Федеральное государственное бюджетное учреждение
Национальный исследовательский центр «Курчатовский институт»

На правах рукописи

Климентов Алексей Анатольевич

МЕТОДЫ ОБРАБОТКИ СВЕРХБОЛЬШИХ ОБЪЕМОВ ДАННЫХ В РАСПРЕДЕЛЕННОЙ ГЕТЕРОГЕННОЙ КОМПЬЮТЕРНОЙ СРЕДЕ ДЛЯ ПРИЛОЖЕНИЙ В ЯДЕРНОЙ ФИЗИКЕ И ФИЗИКЕ ВЫСОКИХ ЭНЕРГИЙ

05.13.11 - Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

Диссертация на соискание ученой степени
доктора физико-математических наук

Научный Консультант : Кореньков Владимир Васильевич,
доктор технических наук,
директор Лаборатории информационных технологий ОИЯИ

Москва - 2017

ОГЛАВЛЕНИЕ

Введение

Исследования в области физики высоких энергий (ФВЭ) и ядерной физики (ЯФ) невозможны без использования значительных вычислительных мощностей и программного обеспечения для обработки, моделирования и анализа данных. Это определяется рядом факторов:

- большими объемами информации, получаемыми с установок на современных ускорителях;
- сложностью алгоритмов обработки данных;
- статистической природой анализа данных;
- необходимостью (пере)обрабатывать данные после уточнения условий работы детекторов и ускорителя и/или проведения калибровки каналов считывания;
- необходимостью моделирования условий работы современных установок и физических процессов одновременно с набором и обработкой «реальных» данных.

Введение в строй Большого адронного коллайдера (БАК, LHC) [1], создание и запуск установок такого масштаба, как ATLAS, CMS, ALICE [2-4], новые и будущие проекты класса мегасайенс (FAIR[5], XFEL[6], NICA[7]), характеризующиеся сверхбольшими объемами информации, потребовали новых подходов, методов и решений в области информационных технологий. Во многом это связано:

- со сложностью современных детекторов и количеством каналов считывания, например, размеры детектора ATLAS составляют 44 х 25м, при весе 7000 тонн, детектор имеет 150 миллионов датчиков для считывания первичной информации;
- со скоростью набора данных (до 1 Пбайт/с);

- с международным характером современных научных сообществ и требованием доступа к информации для тысяч ученых из десятков стран (в научные коллаборации на LHC входят более восьми тысяч ученых из десятков стран, сравнимое количество ученых будет работать в проектах FAIR и NICA);
- с высокими требованиями к обработке данных и получению физических результатов в относительно короткие сроки.

Научный прорыв 2012 года — открытие бозона Хиггса [8], стал триумфом научного мегапроекта Большого адронного коллайдера. В последующие годы эксперименты на LHC исследовали свойства новой частицы, одновременно были увеличены светимость и энергия коллайдера. Современные эксперименты работают с данными в эксабайтном диапазоне и являются заметными “поставщиками” так называемых Больших данных и методов работы с ними. Как и в случае со Всемирной паутиной (WWW), технологией, созданной в ЦЕРН для удовлетворения растущих потребностей со стороны ФВЭ к обмену информацией между учеными, и совместному доступу к ней, вызвавшей бурное развитие информационных технологий и систем связи в конце XX века, технологии Больших данных начинают влиять на исследования в других научных областях, включая нанотехнологии, астрофизику, биологию и медицину. Большие данные часто является связующим звеном, которое объединяет разработки в различных областях науки в единый мегапроект [9]. В речи, произнесенной всего за несколько недель до того, как он был потерян в море недалеко от Калифорнийского побережья в январе 2007, Джим Грэй, пионер программного обеспечения для баз данных и исследователь, работавший в Microsoft, изложил набросок аргументов, которые показывают, что "экзапоток" научной информации существенно преобразует практику науки [10]. Доктор Грэй назвал это изменение “четвертой парадигмой” [11,12].

Стратегия научно-технологического развития России [13] определяет цель и основные задачи, а также основные приоритеты научных исследований и технологических разработок. Российские информационноемкие программы

исследований, поддерживаемые Правительством РФ, такие, как физика высоких энергий и ядерная физика, астрофизика, науки о Земле, биоинформатика и материаловедение, будут производить эксабайты данных в ближайшем будущем. Проблемы, которые ставит развитие таких областей науки с большими объемами данных, многочисленны. Данные эксабайтного масштаба, как правило, распределены и должны быть доступны для больших международных сообществ. Для управления и обработки больших массивов данных необходимы многоуровневые интеллектуальные системы, системы управления потоками данных, контроля и мониторирования, а также системы хранения информации.

Вопросы разработки компьютерной модели, архитектуры распределенных и параллельных вычислительных систем для обработки данных, рассмотрение основополагающих принципов и моделей таких систем, анализ алгоритмов параллельных вычислений обсуждаются в классических работах начала XXI века Э.Таненбаума и М. ван Стеена [14], а также В.В. Воеводина и Вл.В. Воеводина [15]. Следует отметить, что во второй половине XX века классические работы Н.Н. Говоруна [16] о применении ЭВМ для обработки и анализа данных в области физики частиц, совпавшие по времени с запуском новых ускорителей в СССР (У10, У70), ЦЕРН (PS, SPS) и США (AGS, SLAC), оказали большое влияния на развитие методики обработки данных в ФВЭ и ЯФ, и во многом заложили основу будущих компьютерных моделей обработки данных.

Уже на этапе создания архитектуры и компьютерной модели для экспериментов на Большом адронном коллайдере (1998/2001 гг.) стало очевидным, что хранение и обработка данных не могут быть выполнены в одном центре, даже таком крупном как Европейский центр ядерных исследований (ЦЕРН). Следует отметить, что это понимание было вызвано техническими, финансовыми и социологическими причинами, в том числе и отсутствием на начало XXI века решений, предложенных десятилетием позже ведущими коммерческими ИТ компаниями.

LHC – уникальный ускоритель, в котором каждые 50 нс происходит столкновение протонов при энергии 13 ТэВ с рождением около 1600 заряженных частиц, каждая из них регистрируется и анализируется триггером высокого уровня. В результате работы триггера около 1000 событий ежесекундно отбираются для дальнейшей обработки и анализа. Статистика, набранная за время работы LHC в 2010-2017 гг, составляет более 60 Пбайт “сырых” (неприведенных) данных. Управляемый объем данных современного физического эксперимента близок к 300 Пбайт. В 2014 и в 2016 годах физиками международного сотрудничества ATLAS было обработано и проанализировано 1.4 Эбайта данных. Беспрецедентный объем информации, поступающий во время второй фазы работы LHC (2015-2019), и ожидаемое возрастание объема информации на следующих этапах работы коллайдера, как и требования к вычислительным комплексам на современных и будущих установках (FAIR, XFEL, NICA), потребовали разработки новой компьютерной модели, методики и методов управления загрузкой, созданию новых систем для обработки данных. Необходимым условием для своевременной обработки данных и получения физического результата в короткие сроки (в течение года) стал переход от использования гомогенной вычислительной среды (грид) к гетерогенной вычислительной инфраструктуре с использованием суперкомпьютеров (СК), академических и коммерческих центров облачных вычислений, “волонтерских” компьютеров и отдельных вычислительных кластеров.

Еще на раннем этапе развития компьютерной модели LHC (2000-е годы) было принято решение объединить существующие и вновь создаваемые вычислительные центры (более 200) в распределенный центр обработки данных, и сделать это таким образом, чтобы физики университетов и научных организаций участвующих стран имели равные возможности для анализа информации. В результате работы физиков, ученых и инженеров в области ИТ была создана система известная сегодня как WLCG (Worldwide LHC Computing Grid) [17]. На сегодня WLCG - самая большая академическая распределенная вычислительная сеть в мире, состоящая из около 300

вычислительных центров в 70 странах мира. Более 8000 ученых использовали эти центры для анализа данных коллайдера в поисках новых физических явлений (на рисунке 1 показана карта вычислительных центров и проектов, входящих в консорциум WLCG).

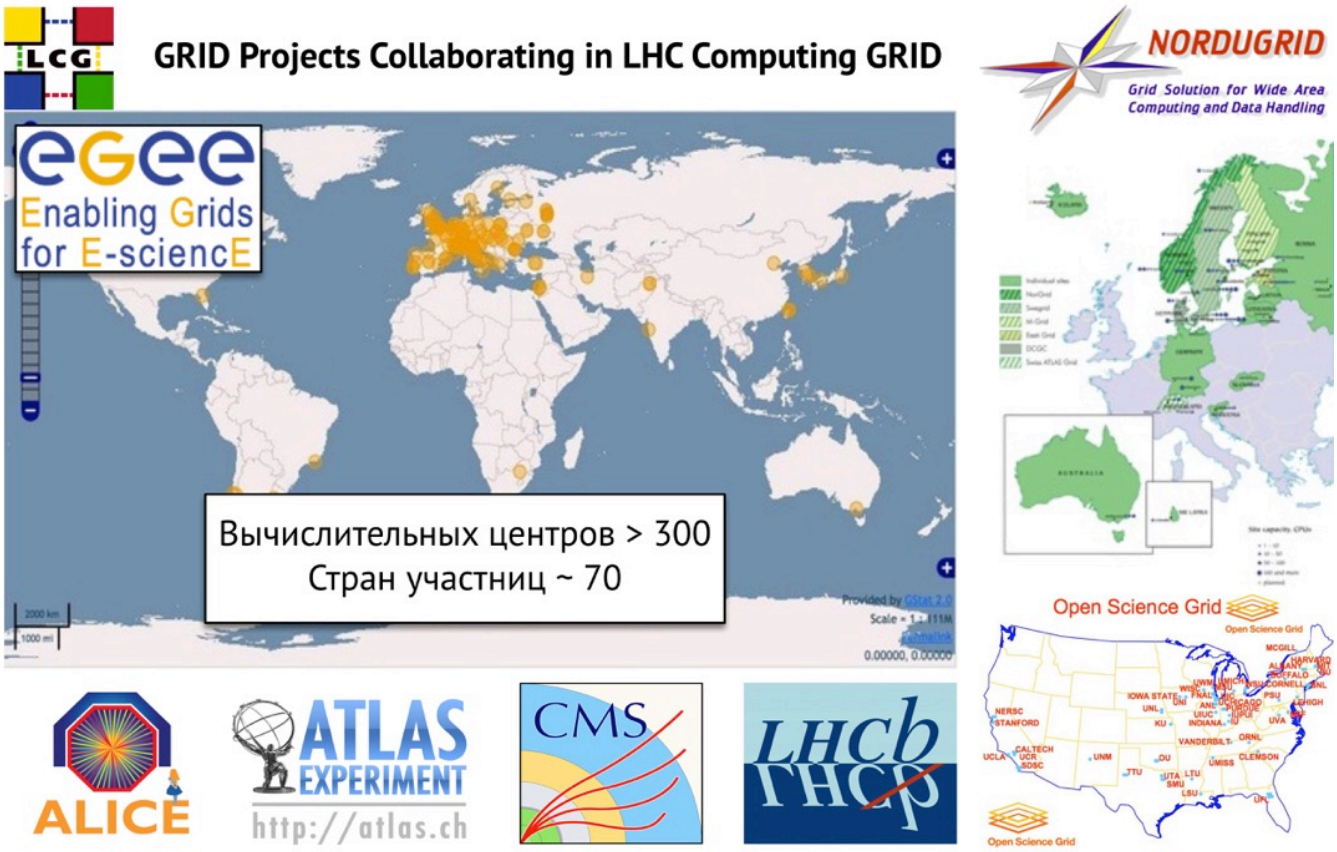


Рисунок 1 - Вычислительные центры и проекты, входящие в консорциум WLCG

Грид технологии были предложены в конце прошлого века Я. Фостером и К. Кессельманом. Основная концепция грид изложена в книге «The Grid: a Blueprint to the New Computing Infrastructure» [18]. Именно задачи ФВЭ и ЯФ привели к широкому использованию грид-технологий и потребовали существенных изменений и развития информационно-вычислительных комплексов (ИВК) в составе физических центров (в работе В.В. Коренькова [19] подробно рассмотрена эволюция ИВК ОИЯИ в составе грид-инфраструктуры и приведено обоснование этого развития).

