Реализована мультидоменная защита;
- Введена поддержка восьми уровней привилегий выполнения команд, что позволяет реализовать многоуровневую («усиленную») ролевую политику безопасности.
- Вместо классического FIFO стека реализован конвейер обработки сематических запросов. Более того, если не инициализированы все поля записи генерируемых (или генеративных) таблиц на этапе подготовки задачи, то данный запрос блокируется и не отправляется на выполнение аппаратному устройству;

- Введен контроль выполнения операций микропроцессором на уровне S8: сначала идет проверка всего набора переменных генеративных таблиц для каждого запускаемого процесса-треда;
- Встроена поддержка глобально-адресуемой памяти размером в несколько десятков Пбайт и разработан прозрачный контроль контекстно-зависимых переходов при выполнении команд посредством набора специальных полей <W, B, T >;
- Разработана и успешно апробирована встроенная защита данных, команд и тредовых устройств посредством реализации маркерного сканирования, что позволяет создать изолированную программно-аппаратную среду для выполнения задач и обнаруживать вредоносное ПО на всех восьми уровнях иерархии выполнения команд.

В качестве средств разработки использовались: программный пакет схемотехнической разработки VHDL, GNU ассемблер, отладчик Total View.

Исследования проводились на испытательном стенде, который объединяет 50 сервисных узлов и узлов подсистемы хранения данных, 2 сервера управления, 30 вычислительных узлов и 20 рабочих станций, и представляет собой виртуальную защищенную сеть.

Предложенные технологические решения не отменяют классические методы контроля и защиты программного обеспечения, а позволяют реализовать «комплексный подход»: сохраняя существующие архитектурные решения промышленного масштаба, авторский метод встраивания на уровнях S1 – S8, с одной стороны, не сказывается на работе модулей систем защиты на более высоких интерфейсных уровнях, с другой – создание программных агентов системы контроля на дополнительных уровнях позволяет повысить уровень защищенности программного обеспечения (ПО) и не влияет на общую производительность вычислительной системы в целом.

В отличие от традиционного конвейера обработки запросов, где передается только сигнатуры операций, в программно-техническом комплексе «Альфа-монитор» формируются генеративные таблицы на этапе приготовления задачи и хранятся в специальных тредовых регистрах *t_reg*.

Они представляют непроцедурную нефункциональную, а семантическую часть кодирования цепочки клиентских запросов. После чего передается стандартная процедурная составляющая – опкод инструкций (набор кода данных и кода команд, модификаторов и флагов условных переходов).

Модуль доверенной загрузки и защиты загружается с уровнем привилегий IPL и осуществляет монополярный контроль выделения вычислительных ресурсов сервера, просматривая данные из *t_reg*. Используется 4 кольца защиты процессора: Dom_id = 00, Dom_id= 01, Dom_id= 10, Dom_id = 11.

С учетом поддержки аппаратной виртуализации «склейка» параметров Dom_id и s (строки записываются в стек в обратном порядке) позволяет задать тот интерфейсный уровень, на котором обрабатывается транзакция по схеме «клиент-серверное взаимодействие». На основе декомпозиции выделяется 8-уровней функционирования компонентов гипервизора **modi**. Заданы 4 непривилегированных уровня – наборы 111, 110, 101, 100, 4 привилегированных уровня – 011, 010, 001, 000.

При использовании стационарных и бортовых СК с общей памятью не обойтись без синхронизации доступа параллельных процессов к общему ресурсу (памяти). Последние аппаратные реализации транзакционной памяти осуществляется с помощью префиксов, модифицирующих исполнение инструкций. Вводятся специальные команды XBEGIN, XEND, XABORT – новые расширения, которые схожи по мнемонике с командами SYSENTER, SYSEXIT, VXMON, служащими для поддержки аппаратной виртуализации. Изоляция участков памяти реализуется на основе нескольких колец защиты.

Эффективность разработанной системы защиты оценивалось на основе применения разных средств защиты и анализа числа успешных распознаваний и ошибок вредоносного ПО. Касательно оценки уровня отказоустойчивости следует отметить, что в управляющей операционной системы (ОС) реализованы механизмы корректной блокировки недопустимых операций на всех уровнях иерархии выполнения команд.

Теоретический вклад заключается в том, что соискателем впервые разработаны подходы и принципы самой методологии оценки безопасности в виде теоремы о защищенности СК, построена функция оценки безопасности, предложена оптимизация вычисления дескрипторов функции оценки безопасности за счет введения дополнительных полей контроля в виде набора восьми переменных во фреймах запросов и восьми интервальных ограничений. Независимо от архитектуры вычислительных узлов СК найдено такое число уровней иерархии, чтобы в n -мерной алгебраической системе существовал инвариант, при этом с изменением размерности параметра n не меняются регулярные отношения между элементами множества и атрибутами сущностей, не появляются новые свойства у объектов алгебраической структуры модели безопасности.