В WLCG ежедневно выполняется до трех миллионов физических задач, общее дисковое пространство превышает 400 Пбайт, результаты обработки данных архивируются, распределяются между центрами обработки и анализа данных и поступают непосредственно на «рабочее место» физика. Подобную систему можно сравнить с огромным вычислительным комплексом узлы которого соединены высокоскоростным интернетом. Объемы передачи данных между центрами составляют до 10 Гбайт/с (среднее значение в течение дня). Создание системы заняло около 10 лет и потребовало вложений не только в инфраструктуру вычислительных центров во многих странах мира, но и развития сетевых средств. Для обмена данными между центрами WLCG были созданы две компьютерные сети, ориентированные на задачи LHC: LHCOPN (LHC Optical Private Network) [20] и LHCONE (LHC Open Network Environment) [21]. Создание WLCG стало возможно в результате совместной работы тысяч ученых и специалистов, и больших финансовых вложений.

Д-р. Фабиола Джианотти (руководитель эксперимента ATLAS в 2008/2013 гг., директор ЦЕРН с 2014 г) на семинаре, посвященном открытию новой частицы, сказала: «Мы наблюдаем новую частицу с массой около 126 ГэВ. Мы не смогли бы провести обработку и анализ данных так быстро, если бы не использовали грид. Центры во всех странах участницах эксперимента были задействованы в обработке данных LHC, практически это был стресс-тест для вычислительных мощностей, и грид показал себя высокоэффективной и надежной системой».

Роль распределенных компьютерных инфраструктур при обработке данных на первом этапе работы LHC подробно рассмотрены в работах автора, в том числе в соавторстве с В.В. Кореньковым и А.В. Ваняшиным [22,23], опубликованных в 2012-2014 годах. Тогда же автором были сформулированы основополагающие принципы развития компьютерной модели современных экспериментов в области физики частиц, новые требования к федерированию географически распределенных вычислительных ресурсов, требования к глобальным системам для распределенной

обработки данных и методам управления загрузкой в гетерогенной компьютерной среде [24].

Можем ли мы сказать, что LHC и WLCG выполнили поставленную задачу ? Если говорить об открытии новой частицы, то да. Ни ускоритель Теватрон (в лаборатории имени Э. Ферми, США), ни Большой электрон-позитронный коллайдер ЛЭП (LEP) в ЦЕРН за десятилетия работы не смогли зарегистрировать предсказанную в 1964 году частицу. Однако более важно получить ответ на следующие вопросы. Достаточно ли классическое решение грид, реализованное в рамках проекта WLCG, для решения задач следующих этапов работы коллайдера ? Как должна развиваться компьютерная модель для этапа superLHC (2022/2028 годы), а также для новых комплексов, таких как FAIR, XFEL, NICA ? Ответить на эти вопросы невозможно без понимания логики создания проекта WLCG и тех условий, в которых была разработана и реализована первая компьютерная модель распределенных вычислений для LHC. Необходимо проанализировать ограничения компьютерной модели и понять, насколько они носят фундаментальный характер, почему потребовалось создание новой компьютерной модели и распределенной системы обработки данных для второго и последующих этапов работы LHC. Применима ли новая компьютерная модель для экспериментов на установках класса мегасайенс в «эпоху Больших данных».

Работы по созданию концепции и архитектуры систем для распределенной обработки данных экспериментов в области ФВЭ и ЯФ, а также астрофизики была начата в конце XX века. Создание программного пакета Globus Toolkit [25] стало на десятилетия основным набором инструментов для построения грид-инфраструктуры и важнейшим этапом в развитии концепции грид. Тогда же были разработаны и реализованы первые сервисы для обнаружения ошибок и защиты информации, сервисы управления данными и ресурсами, сформулированы требования по взаимодействию сервисов внутри грид-систем. Следует отметить пионерские работы по развитию и созданию грид в России, в первую очередь в ЛИТ ОИЯИ (В.В.

Кореньков), НИИЯФ МГУ (В.А. Ильин) [26-28], а также разработки ИПМ им. М.В. Келдыша [29], кроме того многие идеи по концепции вычислительных сред, определившие нынешние подходы, были предложены в работах Института системного анализа РАН (А.П. Афанасьев) [30,31], а в работах НИВЦ МГУ адресованы вопросы эффективности работы суперкомпьютерных центров и проблемы их интеграции (Вл.В. Воеводин) [32,33]. Многие из предложенных идей, повлияли на развитие архитектур вычислительных систем и систем обработки и управления данными, а также на развитие компьютерной модели современных физических экспериментов.

Важным этапом развития систем для обработки данных явилось обоснование принципов построения и архитектуры системы, разработка методов планирования выполнения заданий. Это позволило создать принципиально новое программное обеспечение, необходимое для управления данными и заданиями в распределенной среде, разработать методы оценки эффективности функционирования систем управления загрузкой, методы оценки эффективности работы ВЦ (в рамках грид инфраструктуры) и методы распределения задач обработки и данных с целью оптимального использования вычислительного ресурса [34,35].

Компьютерная модель обработки данных физического эксперимента прошла в своем развитии много этапов, от модели централизованной обработки данных, когда все вычислительные ресурсы были расположены в одном месте (как правило там же, где находилась экспериментальная установка), к разделению обработки и анализа. которые по-прежнему велись централизованно, и моделирования данных, проводившегося в удаленных центрах. В эпоху LHC была предложена и реализована иерархическая компьютерная модель MONARC [36]. Следующим этапом стала модель равноправных центров внутри однородной грид инфраструктуры – “смешанная компьютерная модель» [37,38]. В настоящее время компьютерная модель, предложенная и реализованная автором [39], предполагает равноправное использование центров грид и интегрированных с грид ресурсов облачных

вычислений и суперкомпьютерных центров в рамках единой гетерогенной среды. Дальнейшее развитие компьютерной модели для этапа superLHC и комплексов FAIR, XFEL, NICA потребовало разработки концепции и архитектуры единой федеративной киберинфраструктуры в гетерогенной вычислительной [40].

Для обработки и управления большими массивами данных необходимы многоуровневые интеллектуальные системы и системы управления потоками заданий. Создание таких систем имеет свою эволюцию, сравнимую по количеству этапов с развитием компьютерной модели физических экспериментов. От набора программ, написанных на скриптовых языках и имитирующих работу планировщика в рамках одного компьютера, до систем пакетной обработки, таких как LSF[41] или PBS[42], с последующей разработкой пакетов программ управления загрузкой промежуточного уровня грид (HTCondor [43]), и на последнем этапе развития - разработка и создание высокоинтеллектуальных систем управления загрузкой (AliEN, Dirac, PanDA [44-46]). Эти системы способны управлять загрузкой и позволяют обрабатывать данные одновременно в сотнях вычислительных центров. Практическое использование систем управления загрузкой показало их ограничения по параметрам масштабируемости, стабильности, возможности использования компьютерных ресурсов вне грид. Выявились трудности при интегрировании информации глобальных вычислительных сетей с информацией об имеющемся вычислительном ресурсе, быстроте “захвата” этого вычислительного ресурса (что стало особенно заметно при переходе от модели MONARC к смешанной компьютерной модели, а также при использовании СК и коммерческих ресурсов облачных вычислений). Другой существенной проблемой стала реализация способа разделения вычислительного ресурса между различными потоками заданий : обработки данных, моделирования, анализа, а также предоставления вычислительного ресурса для задач эксперимента (“виртуальной организации”), отдельных научных групп и ученых, в рамках установленных квот использования вычислительного ресурса.

Таким образом, запуск Большого адронного коллайдера и создание новых ускорительных комплексов класса мегасайенс, характеризующихся сверхбольшими объемами информации и многотысячными коллективами ученых, обусловили новые требования к информационным технологиям и программному обеспечению. В эти же годы произошло качественное развитие информационных технологий, появление коммерческих вычислительных мощностей, превышающих возможности крупнейших ВЦ в области ФВЭ и ЯФ, развитие и резкое повышение пропускной способности глобальных вычислительных сетей. Требования по обработке данных на LHC и развитие ИТ привели к необходимости решения фундаментальной проблемы - разработки систем нового поколения для глобально распределенной обработки данных, разработки новой компьютерной модели физического эксперимента, позволяющей объединять различные вычислительные ресурсы и включать новые ресурсы (например, интегрировать ресурсы грид и суперкомпьютеры в единую вычислительную среду) .

Цель и задачи работы. Разработка и развитие методов, архитектур, компьютерных моделей и программных систем, реализация соответствующих программных и инструментальных средств для приложений физики высоких энергий и ядерной физики при обработке сверхбольших объемов данных.

Для достижения поставленной цели в диссертационной работе необходимо решить следующие основные задачи:

- Разработать компьютерную модель для экспериментов в области физики высоких энергий и ядерной физики, позволяющую объединять высокопропускные вычислительные мощности (грид), высокоскоростные вычислительные мощности (суперкомпьютеры), ресурсы облачных вычислений и университетские кластеры в единую вычислительную среду.

- Разработать принципы построения и архитектуру системы для глобальной обработки данных эксабайтного масштаба для тысяч пользователей в гетерогенной вычислительной среде.
- Разработать методы управления потоками заданий в гетерогенной вычислительной среде, позволяющие учитывать неоднородность потоков заданий и запросов пользователей, с целью оптимального использования вычислительных ресурсов, доступных в современном физическом эксперименте.
- На основе разработанных принципов и архитектуры создать масштабируемую (обработка данных эксабайтного диапазона в O(100) центрах O(1000) пользователями O(10^6) научных заданий/день) систему для обработки данных современного физического эксперимента.
- Разработать систему мониторирования и оценки эффективности работы глобальной системы для обработки данных в распределенной гетерогенной компьютерной среде.

Научная новизна работы

- Разработана компьютерная модель современного физического эксперимента для управления, обработки и анализа данных эксабайтного диапазона в гетерогенной вычислительной среде.
- Реализация разработанной модели для приложений в области физики частиц впервые позволила использовать различные архитектуры: грид, суперкомпьютеры и ресурсы облачных вычислений для обработки данных физического эксперимента через единую систему управления потоками заданий, сделав это “прозрачно” для пользователя.
- Разработаны принципы построения, методы, архитектура и программная инфраструктура системы для глобальной распределенной обработки данных. На этой основе создана система управления потоками заданий, не

имеющая мирового аналога по производительности и масштабируемости (более 2М задач, выполняемых ежедневно в 250 вычислительных центрах по всему миру).

- Решена проблема разделения вычислительного ресурса между различными потоками научных заданий (обработка данных, Монте-Карло моделирование, физический анализ данных, приложения для триггера высшего уровня) и группами пользователей (эксперимент, научная группа, университетская группа, ученый).
- Разработаны новые методы управления научными приложения ФВЭ и ЯФ для суперкомпьютеров, с использованием информации о временно свободных ресурсах, позволяющие повысить эффективность использования суперкомпьютеров, в частности, для LCF Titan, СК Anselm, СК НИЦ КИ.

Защищаемые положения

- Новая компьютерная модель современного физического эксперимента позволяет использовать гетерогенные вычислительные мощности, включая грид, облачные ресурсы и суперкомпьютеры, в рамках единой вычислительной среды.
- Новые принципы построения и архитектура глобальной системы для обработки данных в гетерогенной вычислительной среде, позволяют эффективно использовать вычислительные ресурсы и снимают противоречие по доступу к ресурсу между физическим экспериментом, группами пользователей и отдельными учеными.
- Разработанный комплекс методик, методов и система для управления потоками заданий, созданная на их основе, повышают эффективность обработки данных физических экспериментов и обеспечивает обработку данных в эксабайтном диапазоне, в масштабе более 2М задач в день, в 200 вычислительных центрах, для 1000 пользователей.

- Новые методы предсказания популярности (востребованности) классов данных и наборов данных, а также модель динамического управления данными в распределенной среде для сверхбольших объемов данных, повышают эффективность использования распределенного вычислительного ресурса.
- Подсистема мониторирования и оценки эффективности работы глобальной системы для обработки данных обеспечивает высокий уровень автоматизации при анализе работы системы и сбоев в работе распределенной вычислительной инфраструктуры, и ее аппаратно-программных компонент.

Практическая значимость. Основные результаты данной работы являются пионерскими и используются в действующих экспериментах в области ФВЭ и ЯФ и в других областях науки. В том числе, результаты работ, положенных в основу диссертации, используются в двух крупнейших экспериментах в области ФВЭ и ЯФ - ATLAS и ALICE на LHC, эксперименте COMPASS на SPS, а также для приложений биоинформатики на суперкомпьютерах НИЦ КИ :

- вычислительные модели экспериментов ATLAS и AMS опираются на результаты работ, положенных в основу диссертации;
- разработанная и созданная система управления потоками заданий в гетерогенной компьютерной среде используется в экспериментах на ускорителях LHC и SPS и принята в качестве базовой для будущего коллайдера NICA;
- разработанная система для обработки данных была также применена для исследований ДНК мамонта на суперкомпьютере НИЦ КИ и в европейском проекте BlueBrain.

Разработанная система управления загрузкой не имеет мировых аналогов по масштабируемости и отказоустойчивости. До 2М заданий выполняются ежедневно, в

2016 году физиками эксперимента ATLAS было обработано 1.4 Эбайта данных. Таким образом система уже сейчас работает в эксабайтном диапазоне.

Реализация результатов работы. Результаты диссертации были получены под руководством и личном участии соискателя в следующих международных проектах: WLCG - проект грид для LHC, megaPanDA - проект по созданию нового поколения системы управления заданиями в гетерогенной компьютерной среде, проект ATLAS на LHC, проекты AMS-01 и AMS-02 на Международной космической станции (МКС), проект metaMiner - по созданию системы поиска аномалий и предсказания поведения комплексных распределенных вычислительных систем, проект Federated Storage - по созданию прототипа распределенной компьютерной среды.

Автор диссертации внес определяющий вклад при выполнении ряда национальных российских и международных проектов, в том числе L3, AMS, ATLAS, megaPanDA, в которых автор являлся одним из руководителей (или руководителем) компьютерной и программной частями проекта и одновременно основным архитектором создаваемых систем и программного обеспечения.

Работы в 2013-2016 годах были поддержаны грантом Министерства образования и науки по привлечению ведущих ученых, тремя грантами РФФИ и грантом РНФ. В настоящее время автор является руководителем мегагранта и руководителем двух международных проектов совместно с ЦЕРН и DESY - “Создание федеративного распределенного дискового пространства”, “Использование алгоритмов машинного обучения для приложений ФВЭ”.

Базовая вычислительная модель реализуется в проекте ATLAS на LHC, и рассматривается как основная для ускорительного комплекса NICA (ОИЯИ).

Созданы системы управления загрузкой для распределенной обработки данных в НИЦ КИ (для приложений биоинформатики), ОИЯИ (для эксперимента COMPASS в ЦЕРН), ЦЕРН (эксперименты ATLAS), ORNL (для высокоинтенсивных научных

приложений), EPFL (проект BlueBrain, Лозанна, Швейцария), ASGC (эксперимент AMS-02, Тайпей, Тайвань).

Апробация диссертации. Результаты работы являются итогом более чем 20-летней научной и организационной деятельности соискателя. Основные результаты диссертации докладывались и обсуждались на научных семинарах НИЦ “Курчатовский институт”, ОИЯИ, ЦЕРН, БНЛ, НИЯУ МИФИ, ТПУ, докладывались на конференциях, рабочих совещаниях и научных семинарах экспериментов COMPASS, AMS, L3. Результаты работ регулярно обсуждались международными научными коллаборациями ATLAS и ALICE, в том числе на пленарных заседаниях во время конференций и на симпозиумах консорциума WLCG. Результаты, представленные в диссертации, докладывались на международных и российских конференциях, в том числе:

- международных конференциях "Computing in High Energy Physics» (CHEP): 2002 (Пекин, КНР), 2004 (Интерлакен, Швейцария), 2007 (Ванкувер, Канада), 2009 (Прага, Чехия), 2011 (Тайпей, Тайвань), 2012 (Нью Йорк, США), 2015 (Окинава, Япония), 2016 (Сан-Франциско, США);
- международных конференциях «Advanced computing and analysis techniques in physics research» (ACAT): 1994 (Комо, Италия), 2002 (Москва, Россия), 2008 (Эричи, Италия), 2014 (Прага, Чехия - пленарный доклад), 2016 (Вальпараисо, Чили), 2017 (Сиэттл, США);
- международных конференциях по физике высоких энергий ICHEP ; 2012 (Мельбурн, Австралия), 2014 (Валенсия, Испания), 2016 (Чикаго, США);
- международной конференции Real-Time Computer Applications in Nuclear and Plasma Physics, 1994 (Дубна, Россия);
- международной конференции Calorimetry in High Energy Physics 1999 (Лиссабон, Португалия);

- международном симпозиуме IEEE Nuclear Science Symposium and medical imaging conference (IEEE NCC/MIC), 2003 (Портланд, США);
- международном совещании "Physics and Computing at ATLAS", Дубна, 2008;
- международном симпозиуме “Grid and Clouds Computing”: 2010 (Тайпей, Тайвань);
- международных симпозиумах “Nuclear Electronics and Computing” (NEC): 2011 (Варна, Болгария - приглашенный доклад), 2015 (Будва, Черногория), 2017 (Будва, Черногория);
- международных конференциях “Распределенные вычисления и грид технологии в науке и образовании” (GRID): 2012 (Дубна, Россия), 2016 (Дубна, Россия - приглашенный доклад);
- международных конференциях “Наука Будущего” (Science of the Future): 2014 (Санкт-Петербург, Россия), 2016 (Казань, Россия - приглашенный доклад), 2017 (Нижний Новгород, Россия);
- международных конференциях Smoky Mountains Computational Science and Engineering, июль 2014, (Ноксвилл, США - приглашенный доклад), сентябрь 2017 (Гатлинбург, США - приглашенный доклад);
- международной конференции Data Analytics and Management in Data Intensive Domain, 2015 (Обнинск, Россия);
- VI Московском суперкомпьютерном форуме, 2015 (Москва, Россия);
- международной конференции “Supercomputing 2016”, 2016 (Солт-Лейк Сити, США);
- Конференции консорциума World LHC Computing Grid, 2016 (Лиссабон Португалия);
- международной конференции Instrumentation for Colliding Beam Physics. 2017 (Новосибирск, Россия - приглашенный доклад).

Соискатель являлся членом программных и международных комитетов конференций CHEP, NEC, GRID, DAMDID, а также (со)руководителем международных симпозиумов по обработке данных LHC (Дубна 2008, 2014), суперкомпьютерам (Нью-Йорк, США, 2013), методам машинного обучения для научных приложений в физике высоких энергий (Москва 2016) и Программное обеспечение для будущих экспериментов (Петергоф, 2017), где также были представлены результаты работ, положенных в основу данной диссертации.

Публикации и личный вклад автора. Изложенные в диссертации результаты получены соискателем в результате его многолетней научной и организационной деятельности по разработке и созданию программного обеспечения, систем для обработки и анализа данных и компьютерных моделей для экспериментов ФВЭ, ЯФ и астрофизики (L3, AMS, ATLAS), в частности, системы управления загрузкой в гетерогенной компьютерной среде для этих экспериментов, а также выполненных им работ для экспериментов класса мегасайенс в в Лаборатории “Технологии Больших данных НИЦ “Курчатовский институт”, созданной и руководимой соискателем.

Все исследовательские работы и разработки по теме диссертации - от постановки задачи и выбора методики до получения результатов - выполнены соискателем и/или под его непосредственным руководством, вклад соискателя в этих работах является определяющим. Все выносимые на защиту результаты получены соискателем лично.

По теме диссертации автором опубликовано свыше 150 печатных работ, в том числе по основным результатам - 68 работ (из них 47 работ в изданиях из перечня ведущих рецензируемых научных изданий). Результаты работы также опубликованы в отчетах по руководимым автором инфраструктурным и научным проектам в рамках мегагранта Правительства РФ и проектам, поддержанных РНФ и РФФИ.

Структура и объем диссертации. Диссертация состоит из введения, 4 глав, заключения, списка литературы из 115 наименований; полный объем работы составляет 238 страниц.

Глава 1. Развитие вычислительной модели экспериментов в области физики элементарных частиц и астрофизики

В данной главе приведен краткий обзор компьютерных моделей наиболее значимых экспериментов в области физики элементарных частиц и ядерной физики на ускорителях и коллайдерах в последние десятилетия, а также астрофизического эксперимента AMS/AMS-02 на международной космической станции (МКС). В эксперименте AMS-02 автором была предложена и реализована одна из первых компьютерных моделей для распределенной обработки данных. Подробно рассмотрена иерархическая компьютерная модель MONARC, предложенная для экспериментов на LHC, и метод ее реализация. Проанализированы ограничения модели, обоснована необходимость эволюции компьютерной модели после первого этапа работы LHC, рассмотрены вопросы классификации данных физического эксперимента, вопросы популярности данных, рассмотрен вопрос о роли глобальных вычислительных сетей при создании распределенной вычислительной инфраструктуры. Приводится описание и способ реализации «смешанной компьютерной модели» в рамках грид инфраструктуры, что позволило создать предпосылки для дальнейшего развития компьютерной модели и к переходу к гетерогенной компьютерной модели на втором и последующих этапах работы LHC.

1.1 Этапы развития компьютинга в области физики высоких энергий, ядерной физики и астрофизики

1.1.1 Компьютерные модели обработки данных в физике частиц до запуска Большого адронного коллайдера

В таблице 1 представлены сравнительные характеристики экспериментов в области физики частиц в последние 60 лет. Первым прорывом явилось использование компьютеров для online и offline обработки данных, и магнитных

лент для архивирования информации, с последующей обработкой данных на машинах серий ЕС и IBM. Развитие вычислительной техники в конце 80х годов прошлого века, появление поколения машин серий СМ, PDP, VAX, а также сетевого протокола DECNET, наряду с созданием сегментов скоростной локальной сети ETHERNET, функционирующей с пропускной способностью 10 Мбит/сек., впервые позволило перейти от изолированных ЭВМ,

Таблица 1 - Характеристики экспериментов в области физики частиц в последние 60 лет

Годы	Число Сотрудников эксперимента	Объем данных, технология хранения и обработки данных, и информации
Конец 1950	2-3	Кбиты, записи в рабочих журналах
1960 (У7)	10-15	Кбайты, перфокарты, бумажные носители
1970 (У10,У70, PS, AGS)	~35	Мбайты, магнитные ленты Онлайн обработка: PDP 8, оффлайн обработка: ЕС, IBM 360
1980 (SPS, У70)	~100	Гбайты, магнитные ленты и диски Онлайн обработка: Caviar, PDP 70, VAX, СМ4, Оффлайн обработка: ЕС, IBM 370, БЭСМ 6, VAX 8800
1990 (LEP, SLAC, Теватрон, RHIC)	700-800	Тбайты, магнитные ленты, диски Онлайн обработка: VAX, спец.процессоры; Оффлайн обработка : ЕС, IBM 370, VAX 8800, Appollo, SGI, Sun
2010 (LHC)	~3000	Пбайты, магнитные ленты, диски Онлайн обработка: кластеры, графические процессоры Оффлайн обработка : грид

к кластерам из нескольких машин и рабочих станций (как правило аппаратно-совместимых с основной ЭВМ), а также “связать” обработку в реальном масштабе времени (online) с постобработкой (offline), и передачей данных между центрами online и offline и обработки. Такие работы практически одновременно были выполнены в ЦЕРН (эксперименты UA1/UA2) [47], ОИЯИ и ИТЭФ (адронный калориметр установки L3) [48]. Работы в ИТЭФ по созданию распределенной системы сбора и обработки данных прототипов адронного калориметра были выполнены под руководством автора диссертации и явились важным этапом в развитии систем обработки данных на коллайдере LEP [49]. Эта работа послужила основой следующего этапа, в 1993 году под руководством автора диссертации была впервые реализована распределенная обработка данных для эксперимента L3. Локальной сетью ETHERNET (с пропускной способностью 100 Мбайт/сек) были связаны компьютеры, работающие в точке пересечения пучков N2 LEP (ВЦ для online обработки), и центром обработки данных эксперимента L3 в ЦЕРН (ВЦ для offline обработки), находившегося в 10 км от источника исходных данных (на рисунке 2 схематично показан классический подход по использованию вычислительных мощностей в физическом эксперименте на всех этапах управления данными, а на рисунке 3 показано управление потоком данных от эксперимента к центру обработки до принятия концепции грид). Это решение позволило использовать центр online-обработки, в период заполнения ускорителя, а также во время его плановых остановок, совместно с центром offline-обработки. Для конечного пользователя данная система выглядела, как единый вычислительный комплекс. Кроме того, это позволило перестать ежедневно перевозить магнитные ленты, т.к. данные передавались по мере набора в центр offline-обработки и архивировались в нем. Дальнейшее развитие ВТ, ИТ и увеличение пропускной способности глобальной вычислительной сети (WAN), позволили создать распределенную обработку для эксперимента AMS (на рисунке 4 схематично

изображены потоки данных и организация центров обработки данных для эксперимента AMS-02 на МКС).