С точки зрения практического значения для такой отрасли науки, как информатика и вычислительная техника, решена технологическая проблема супервентности. Разработано с учетом специфики суперкомпьютеров принципиально *новое технологическое решение*, создающее изолированную среду исполнения программ в виде 8-уровневой “песочницы”. Технология следящих действий со стороны гипервизоров и управляющих ОС суперкомпьютеров, основанная на использовании проксирующих модулей и верификатора команд, которая позволяет по сравнению с существующими отечественными и зарубежными подходами значительно повысить уровень защищенности суперкомпьютеров от информационных атак и внешних специальных воздействий.

Потери производительности при использовании комплекса верификации команд не более 6-7 %.

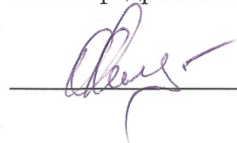
Практический результат – повышение уровня защищенности ОС и эффективности обнаружения вредоносного ПО.

Метод реактивной защиты и метод реконфигурации среды исполнения с реализацией принципиально новых механизмов работы с памятью, «прозрачному» управлению ресурсами задач и поддержки микропроцессорных вычислительных блоков с MTDF-архитектурой позволили обнаруживать вредоносное ПО на всех уровнях иерархии выполнения команд по сравнению с традиционными СЗИ.

Для обеспечения требований международного стандарта ISO 5725 к сходимости и воспроизводимости данных измерений опыты выполнялись сериями из 5-ти повторных опытов в течение 14-ти дней до стабилизации наблюдаемых данных. Испытания на экспериментальном стенде подтвердили, что показатели защищенности у прототипов с принципиально новыми архитектурными доработками на 86 % выше, чем у существующих технологических решений на рынке СЗИ.

Личный вклад автора неоценим. За время работы проявил себя высококвалифицированным специалистом, умело и грамотно решающим сложные вопросы разработки систем безопасности суперкомпьютерных вычислительных комплексов.

Зав. кафедрой «Комплексная защита информации», д.т.н., с.н.с.:



О.В. Казарин

Реализация теоретических и методических результатов диссертации в учебном процессе РГГУ

МИНОБРНАУКИ РОССИИ



Федеральное государственное
бюджетное образовательное учреждение
высшего образования

**«Российский государственный
гуманитарный университет»
(ФГБОУ ВО «РГГУ»)**

Миусская пл., д. 6, Москва, ГСП-3, 125993
тел. +7 495 250-61-18; факс +7 499 250-51-09
e-mail: rsuh@rsuh.ru; <http://www.rgu.ru>
ОКПО 02068746; ОГРН 1037700067118;
ИНН/КПП 7707033405/770701001

18.11.2019 № 1703-22/53

На № _____ от _____

Справка о внедрении теоретических
и методических результатов
диссертации

Настоящая справка выдана Молякову Андрею Сергеевичу, кандидату технических наук, доценту кафедры комплексной защиты информации ИИНТБ РГГУ в том, что теоретические и методические результаты его диссертационной работы «Модели и методы обеспечения информационной безопасности стационарных и бортовых суперкомпьютерных вычислительных систем», представленной на соискание ученой степени доктора технических наук, использованы в учебном процессе и научных исследованиях на кафедре комплексной защиты информации ФГБОУ ВО «РГГУ» по дисциплинам «Программно-аппаратные средства защиты информации. Основная часть», «Актуальные тенденции в области защиты информации», «Безопасность операционных систем». Автором разработаны и введены новые теоретические и научно-методические принципы в области обеспечения информационной безопасности суперкомпьютеров.

Разработан и внедрен лабораторный практикум для успешного овладения материалом и практических навыков у студентов, обучающихся по специальностям «Комплексная защита объектов информатизации» и «Организация и технология защиты информации», включающий в себя использование следующих моделей:

- модели угроз целостности среды исполнения процессов, в которой реализована новая парадигма написания правил политики безопасности и заданы новые сущности – «субъект», «объект» и «дескриптор оценки безопасности состояний»;
- модели безопасных операций, рассматривающей с позиций «целостности» среды исполнения процессов суперкомпьютерных систем, описывающей ее процессы в виде декомпозиции на 8-уровневую иерархическую структуру;
- модели функционирования вычислительного устройства суперкомпьютеров на основе мультитредовой архитектуры с управлением потока данных (MTDF), повышающие его уровень защищенности за счет избыточного параллелизма, многоуровневой защиты, проксирующих модулей и верификатора команд.

Первый проректор-проректор по научной работе

Директор ИИНТБ

И.о. зав. кафедрой комплексной защиты информации

О.В. Павленко

А.А. Роганов

Д.А. Митюшин



Реализация авторских изобретательских и рационализаторских предложений в рамках китайского проекта 863/ИТ по созданию защищенных суперкомпьютерных вычислительных комплексов серии СТ-2 :