Для эксперимента AMS существуют следующие потоки информации:

- команды, поступающие со станций контроля работы установки;
- телеметрия от станции;
- информация о состоянии детектора (в ФВЭ и ЯФ, имеющая название *slow control*: показания датчиков, измеряющих температуру, напряжение, давление, и т.д. В NASA - эти данные называются H&S - health and status);
- научные данные с установки.

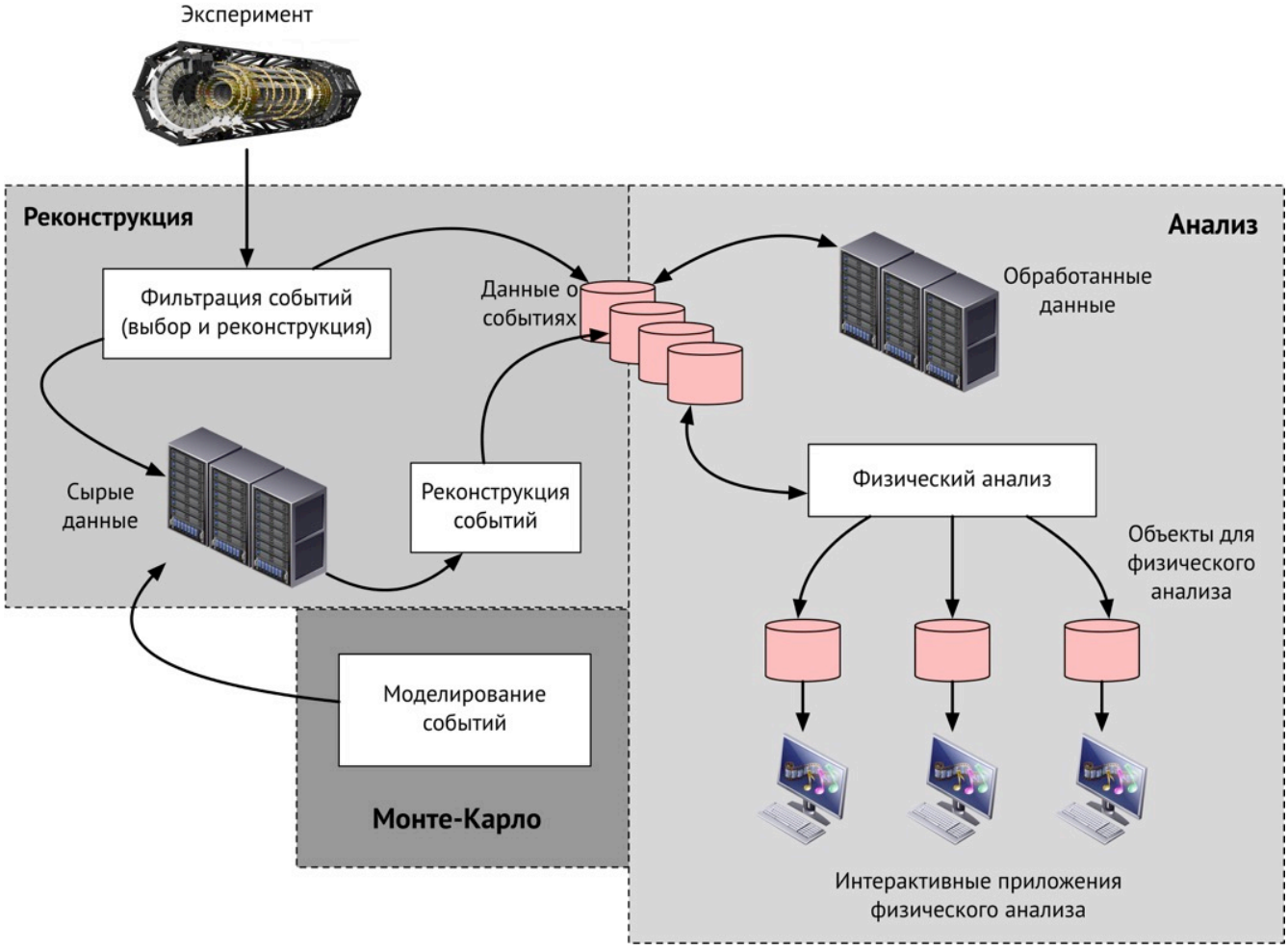


Рисунок 2 - Роль компьютинга в физике высоких энергий и ядерной физике

В AMS-02 перечисленная выше информация с МКС поступает в центр NASA (Marshall Space Flight Center - MSFC, Алабама, США), буферизуется на серверах AMS-02, установленных в MSFC. Телеметрия и H&S информация передаются в POCC (Payload Operations and Control Center - центр контроля работы AMS-02), а научные данные передаются в SOC (Science Operations Center - центр научной обработки данных), оба центра находятся в ЦЕРН. Первичная обработка данных проходит в ЦЕРН после чего данные распределяются между всеми центрами AMS-02 в Европе, США и Азии для физического анализа. Моделирование методом Монте-Карло ведется более, чем в 10 центрах по всему миру. Следует отметить, что наряду с основным центром контроля и управления (POCC) в ЦЕРН, существуют сателлитные центры во многих странах, имеющие функции контроля и мониторирования (похожая концепция была применена позже в эксперименте CMS на LHC, когда основной центр управления находится в ЦЕРН, рядом с экспериментальной установкой, а сателлитные центры находятся в ОИЯИ (Дубна, Россия) и Лаборатории Ферми (США)).

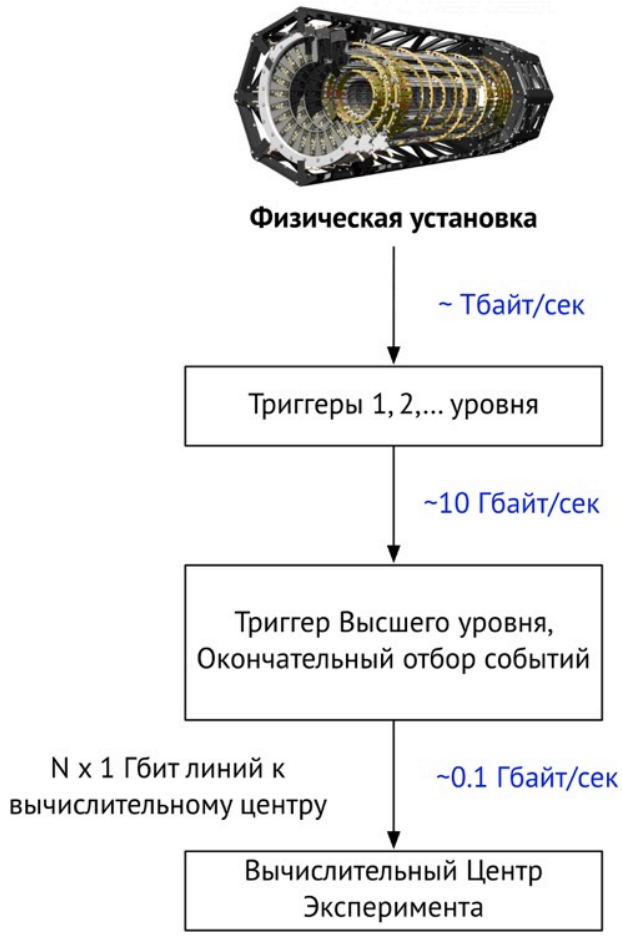


Рисунок 3 - Поток данных физического эксперимента

Концепция и архитектура системы управления данными, а также основное программное обеспечение были предложены и разработаны автором диссертации и начально реализованы для моделирования физических процессов, а после начала работы детектора на МКС и для обработки данных эксперимента [50]. В те же годы

(1997/2002) группы, работающие на ускорителях SLAC (эксперимент BaBar) и Теватрон (Лаборатория Э.Ферми, эксперименты D0, CDF), пытались найти оптимальное решение для организации обработки данных как в центральном ВЦ (находящемся географически в той же точке, где и установка, т.е. источник данных), так и в удаленных ВЦ (в том числе в других странах).

В лучшем случае, удаленные ВЦ использовались для моделирования событий и/или работы детектора методом Монте-Карло, и результаты моделирования в каждом из случаев были доступны для пользователей, после их передачи (через WAN или на магнитных носителях) в SLAC или Лабораторию имени Ферми. Долгие годы AMS был единственным экспериментов в области физики частиц, когда обработка данных и установка находились в разных географических точках [51]. Разработка и создание системы распределенной обработки данных эксперимента AMS стало первой попыткой предложить новый архитектурный подход к реализации глобальной системы обработки данных на распределенных вычислительных ресурсах.

К середине 90-х годов прошлого века стало очевидным, что дальнейшее развитие систем обработки данных для будущих ускорителей невозможно без создания новых информационных технологий, а также кардинального пересмотра компьютерной модели для экспериментов в ФВЭ и ЯФ. Следует отметить, что в это время развитие коммерческой индустрии ИТ еще не было столь стремительным, как в последующие десятилетия, и после военных приложений, исследования климата, приложения физики частиц являлись одними из наиболее информационноемких. Также следует отметить, что изначально запуск ускорителя LHC планировался в 2000/2004 годах, и это требовало разработки новых информационных технологий (а также их апробацию) в сравнительно короткие сроки.