1. Цель проекта – развитие стратегических ИТ и создание на их основе перспективных стационарных и бортовых суперкомпьютеров, предназначенных для решения важнейших научно-технических задач.
2. Запатентованный авторский оригинальный способ функционирования вычислительного устройства СК позволил внедрить новую компьютерную технологию сбора, хранения, обработки и защиты информации на основе мультитредовой архитектуры с управлением потока данных (MTDF).
3. Спроектированный автором макет вычислительного устройства суперкомпьютера в защищенном исполнении собран в виде опытного образца массово-мультитредового процессора и материнской платы с поддержкой MTDF-архитектуры (см. чертежи и пояснения ниже). В 2020 году опытные образцы вычислительных модулей запущены в серийное производство.
4. Разработанные автором архитектурно-технологические принципы, метод реактивной защиты и метод реконфигурации среды выполнения программ суперкомпьютеров проактивной защиты использованы китайскими специалистами и реализованы при разработке и сборке вычислительных блоков СТ-2 под управлением защищенного мультидоменного распределенного гипервизора.
5. Разработанные базовые гибридные микропроцессоры проекта СТ-2 позволяют программно эмулировать системы команд PowerPC, X86 и микропроцессора Monarch (этот микропроцессор включает решающее поле неоднородных элементов и универсальные процессорные ядра RISC-типа), что обеспечивает переносимость ранее созданных комплексов.
6. Подсистема обеспечения ИБ суперкомпьютеров серии Tian-He/СТ-2 реализована на базе мультидоменного гипервизора и коммутаторов Arch.
7. Созданный автором алгоритм вычисления дескрипторов оценки безопасности состояний в виде свертки на основе атрибутов протекающих процессов, реализован на уровне ядер микропроцессоров и среды межсетевого взаимодействия.

МИНОБРНАУКИ РОССИИ



Федеральное государственное
бюджетное образовательное учреждение
высшего образования

**«Российский государственный
гуманитарный университет»
(ФГБОУ ВО «РГГУ»)**

Миусская пл., д. 6, Москва, ГСП-3, 125993
тел. +7 495 250-61-18; факс +7 499 250-51-09
e-mail: rsuh@rsuh.ru; <http://www.rgu.ru>
ОКПО 02068746; ОГРН 1037700067118;
ИНН/КПП 7707033405/770701001

18.11.2019 № 1703-22/58

На № _____ от _____

Справка о внедрении
научно-технологических результатов
диссертации

Настоящая справка выдана Молякову Андрею Сергеевичу, кандидату технических наук, доценту кафедры комплексной защиты информации ИИНТБ РГГУ в том, что научно-технологические результаты его диссертационной работы «Модели и методы обеспечения информационной безопасности стационарных и бортовых суперкомпьютерных вычислительных систем», представленной на соискание ученой степени доктора технических наук, использованы в деятельности лаборатории компьютерной техники и средств защиты информации ИИНТБ РГГУ в ходе разработки новых методов и методики для обеспечения информационной безопасности суперкомпьютерных вычислительных систем в виде экспериментального стенда с реализацией:

- метода реактивной защиты, заключающегося в виртуализации среды исполнения процессов суперкомпьютеров, если вычисленный дескриптор состояния попадает в зону «риска», и основанный на контроле запросов на выделение ресурсов в соответствие с оценкой безопасности в виде модальных правил темпоральной логики CTL;
- метода реконфигурации среды исполнения суперкомпьютеров с учетом требований мобильности на основе применения траекторий вычислений дескрипторов безопасности состояний на структурах Крипке;
- методики тестирования уровня защищенности суперкомпьютеров, в которой алгоритм вычисления дескрипторов оценки безопасности состояний адаптирован к типу конфигурации суперкомпьютеров и потоку обрабатываемых данных.

Разработанный новаторский подход в области формализации информационных процессов с учетом специфики суперкомпьютеров и алгоритм идентификации угроз с использованием маркеров показали эффективность его применения в виде авторского программного продукта «Альфа-монитор». Реализация подобного решения позволила значительно снизить потери в производительности и повысить уровень защищенности ресурсов вычислительных систем.