1.1.2 Распределенная иерархическая компьютерная модель для обработки данных Большого адронного коллайдера

Грид технологии были предложены в конце прошлого века Я. Фостером и К.Кессельманом и основная концепция изложена в книге «The Grid : a Blueprint to the New Computing Infrastructure», именно задачи ФВЭ и ЯФ привели к широкому использованию грид технологий. Еще на раннем этапе развития компьютерной модели LHC (конец XX века) было принято решение объединить существующие и вновь создаваемые вычислительные центры (более 300 центров на сегодняшний день) в распределенный центр обработки данных, и сделать это таким образом, чтобы физики университетов и научных организаций участвующих стран имели равные возможности для анализа информации . Для такого решения было несколько причин:

- экономические и социологические:
  - даже предварительная оценка будущего объема данных LHC, не позволяла просто расширить существующий ВЦ, даже такой крупный как в ЦЕРН, и использовать его для хранения, обработки и анализа данных. Требовались капитальные вложения в инфраструктуру и в случае использования централизованной модели взнос стран участниц в бюджет организации мог существенно возрасти, при этом ЦЕРН должен был одновременно обеспечить строительство самой “машины” и сопутствующей инфраструктуры;
  - количество ученых, участвующих в экспериментах на LHC, уже на первом этапе заявок, было близко к 5 тысячам (в настоящее время около 9 тысяч) из более чем 50 стран мира. В случае централизованного решения, анализ данных в ВЦ ЦЕРН создавал неравноправные условия для стран, находящихся на значительном расстоянии от Женевы (таких как Россия, США, Япония, Австралия,

Канада), доступ к данным для них был бы не столь эффективен как для стран западной Европы;

- многие страны, университеты, исследовательские институты имели значительные вычислительные мощности, и были заинтересованы в их развитии и использовании;

  экономическая ситуация во многих странах мира требовала вложений в национальные проекты и создания рабочих мест в странах ЕС, поэтому идея дополнительного финансирования компьютерных мощностей ЦЕРН не была поддержана экспериментами. Одновременно идея о расширении национальных ВЦ для потребностей LHC была воспринята позитивно мировым сообществом;

- технические :
  - ни ЦЕРН, ни другие центры ФВЭ и ЯФ не имели опыта строительства ВЦ для обработки данных в мульти-петабайтном диапазоне и одновременного доступа к данным тысяч пользователей.
  - характеристики будущего центра в части потребляемой мощности и систем охлаждения не могли быть реализованы на территории ЦЕРН в Швейцарии и Франции без изменения двухсторонних соглашений организации с этими странами;
  - использование суперкомпьютера (или нескольких СК) для проведения централизованной обработки данных не позволяло решить вопрос с анализом данных, не говоря о стоимости такого решения;

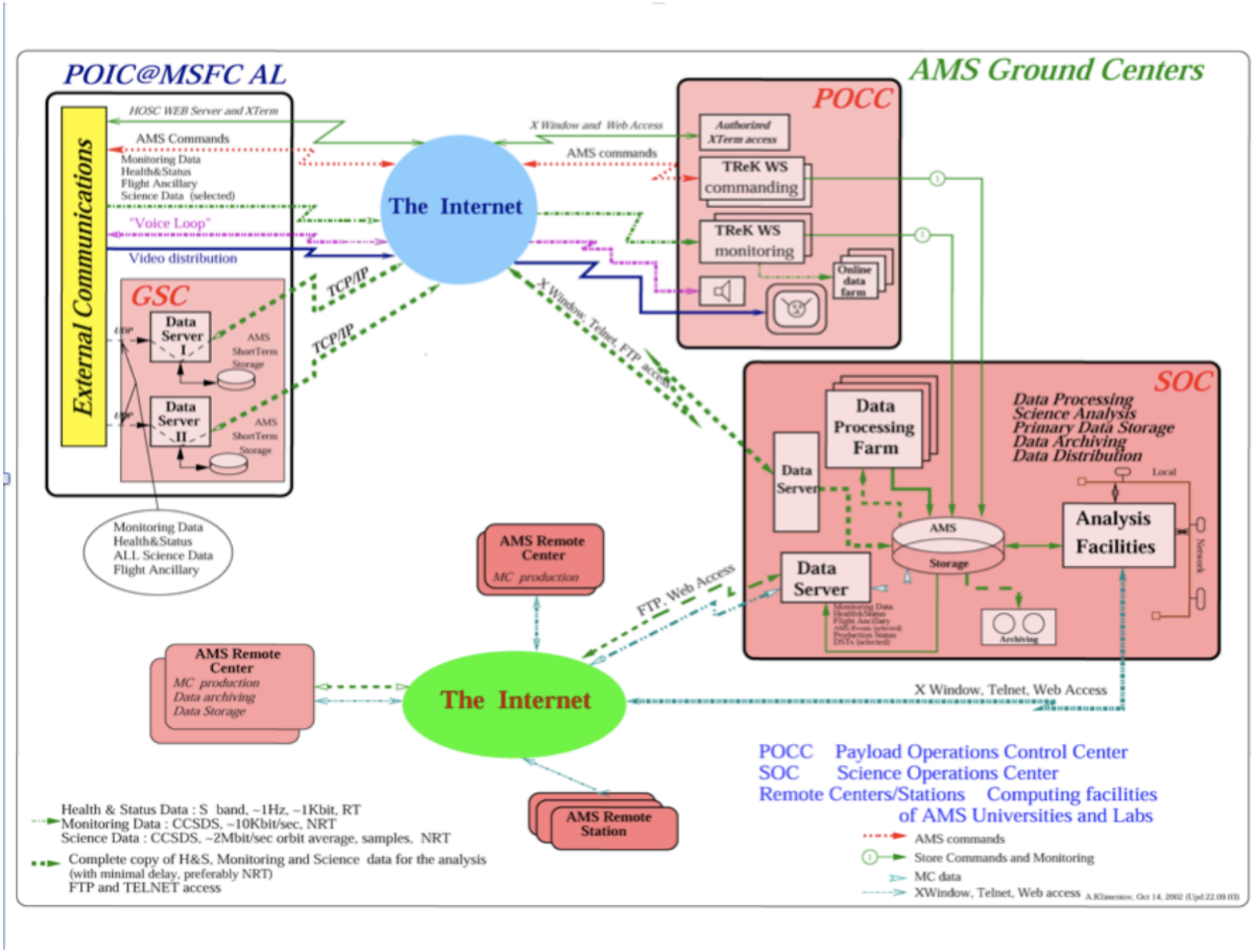


Рисунок 4 - Потоки информации и организация центров обработки данных для эксперимента AMS-02 на МКС

- ПО физических экспериментов (а это четыре миллиона инструкций кода) не было оптимизировано для суперкомпьютеров и, в частности, для параллельных вычислений и графических процессоров;
- существующие на тот момент технологии иерархического гибридного хранения данных (диск-лента), например, CASTOR [52] не позволяли эффективно перемещать файлы между постоянным (лента) и временным (диск) хранилищами с частотой и объемами, требуемыми для обработки будущих данных LHC в одном центре;
- оценка и прогнозирование возможностей WAN не гарантировала эффективный удаленный доступ к данным;
- требования к вычислительному и дисковому ресурсу значительно менялись в течение подготовки экспериментов на LHC. В таблице 2 приведено как менялась оценка необходимого ресурса для эксперимента ATLAS. Из таблицы видно, что разница в оценке необходимых мощностей составила три порядка для дискового ресурса и два порядка для вычислительного ресурса между моментом подачи меморандума о создании эксперимента и этапом начала работы ускорителя.

Ниже мы рассмотрим компьютерную модель для распределенной обработки данных применительно к экспериментам на Большом адронном коллайдере, для которых это проявилось наиболее явно. Эксперименты на KEK (Belle II), LSST, RHIC и будущих комплексах FAIR, NICA, астрофизический эксперимент AMS-02 приняли аналогичную компьютерную модель.

В 1998 году был создан проект MONARC (Models of Networked Analysis at Regional Centres for LHC Experiments) под общим руководством профессора Калифорнийского Технологического Института Харви Ньюмана.

Таблица 2 – Изменение оценки вычислительных ресурсов, необходимых для эксперимента ATLAS

Год	Дисковый ресурс (Терабайты)	Вычислительный ресурс (MIPS)	Комментарий
1995	100	10^7	Техническое предложение по ПО и вычислительным мощностям эксперимента
2001	1900	$7*10^7$	Рецензирование требований LHC экспериментов по ПО и вычислительным мощностям
2005	70000	$55*10^7$	Окончательное техническое предложения ATLAS
2011	83000	$100*10^7$	Оценка по результатам первого года работы

Задачей проекта являлась разработка компьютерной модели для экспериментов на LHC, а также пакета моделирования, позволяющего провести анализ необходимых аппаратно-программных средств и компонент (пропускная способность WAN, количество и характеристики линий передачи данных) для реализации такой модели. Перечислим основные результатами проекта MONARC :

- предложение распределенной компьютерной модели для обработки и анализа данных LHC;
- концепция иерархии центров обработки, моделирования и анализа данных ;
  - введение определения трех уровней центров
    - Тир0, Тир1, Тир2 (Tier0 (T0), Tier1 (T1), Tier2 (T2))
    - строгое определение функций для центров каждого уровня;
- создание пакета программ MonALISA [53]

– пакет MonALISA изначально позволял проводить моделирование распределенной работы центров, при условии, что архитектура отвечает критериям, определенным в проекте MONARC. По мере изменения компьютерной модели для экспериментов ФВЭ и ЯФ функции пакета стали более ограниченными, и в настоящее время он используется в одном из LHC экспериментов для мониторирования работы ВЦ, входящих в инфраструктуру грид [54].

Основным аргументом при выборе иерархической компьютерной модели послужило предположение о том, что пропускная способность WAN не позволит передавать данные в объемах необходимых для физического анализа между всеми центрами обработки. На рисунке 5 схематично представлена компьютерная модель, предложенная проектом MONARC. Следует отметить, что в последние годы XX века Калифорнийский Технологический Институт (КТИ) и Лаборатория Стэнфордского Линейного Ускорителя (SLAC, США) были “законодателями моды” в вопросах архитектуры систем обработки и управления данными, и в вопросах создания ПО для физических экспериментов. Это было связано не только с тем, что Х. Ньюман (КТИ) и Р.Моунт (SLAC) руководили на тот момент компьютингом самого крупного эксперимента ФВЭ - BaBar, но и фактом, что исторически сильные позиции Российских физиков и ИТ специалистов были “подорваны” экономической ситуацией в стране, когда многие из них перестали заниматься наукой. Кроме того, ЦЕРН не смог провести широкой дискуссии по вопросам компьютерной модели для экспериментов на LHC.

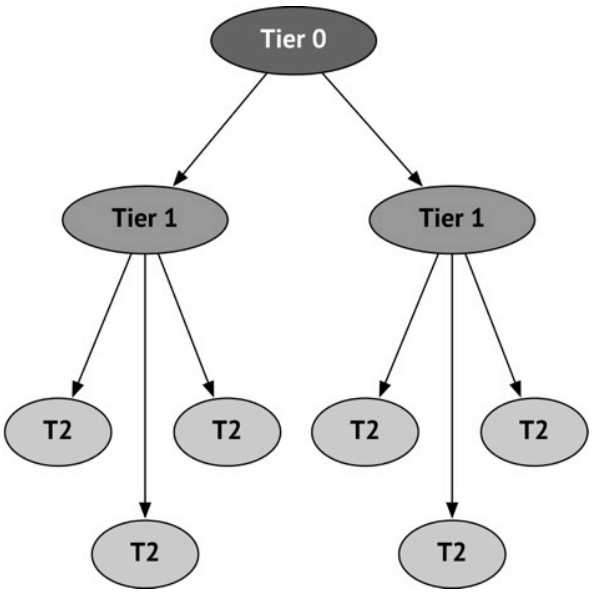


Рисунок 5 - Иерархическая компьютерная модель (проект MONARC)

Отметим, что решения “западного побережья США”: иерархическая компьютерная модель для обработки данных, строгое определение функций центров на каждом уровне и использование объектно-ориентированной базы данных Objectivity - стоили больших денег и принесли много проблем, преодоление которых заняло годы. Таким образом, в начале века сообществом ФВЭ и ЯФ была принята концепция, предложенная MONARC, и было решено использовать грид технологии для ее реализации. В конце 2001 года было предложено создать консорциум, который отвечал бы за компьютинг для LHC - World LHC Computing Grid (WLCG). Создание WLCG стало важным шагом в развитии компьютинга для экспериментов в физике частиц, потому что ни один ВЦ (каким бы большим он ни был) не имел более монополии на принятие решений. Возглавил проект Д-р. Лес Робертсон (ЦЕРН), им были определены два основных направления работы консорциума:

- вычислительные ресурсы (центры по всему миру, которые должны стать единым распределенным центром обработки данных LHC),
- программное обеспечение (которое необходимо разработать и использовать, чтобы скрыть сложности инфраструктуры и предоставить “прозрачный доступ” к вычислительным ресурсам).

Кроме того, Д-р. Робертсон в своем письме руководителям компьютинга и ПО ведущих экспериментов (автор диссертации в этот момент руководил SW&C в эксперименте AMS-02) признал необходимость широкой дискуссии и подчеркнул, что проект MONARC должен продолжать работы по развитию компьютерной модели для экспериментов на LHC, но форумом для обсуждения и принятия решений становятся рабочие совещания консорциума WLCG. В октябре 2003 года состоялось рабочее совещание, на которое были вынесены следующие вопросы:

- компьютерная модель для будущих экспериментов в области физики частиц;
- создание единого распределенного центра для обработки данных LHC. (Каким образом и используя какие технологии сотни центров должны быть организованы в “единый” распределенный центр обработки ?)