Первый проректор-проректор по научной работе

О.В. Павленко

Директор ИИНТБ

А.А. Роганов

И.о. зав. кафедрой комплексной защиты информации

Д.А. Митюшин



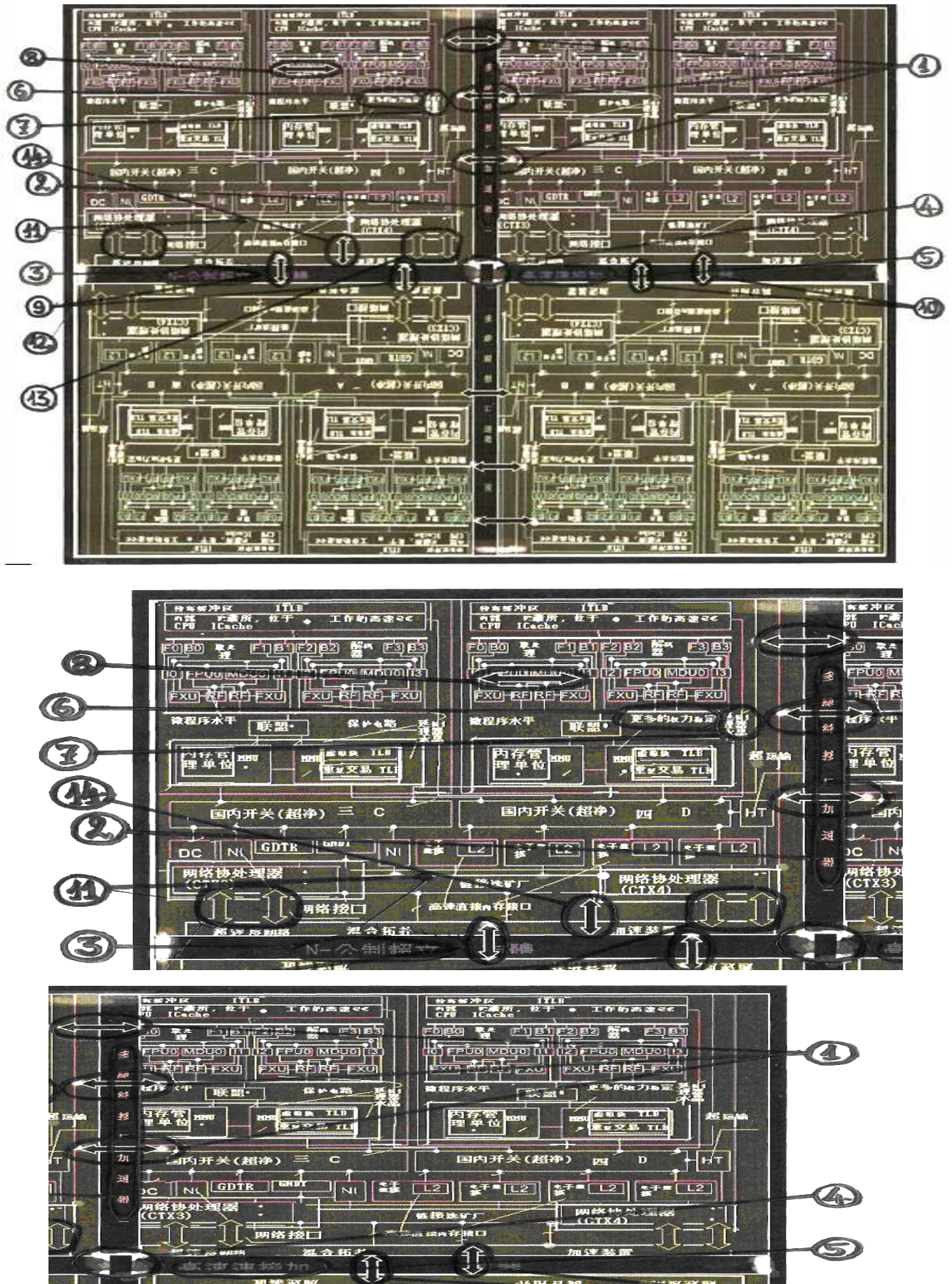


Рисунок 5 – Укрупненный вид одного ядра и подложки платы СТ-2

Пояснения к рисунку 5

- 1 – означает прямую связь кристаллов микропроцессора на трех уровнях, может быть применена для увеличения ядерности микропроцессора, а также для повышения отказоустойчивости (дублирование ресурсов, задействованных при выполнении программы и контроль сверкой выдаваемых ими результатов).
- 2 – высокоскоростные линки, выполненные с использованием нанотехнологий для обеспечения прямой связи кристаллов микропроцессора.
- 3 – высокоскоростные линки подложки, которые используются для реализации N-мерной сети модулей суперкомпьютера и микропроцессоров внутри модулей, также используется для реализации быстрых соединений микропроцессоров одной подложки (см. комментарии 9 и 10), N-мерная сеть, скорее всего, означает не N - тор, а многосвязную сеть типа N-куб или сеть Клоса.
- 4 – коммуникационный чип с бесштырьковыми соединениями.
- 5 – означает, что передача информации идет с сильным уплотнением и на высокой частоте.
- 6 – указывается, что возможна установка маски разрешения копирования данных, т.е. для некоторых сегментов данных можно указать, что их данные не надо кэшировать.
- 7 – цветные линии обозначают внутрикристалльные линии связей.
- 8 – означает, что возможны изменения по составу блоков и функциональных устройств.
- 9, 10 – указывается, что передачи между микропроцессорами одной подложки могут быть специальным образом ускорены и реализованы с учетом обеспечения отказоустойчивости (см. комментарий 1).
- 11 – канал прямого соединения адаптеров/сериализаторов/десериализаторов пакетов соответственно межмодульной сети (СТХ3) и внекристалльной памяти (СТХ4)
- 12 – линки выхода в коммуникационную сеть модулей суперкомпьютера, показаны два линка, один из которых соединен с блоком NI микропроцессора, это принципиально, поскольку для обслуживания некоторых запросов (например, для обращений к фрагментам некэшируемых сегментов данных), дополнительная трансляция в блоке MMU не требуется, можно сразу из адаптера сети СТХ3 передавать запрос в адаптер внекристалльной памяти СТХ4.
- 13 – линки выхода во внекристалльные DRAM-памяти, которые находятся на одной другой или нескольких других подложках 3D-сборки.
- 14 – возможны изменения коммуникационной подсистемы.

Коммерческая сборка представлена в виде серийно выпускаемой суперкомпьютерной вычислительной платформы Tianhe/CT-2.

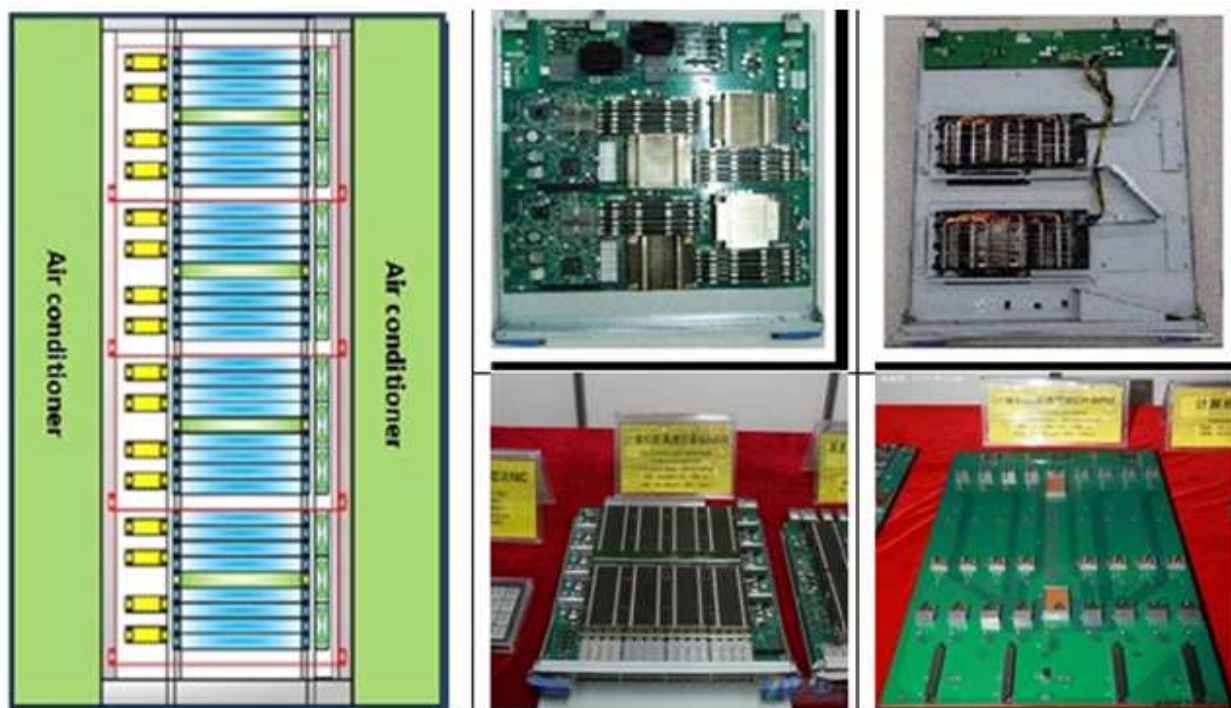


Рисунок 6 – Конструктивы вычислительной стойки

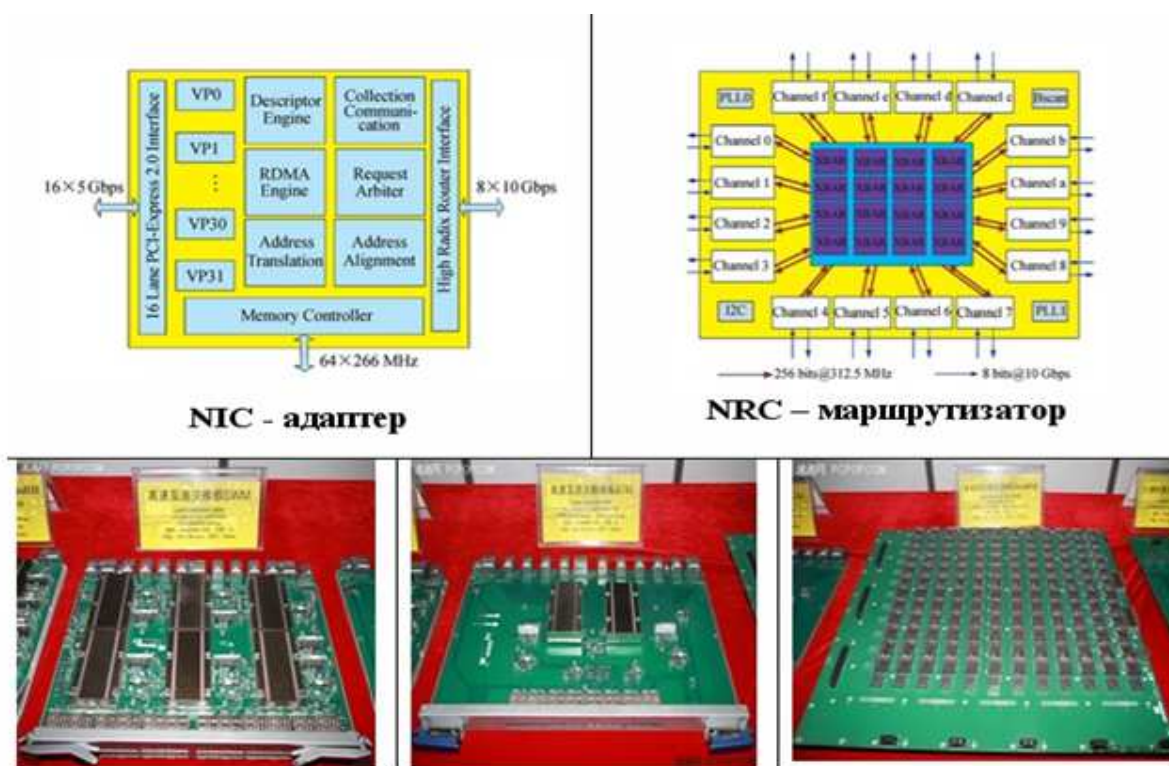


Рисунок 7 – Элементная база и конструктив коммуникационной стойки

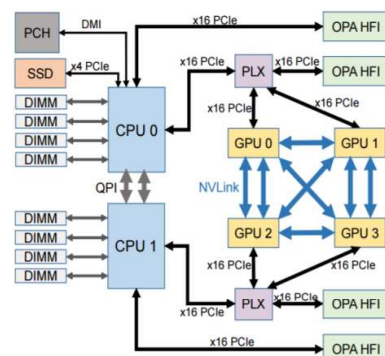
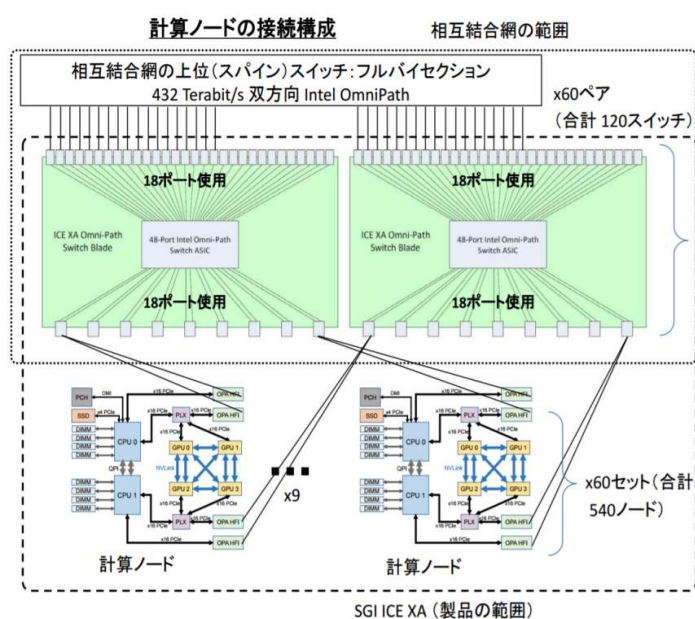
Реализация авторских изобретательских и рационализаторских предложений в рамках японского проекта JST CREST “Разработки новых НРС-технологий” 5-го Базового плана развития науки и технологий по созданию суперкомпьютерных вычислительных комплексов :

1. Спроектированный автором макет вычислительного устройства суперкомпьютера в защищенном исполнении собран в виде опытного образца сборочного конструктива (см. Чертежи и пояснения ниже). В 2021 году опытные образцы вычислительных модулей запущены в серийное производство.
2. Разработанная автором модель безопасных операций успешно внедрена при проектировании суперкомпьютера «Стрела времени» в составе многомодульного защищенного мультидоменного гипервизора.
3. Подсистема обеспечения ИБ суперкомпьютеров серии Tsubame 3.0 реализована на базе мультидоменного гипервизора и коммутаторов Tofu.
4. Разработанный автором метод реконфигурации среды выполнения программ суперкомпьютеров с учетом требований мобильности (встроенности производимых вычислений), адаптированных к контексту входных данных, позволил получить повышенную толерантность к задержкам выполнения операций с памятью и сетью, эффективно поддерживать работу с моделями программ в виде статических графов потоков данных.
5. В силу этого он успешно справляется с обработкой в реальном времени множества потоков данных и эффективно работать через единое адресное пространство с огромным объемом памяти в несколько десятков петабайт даже в режиме интенсивной нерегулярной работы с ней, обладает исключительной отказоустойчивостью и готовностью.

Название суперкомпьютера тенденциозно, поэтично и философски отражает пройденный Японией грандиозный путь в создании и освоении суперкомпьютерных технологий, необходимых для разработки такого суперкомпьютера, ассоциируется со стремлением обрести после его появления господствующее положение в военно-политическом отношении, по крайней мере, в азиатско-тихоокеанском регионе. Коммерческая сборка представлена в серийно выпускаемой суперкомпьютерной вычислительной платформе Tsubame 3.0.