- организация и финансирование центров обработки данных (Как будет организована работа центров, кем и как они будут финансироваться ?)
- ПО для управления данными экспериментов и потоками заданий для их обработки, в случае принятия концепции распределенной компьютерной модели.

На рисунке 6 показано как должно было измениться управление потоком данных для экспериментов на LHC по сравнению с экспериментами на LEP и коллайдерах RHIC, SLAC, Теватрон.

Для новой компьютерной модели не существовало вычислительной инфраструктуры, ее необходимо было создать. Для реализации грид-инфраструктуры

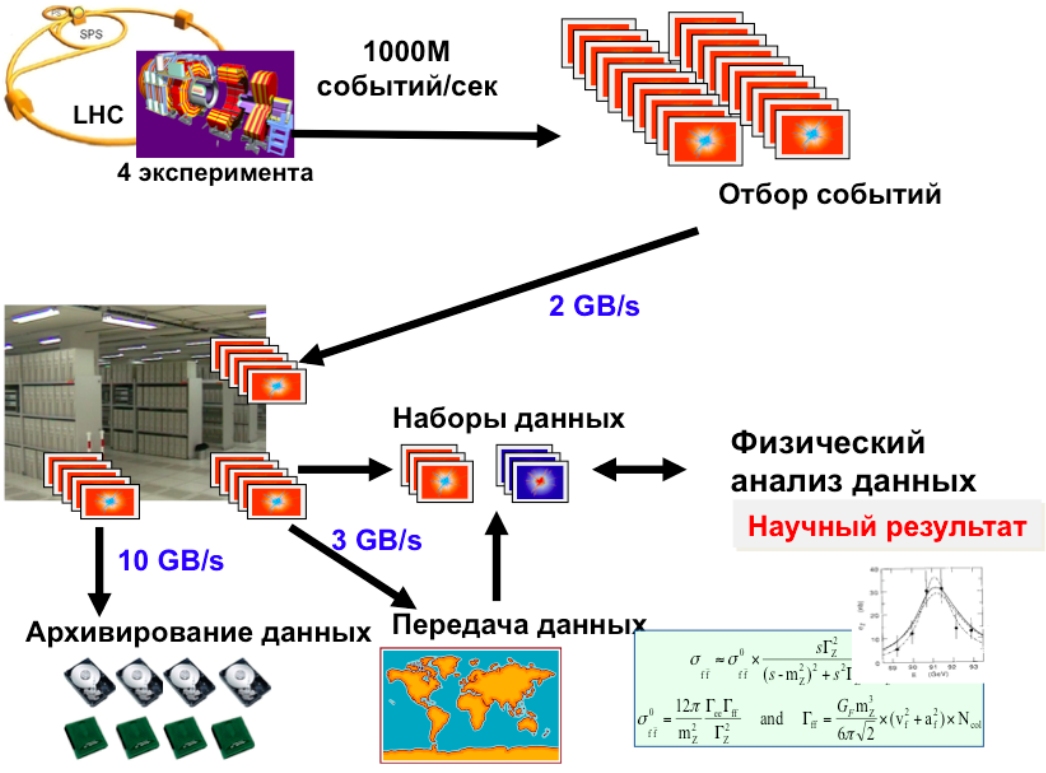


Рисунок 6 - Потоки данных экспериментов на LHC

в рамках ЕС в 2004 году был начат проект EGEE - Enabling Grid for E-Science in Europe) [55]. Проект EGEE имел три этапа, и завершился в 2010 году созданием глобальной грид инфраструктуры (параллельно с EGEE были реализованы

национальные проекты для стран, не входящих в ЕС - Open Science Grid (США); NorduGrid - Норвегия, Дания, страны Балтии, Украина, Швейцария; RDIG (Russian Data Intensive Grid) - Россия (включая ОИЯИ Дубна), аналогичные проекты были реализованы в Австралии, Индии, Китае, Тайване и Японии). В проектах участвовали около 40 стран, более ста организаций (университеты, исследовательские институты, национальные лаборатории). Как будет показано далее создание трех версий Промежуточного программного обеспечения (ППО: EGEE, OSG, NorduGrid) привело к определенным сложностям при разработке систем для обработки и управления данными. Создание грид инфраструктуры для LHC и ФВЭ и ЯФ в целом - стал важным этапом в развитии компьютинга, были определены 11 центров первого уровня (T1), более 100 центров и федераций второго уровня, центром T0 стал ВЦ ЦЕРН. Было определено соотношение разделения вычислительного ресурса между уровнями: T0 - 15%, T1 - 40%, T2 - 45% (центры уровня T3 могли предоставлять вычислительный ресурс на добровольной основе и в ограниченные периоды работы экспериментов). Были подписаны более 40 меморандумов о взаимопонимании в рамках WLCG, в которых финансирующие организации стран брали обязательства обеспечить работу центров первого и второго уровня на срок не менее трех лет.

В рамках проекта MONARC было завершено описание вычислительной модели, и определены функции центров каждого из уровней:

- Tier0 (ЦЕРН) - первичная реконструкция событий, калибровка, постоянное хранение и архивирование полного набора “сырых” и моделируемых данных. Место нахождение основных сервисов (СУБД, репозитории программ) и копий баз данных в случае, если БД находится вне ЦЕРН;
- Tier1 (11 центров) - архивирование второй копии “сырых” (неприведенных) данных, распределенной между всеми T1 центрами, переобработка данных, после уточнения калибровок, Монте-Карло моделирование, постоянное хранение копий данных используемых для анализа;

– Tier2 (около 100 центров) - временное хранение наборов данных, используемых для анализа, моделирование данных и детекторов, физический анализ.
– Tier3 (около 50 центров) - университетские кластеры, или центры, предоставляющие ресурсы на добровольной основе, физический анализ данных.

Схематично компьютерная модель для первого этапа работы экспериментов на LHC представлена на рисунке 7.

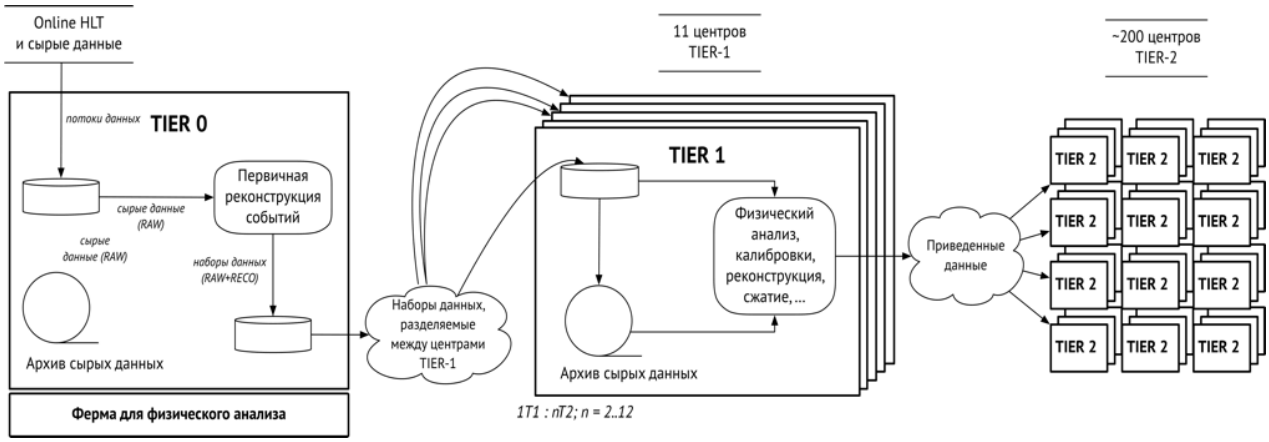


Рисунок 7 – компьютерная модель во время первого этапа работы LHC

1.1.3 Концепция Грид

Грид технологии были предложены в конце прошлого века Я.Фостером и К.Кессельманом и основная концепция грид изложена в книге «The Grid: a Blueprint to the New Computing Infrastructure» (1997 год). Появление технологии грид совпало по времени с моментом поиска новой компьютерной модели для обработки данных в ФВЭ и ЯФ, а также вводом в строй коллайдеров в США (Теватрон, RHIC) и подготовкой к запуску LHC. Как и в случае технологии Всемирной паутины (WWW), созданной в ЦЕРН для удовлетворения растущих потребностей со стороны ФВЭ к обмену информацией между учеными и совместному доступу к ней, вызвавшей бурное развитие информационных технологий и систем связи в конце XX, когда технологии WWW обеспечили бесшовный доступ к информации,

хранящейся на миллионах географически распределенных веб-сайтов. В случае грид предполагалось обеспечить бесшовный доступ к вычислительным мощностям и дисковому ресурсу в ВЦ по всему миру. Из многочисленных существующих определений грид, остановимся на следующем: “Координированное совместное использование ресурсов и решение проблем в динамичных многопрофильных виртуальных организациях” (Я.Фостер [18]). Таким образом в концепции грид отсутствовал централизованный контроль над ресурсами, вводилось понятие ‘виртуальной организации’ - совокупность институтов, университетов, групп, объединённых для решения общей задачи в режиме скоординированного использования распределенных вычислительных ресурсов, выделенных для данного проекта. С точки зрения конечного пользователя, концепция грид выглядела очень притягательно:

- вычислительный ресурс используется пользователем по потребности;
- ресурс может принадлежать неизвестному владельцу и /или находится в неизвестном месте;
- владелец ресурса гарантирует компьютерную безопасность ресурса, данных и ПО пользователя;
- программа пользователя будет выполнена на грид ресурсе;

С точки зрения владельца ресурса концепция грид выглядела следующим образом:

- “мой” вычислительный ресурс может быть использован любым авторизованным пользователем;
- “авторизация” не связана административно с организацией, которой принадлежит ресурс
- ресурс предоставляется не бесплатно;

Важнейшей частью концепции грид-технологий явилось введение понятия - промежуточное программное обеспечение (ППО, middleware). Первым проектом по созданию ППО явился проект globus [56], который был инициирован Я.Фостером и

К.Кессельманом, ими же были определены основные компоненты (и подсистемы) грид-архитектуры. Кратко рассмотрим основные компоненты грид архитектуры (более подробно это описано в работах Я.Фостера, К.Кессельмана и В.В.Коренькова [18,19], мы остановимся на том, что важно для обсуждения развития компьютерной модели и систем для глобально распределенной обработки данных):

Вычислительный элемент (Computing Element, CE) — это вычислительный ресурсный узел грид. На CE выполняются задания пользователей и происходит управление заданиями (запуск, остановка по ошибке и/или по запросу пользователя, или по истечению ресурса, например, оперативной памяти). Состояние и описание ресурсов всех CE публикуется в центральном сервисе (информационной системе) и доступно для всех авторизованных пользователей. Управление загрузкой CE производится через единую систему управления загрузкой.

Элемент хранения (Storage Element, SE) - это узел грид, где хранятся результаты выполнения заданий пользователей на CE. Управление данными, хранимыми на SE производится через систему управления данными. Состояние и описание ресурсов всех SE публикуется в центральном сервисе (информационной системе) и доступно для всех авторизованных пользователей.

Система управления данными (Distributed Data Management - DDM) - система управления данными, включая хранение, передачу и удаление данных. DDM работает с данными на уровне файлов, конкретные реализации DDM высокого уровня часто предполагает использование набора данных (dataset), как единицы управления данными (например, передача между центрами грид производится на уровне dataset). Контроль доступа к данным основан на понятии групп, которые могут определяться, как для всей виртуальной организации, так и для ее отдельных членов.

Система управления загрузкой (Workload Management System, WMS) - система управления пользовательскими заданиями, и распределением их для выполнения на грид ресурсах. WMS выбирает ресурс в соответствии с параметрами

задания, находит оптимально подходящий по параметрам грид ресурс (память, свободное дисковое пространство для промежуточного хранения, географическое расположение входных данных, требование куда должны быть помещены выходные данные).