TSUBAME3.0 計算ノード(Ⅱ) SGI ICE-XA ベース、ノードはSGI社と東工大GSICの共同設計によるTSUBAME3.0用新設計



超高性能な"Fat Node"構成(TSUBAMEの伝統)

- 4 SXM2(NVLink) NVIDIA Pascal P100 GPU
- 高ネットワークバンド幅 - Intel Omnipath ネットワーク 100Gbps x 4 = 400Gbps
- 高 I/O バンド幅- Intel 2 TeraByte NVMe
 - システム合計容量1ペタバイト以上、2Terabyte/秒の合算バンド幅
- 高密度・高温水冷ブレード - 1ラックあたり 36 ブレード = 144 GPU + 72 CPU、50-60KW
 - 一般データセンターの10-20倍

Рисунок 8 – Конструктивы вычислительной стойки и подсистемы коммуникаций

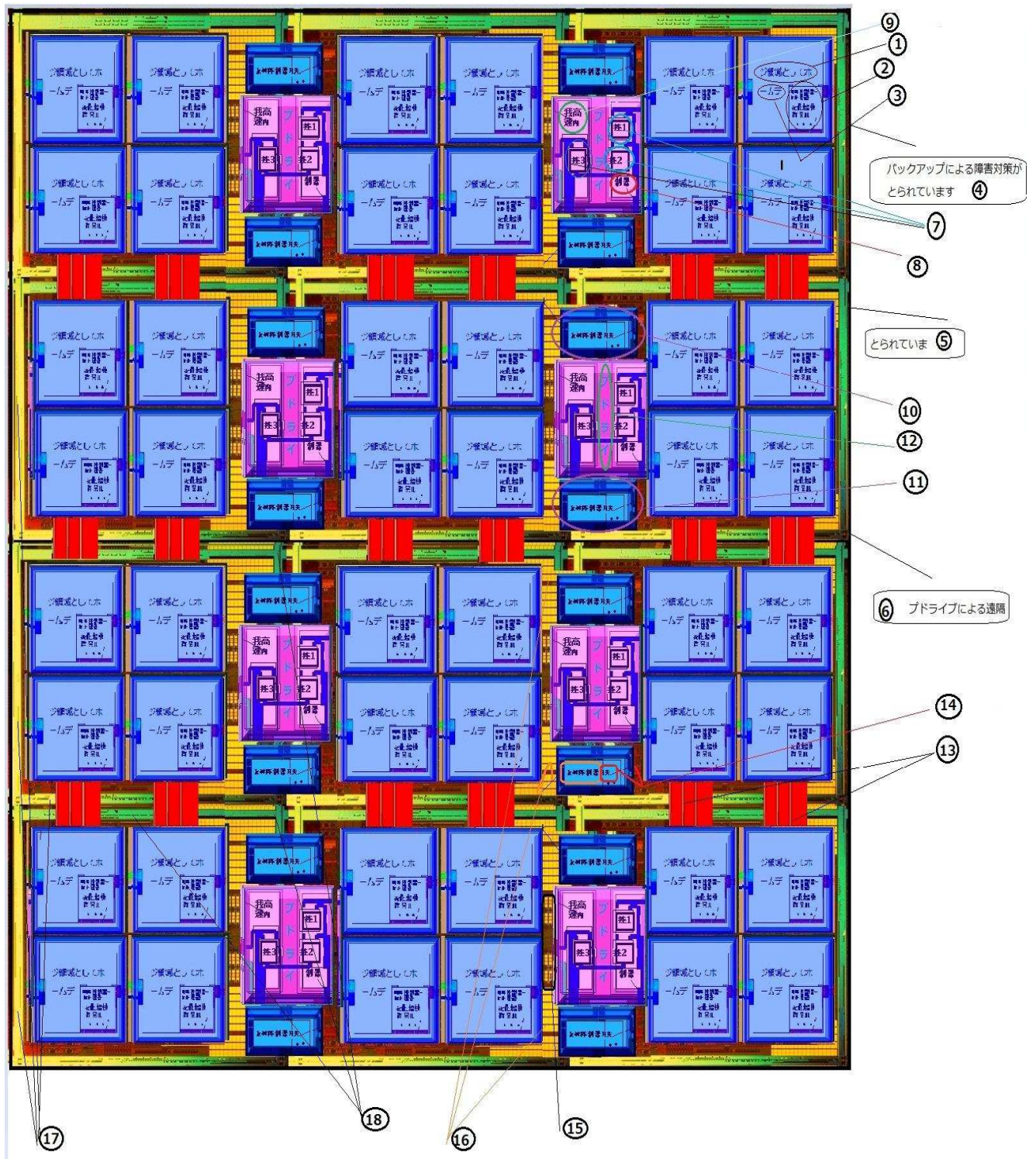


Рисунок 9 – Схема сборочного конструктива. Размер – приблизительно 15х18 см.
Сборочные конструктивы могут быть и больше по количеству узлов.

Комментарии к рисунку 9:

1 -Площадка для установки кристалла многоядерного микропроцессора (о типе ядер см. далее комментарий 2) Эти площадки находятся на “микроплате”, на рисунке показывается микроплата с четырьмя такими площадками.

На микроплате на каждой площадке устанавливается 3D-сборка. В сборку могут входить модули DRAM-памяти. В целом, микроплата - это аналог того, что сейчас называется “сокетом” в современных серверных платах. Каждая 3D-сборка снабжается индивидуальной системой охлаждения.

2 - Процессорные ядра микропроцессоров могут быть разными по типу поддерживаемых в ядре процессов выполнения команд одного или множества процессов-тредов – сверхлёгкие, легкие, средней тяжести, тяжелые. Например – сверхлегкие ядра графических процессоров (GPU), средние ядра RISC-микропроцессоров типа MIPS, тяжелые ядра суперскалярных микропроцессоров с архитектурой x86.