Система протоколирования (Logging and Bookkeeping, LB) - отслеживает выполняющиеся в грид инфраструктуре шаги выполнения заданий, хранит информацию о времени затраченном на каждый шаг выполнения задания (запуск, инициализация,...). Иногда часть функций LB интегрировано с системой управления заданиями, как будет показано ниже в новейших системах эта информация используется для предсказания поведения WMS и обнаружения аномалий в ее работе.

Система информационного обслуживания (Grid Information System, GIS, часто называемая: информационная система - ИС) - данная система отвечает за хранения информации о ВЦ, входящих в грид инфраструктуру, включая информацию о вычислительной мощности узлов сайта, планируемой остановке ВЦ (например, профилактическое обслуживание).

Система мониторирования и учета (аккаунтинга) работы грид - основная система для контроля стабильности и эффективности работы грид инфраструктуры. Сохраняет информацию о качестве работы грид-сайта на протяжении всего времени его функционирования, одновременно предоставляя информацию в режиме реального времени о состоянии грид сайта службами, осуществляющими эксплуатацию ВЦ.

Система компьютерной безопасности (СБ). Система компьютерной безопасности должна защитить доступ к грид инфраструктуре и данным от несанкционированного доступа. СБ рассматривается как средство защиты веб-сервисов и, как правило, реализовано в виде отдельных модулей для сервисов Apache, Axis, Tomcat.

Таким образом в архитектуре грид, основным элементом становится грид-сайт, состоящий из набора “вычислительных элементов” и “элементов хранения”. Описание сайта хранится в информационной системе грид, и на сайте установлено промежуточное программное обеспечения, которое обеспечивает доступ к CE и SE элементам для авторизованных пользователей через системы управления загрузкой и данными (WMS и DDM соответственно).

На рисунке 8 схематично показано взаимодействие различных компонент грид инфраструктуры.

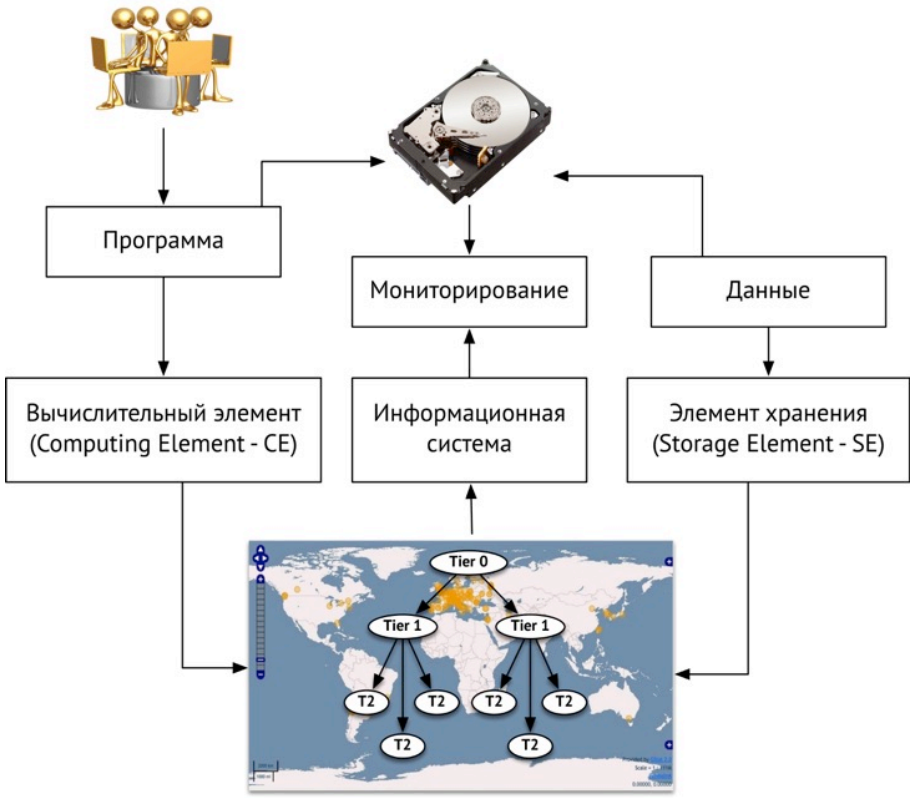


Рисунок 8 - Компоненты грид инфраструктуры и их взаимодействие между собой

1.2 Реализация иерархической компьютерной модели распределенной обработки данных на первом этапе работы Большого адронного коллайдера.

Таким образом к 2004 году были определены компьютерная модель для экспериментов на LHC и метод реализации компьютерной инфраструктуры на основе технологии грид.

Иерархическая модель MONARC предполагала статическое соответствие 1:n между центрами уровней T1 и T2 (T3), в предположении, что такие “связки” будут созданы по географическому и/или национальному признаку и формируют группы ВЦ на постоянной основе. В силу причин политического характера центры T2 в Японии и в Китае оказались в “связке” с центром T1 в Лионе (Франция), а не T1 в Тайпее (Тайвань). В силу причин социологического характера T2 центры России, Израиля и Турции оказались в одной “связке” с центром в Амстердаме (т.к. Нидерландам “не хватило” центров в ЕС). Швейцария оказалась в “связке” с Норвегией. Модель MONARC не допускала нарушения иерархии и отсутствия предопределенного соотношения T1:nT2. Реализация модели была завершена в 2009 году. К моменту запуска LHC были определены и зафиксированы обязательства всех центров в рамках консорциума WLCG. Все центры консорциума предоставляли “выделенный” ресурс в течение всего времени участия в WLCG, что как будет показано далее не является оптимальным для использования мощностей конкретного вычислительного центра и всего ресурса доступного для физики частиц в целом. На рисунке 9 показана реализация модели MONARC на момент запуска LHC (2009 год), на рисунке схематично показано сайты уровня T1 и T2 и их основные функции.

Реализация модели распределенных вычислений, предложенная проектом MONARC – стала значительным шагом в развитии компьютинга в области физики частиц. Более 200 центров в 60 странах мира вошли в консорциум WLCG, был получен первый опыт по распределенной обработке данных. Вычислительные ресурс

распределялся следующим образом: 15% находилось в ЦЕРН (уровень T0), 40% распределялось между 11 центрами уровня T1 (данное распределение было крайне неравномерным между 11 центрами, так для экспериментов ALICE, ATLAS и CMS вклад центров уровня T1 варьировался от 5% до 40%), 45% ресурса распределялось между центрами уровня T2.

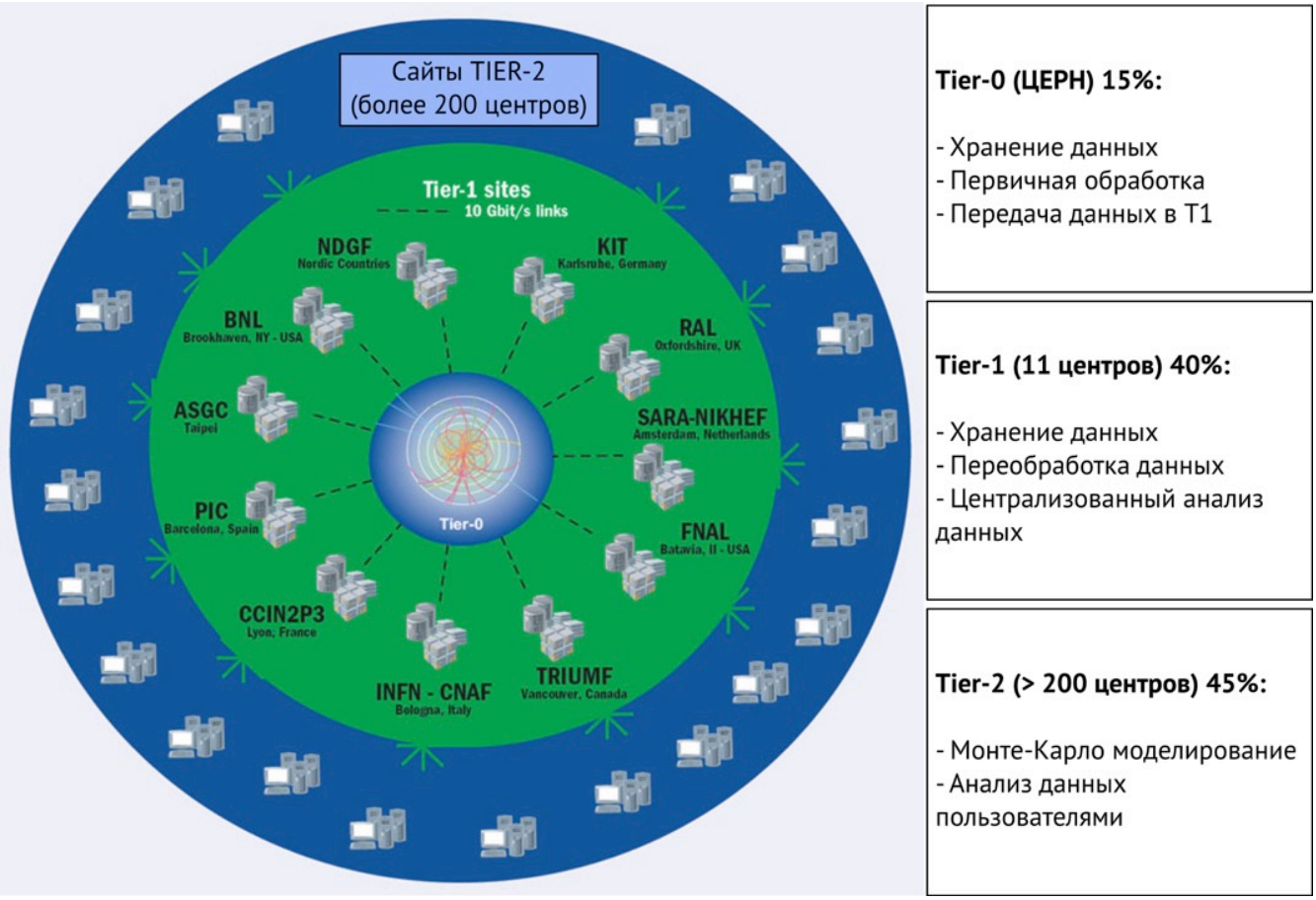


Рисунок 9 - Организация грид сайтов WLCG на момент запуска LHC

Следует отметить, что реализованная модель успешно работала в течение первого этапа работы коллайдера, но поддержание ее в рабочем состоянии требовало больших человеческих затрат, как со стороны ВЦ (инфраструктура), так и со стороны научных коллабораций для поддержания работы сервисов управления данными и загрузкой. Кроме того, ожидание, что гиганты индустрии ИТ (Google, Amazon, Яндекс, Microsoft) будут использовать технологии грид и тем самым способствовать развитию промежуточного программного обеспечения, не подтвердились, поэтому необходимо

было проанализировать первый опыт по реализации иерархической модели распределенной обработки данных и определить дальнейшее развитие компьютерной модели для физики частиц в целом, исходя из реалий и появления новых игроков на поле ИТ. Этого требовали, как подготовка ко второму этапу работы ускорителя LHC, так и подготовка экспериментов на будущих комплексах : КЕК (эксперимент Belle II), FAIR и NICA.

1.3. Ограничения иерархической компьютерной модели MONARC

Реализация модели распределенных вычислений, предложенная проектом MONARC, стала значительным шагом в развитии компьютинга в области физики частиц. В тоже время, уже на первом этапе работы LHC проявились существенные ограничения данной модели, перечислим основные из них:

- Определение вычислительного ресурса как совокупности вычислительных узлов, дискового пространства и систем архивирования информации, без учета пропускной способности WAN, и качества линий связи.
- Статическая методика распределения данных между центрами : а) было определено изначально, какой объем данных (реальных и моделируемых) будет находиться в каждом центре; б) было определено изначально, сколько копий данных каждого типа («сырых» и приведенных) будет распределено между центрами обработки.
- Отсутствие понятия «популярности» (востребованности) для данных и групп данных.
- Предложенная методика обработки данных при статическом характере организации вычислительного ресурса и распределения данных между центрами грид инфраструктуры. Наиболее точно ее можно определить слоганом : «задачи обработки идут к данным». Такой подход, привел к задержке при обработке и моделировании данных, так как требовал

одновременного наличия данных и свободного вычислительного ресурса в одном и том же ВЦ.