3 – В каждой 3D-сборке имеется главный микропроцессор и подчиненные ему, обеспечивается режим работы “master-slave”

4 – Могут использоваться не только 3D-сборка, но и трехмерные СБИС. В системе используются средства работы с очень большими данными (большими объемами данных из многоуровневой памяти системы), а также выделенные магистрали для работы с сетями передачи данных (датчиками, через которые поступают потоки данных). Возможна организация обработки данных с использованием синхронных и асинхронных тредов, а также с использованием векторов разной длины.

5 – Данная магистраль может передавать данные, если нет активных сообщений - парселов для управления и выполнения вычислений на

удаленных узлах с целью локализации вычислений на расположенных в этих узлах данных (см. также комментарий 18).

- 6 – В представленном на рисунке сборочном конструктиве внешние соединения выполнены в виде бесконтактных бесштырьковых оптических соединений (технология IBM Holley-чип с маломощными лазерами и матрицами микролинз, см. комментарий 16). Эти соединения используются Сетевыми процессорами (см. комментарий 10) и Процессорами ввода-вывода (см. комментарий 11).

Оперативная память в сборочном конструктиве реализуется множеством 3D-модулей DRAM-памяти, которые подключены к массово-мультитредовым процессорам памяти (см. комментарий 12) над ними в соответствующей 3D-сборке.

При передаче сообщений по сетям используются: большое количество форматов сообщений; методы агрегирования коротких сообщений в длинные и обратное преобразование; одновременная передача сообщений через оптический канал на разных длинах волн (WDM-технология, см. комментарий 15), активные сообщения типа RPC (удаленный вызов процедуры).

- 7 – Три “сквозных” буфера для временной развязки асинхронно работающих линий передач данных и сообщений

- 8 – Энергонезависимая флэш-память

- 9 – Процессор обработки сообщений и данных памяти “на лету”

- 10 – Сетевой процессор, через него реализуется подключение к внешней коммуникационной сети передачи сообщений между сборочными конструктивами

- 11 - Процессор ввода-вывода, через него реализуется подключение к внешней сети ввода-вывода

- 12 - Массово-мультитредовый процессор работы с памятью, через который реализуется подключение к 3D-модулю DRAM-памяти (он

находится на этом месте в сборочном конструктиве), а также реализуется подключение к Сетевому процессору и Процессору ввода-вывода.

Этот массово-мультитредовый процессор – главный элемент обеспечения толерантности каждого узла сборочного конструктива к задержкам выполнения операций с памятью, коммуникационной сетью и сетью ввода-вывода.

13 – Маршрутизаторы внутренних для сборочного конструктива сетей с топологией двумерной решетки для передачи данных и управляющих сообщений (парселов, зеленые шины). Если парселов немного, то их подсеть автоматически используется для передачи обычных сообщений.

14 – Электрическое соединение с подсистемой ввода-вывода

15 – WDM – трансивер, служит для реализации передачи сообщений через оптоволокно одновременно на разных длинах волн.

16 - Каждый желтый квадрат – матрица маломощных лазеров, подключенных через матрицу микролинз к пучку оптоволокон (технология IBM Holley-чип)

17 – Желтые магистрали предназначены для передачи данных в приведенном сборочном конструктиве

18 – Зеленые магистрали предназначены для передачи парселов (активных сообщений типа удаленного вызова процедур) в пр. Если таких парселов на задаче оказывается мало, то эти магистрали используются для передачи данных, что обозначено переходом зеленой раскраски магистрали в желтую.

Реализация в рамках российского проекта по созданию высокопроизводительных мобильных (бортовых) вычислительных средств морского, наземного и воздушного базирования на базе отечественных вычислителей «Комдив» и «Грифон»

УТВЕРЖДАЮ

Заместитель начальника ФГБУ
«3 ЦНИИ» Минобороны России по
научной работе

кандидат технических наук



А. Волков

«17» декабря 2022 г.

АКТ РЕАЛИЗАЦИИ

результатов научных исследований, полученных в диссертационной работе на соискание ученой степени доктора технических наук «Модели и методы обеспечения информационной безопасности стационарных и бортовых суперкомпьютерных вычислительных систем» Молякова Андрея Сергеевича, доцента кафедры комплексной защиты информации ФГБОУ ВО «Российский государственный гуманитарный университет»:

	Наименование результатов	Где реализованы
1.	Модель угроз целостности среды исполнения процессов суперкомпьютеров, в которой вместо классической связки «субъект, объект, предикат» реализована новая парадигма написания правил политики безопасности и заданы новые сущности – «субъект», «объект» и «дескриптор оценки безопасности состояний»	При разработке проекта ТТЗ на ОКР по созданию семейства унифицированных мобильных вычислительных средств на основе отечественных микропроцессоров для систем и комплексов вооружения, функционирующих в реальном масштабе времени наземного, морского и воздушного базирования

Врио начальника управления
кандидат технических наук, доцент
«19» декабря 2022 г.

В.Воронин

Начальник отдела
кандидат технических наук
«19» декабря 2022 г.

М.Дружинин