- Вычислительный ресурс центров был ориентирован на среднюю загрузку. Как результат это вело к недостатку вычислительного ресурса в периоды пиковой нагрузки (работа коллайдера с повышенной светимостью), анализ данных в период, предшествующий основным научным конференциям, сверка гипотез между несколькими научными группами и/или экспериментами и к неоптимальному использованию вычислительного ресурса во время плановых остановок коллайдера, праздников, и т.д.
- Ограничения самой модели, предпологающй гомогенность используемого ресурса, наличие ПО промежуточного уровня («middleware») во всех центрах обработки данных.

Основной проблемой стала реализация идеи иерархии ВЦ и статический характер связки центров 1:T1-n:T2, когда любой сбой в работе центра первого уровня (T1) практически останавливал работу всех связанных с ним центров второго уровня (T2), в результате эксперименты лишались мощностей до 10 центров одновременно. Кроме того, многие центры уровня T2 были мощнее и стабильнее центров уровня T1, но ресурс уровня T2 не использовался оптимальным образом, так в рамках модели результаты выполнения заданий всегда должны быть быть переданы в центр уровня T1 и только после этого могло быть создано дополнительное количество копий. Сама передача данных между центрами T2 включала до двух промежуточных копий, временного хранения в центрах уровня T1.

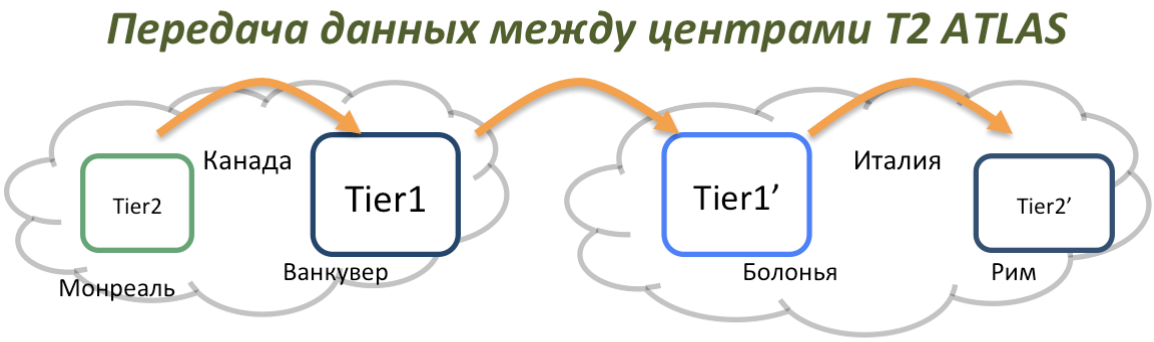


Рисунок 10 - Многоступенчатая передача данных между центрами уровня Т2 при реализации модели MONARC

Эти причины послужили мотивацией для разработки новой концепции распределенной обработки данных и новой модели компьютинга для второго и последующих этапов работы LHC. На рисунке 10 схематично показано, как происходила передача данных между двумя центрами уровня Т2, находящихся в разных странах, при реализации модели MONARC, из рисунка следует, что требовалось по крайней мере две дополнительные передачи с промежуточным хранением данных в двух центрах.

1.4 Разработка новой компьютерной модели для распределенной обработки данных. Переход от иерархической модели обработки к смешанной модели в рамках грид инфраструктуры

При разработке новой модели были введены следующие определения:

- Популярность данных. Насколько часто задачи обработки, анализа или моделирования обращаются к данным определенного типа. Насколько данные популярны у ученых, и научных групп, как часто поступают запросы на копирование данных.
- Температура данных. Как со временем меняется частота обращения к определенному набору данных.
- Вычислительная среда: вычислительный ресурс, дисковый ресурс и ресурс архивирования, пропускная способность и стабильность

глобальной вычислительной сети. Т.е. было предложено рассматривать ресурс глобальной вычислительной сети (WAN) совместно с вычислительным ресурсом и ресурсом хранения данных.

- Отсутствие предопределения функций центров внутри среды. Центры уровня Т2 могут выполнять те же функции, что и центры уровня Т1 (кроме архивирования данных).
- Оценка стабильности работы центров, и как результат решение об использовании их дискового ресурса в качестве постоянного или временного, независимо от уровня центра в классификации WLCG. Также стабильность работы центра стала влиять на выбор центра для выполнения высокоприоритетных задач (например, задачи триггера высшего уровня, которые должны быть выполнены в течение 12 часов)
  - для этого была разработана методика постоянной проверки стабильности работы центров грид инфраструктуры (она подробно рассмотрена в разделе 1.4.3) и на основании результатов проверки была введена классификация стабильности центров.

1.4.1 Методика определения популярности данных. Классификация данных.

Шаги обработки данных в ФВЭ и ЯФ показаны на рисунке 11.

Единичным объектом при обработке данных является событие, созданное на этапе набора информации с экспериментальной установки и прошедшее сито отбора, согласно меню триггеров разных уровней (состав и количество меню соответствуют научной программе эксперимента). Такое событие называется «сырым» или «неприведенным» (RAW). Все события являются независимыми, что позволяет применить тривиальный параллелизм при их обработке (пример части “сырого” события показан на рисунке 12).

Рисунок 11 - Шаги обработки данных в ФВЭ и ЯФ

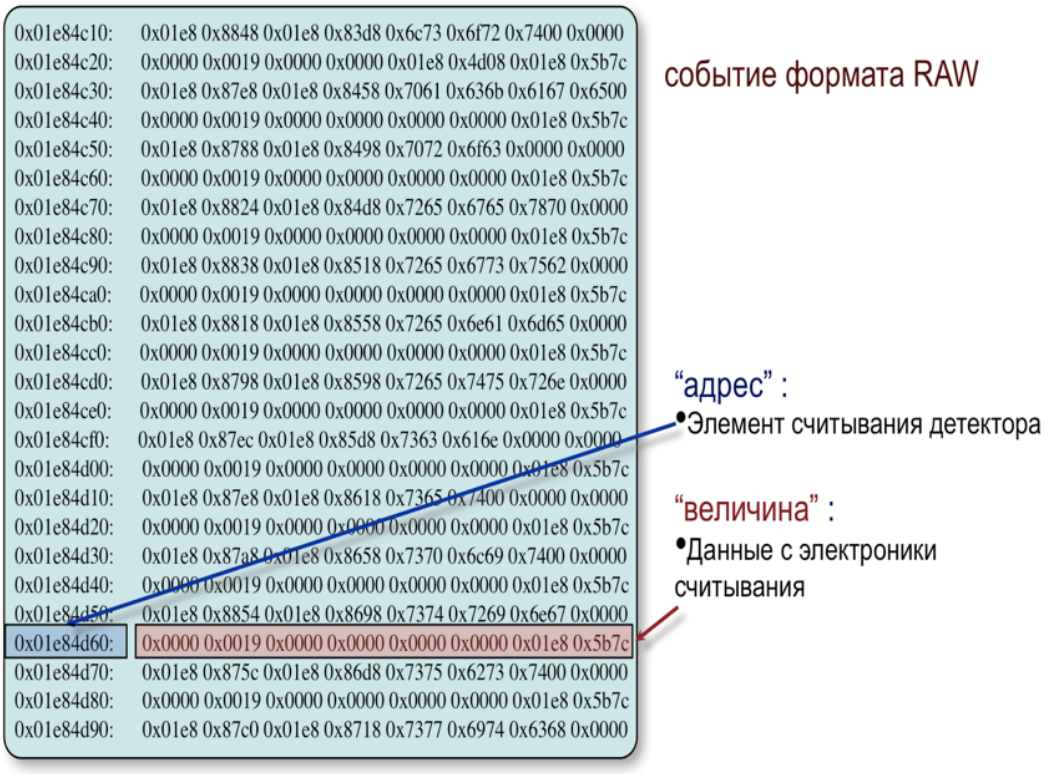


Рисунок 12 - Пример содержимого неприведенного (“сырого”) события

На первом этапе обработки проводится реконструкция события, когда восстанавливаются треки частиц, определяется масса и заряд частиц, импульс и другие параметры, на этом же этапе учитываются калибровочные параметры и неэффективность работы отдельных элементов экспериментальной установки (результат работы программы реконструкции помещается во временное хранилище в формате ESD – Event Summary Data). На втором этапе обработки создаются данные в формате необходимом для проведения физического анализа (AOD – Analysis Object Data), окончательным этапом является дополнительный отбор событий и запись информации в табличном виде (формат NTUP, DAOD) удобном для программ анализа данных (например, ROOT [57]). На рисунке 13 показана последовательность шагов обработки для реальных событий. Для этапа моделирования методом Монте-Карло необходима генерация событий (результат записывается в формате EVGEN), отцифровка событий и имитация “реальных сырых событий” (формат HITS).

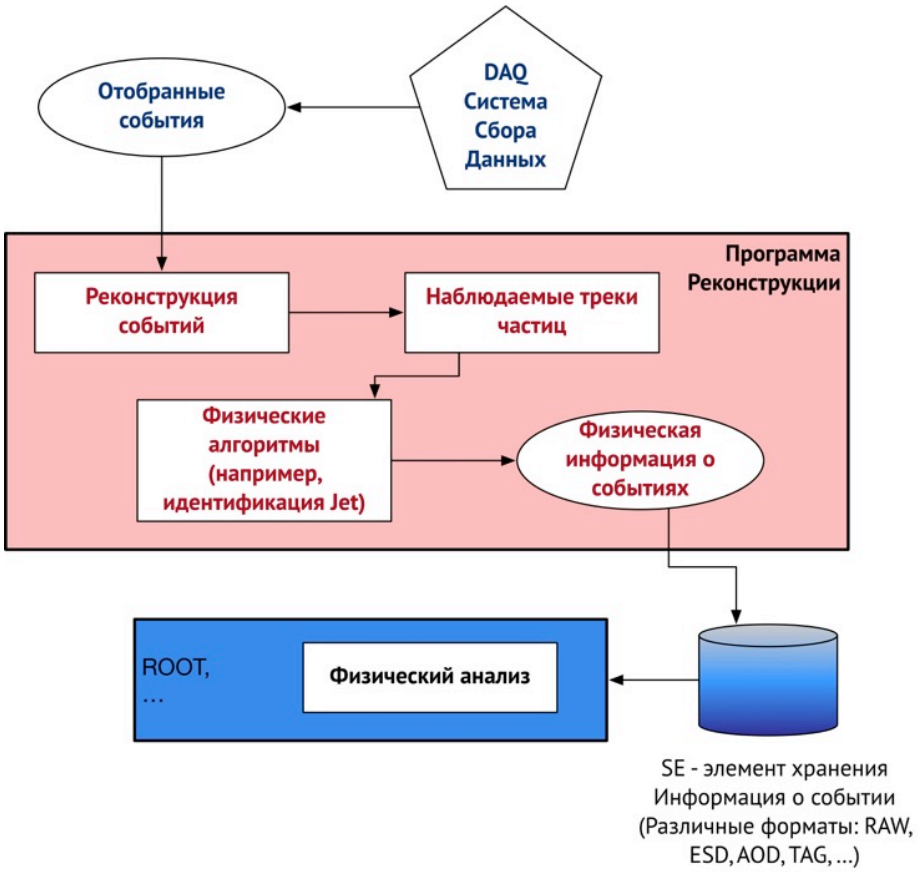


Рисунок 13 - Последовательность шагов обработки для реальных данных

Было введено понятие “класс данных”. Состав класса определялся следующими параметрами: тип события (моделированные или реальные данные), шаг обработки, формат события (RAW, ESD, AOD, NTUP…), версия программного обеспечения, используемого для его обработки, ценностью информации и затратами на ее восстановление (на рисунке 14 показана последовательность шагов при моделировании событий. Важной особенностью является “разбиение” на этапы и запоминание (иногда временное) результатов работы каждого этапа, что позволяет существенно уменьшить время необходимое для моделирования работы установки и физических процессов, более подробно это рассмотрено в третьей главе диссертации).

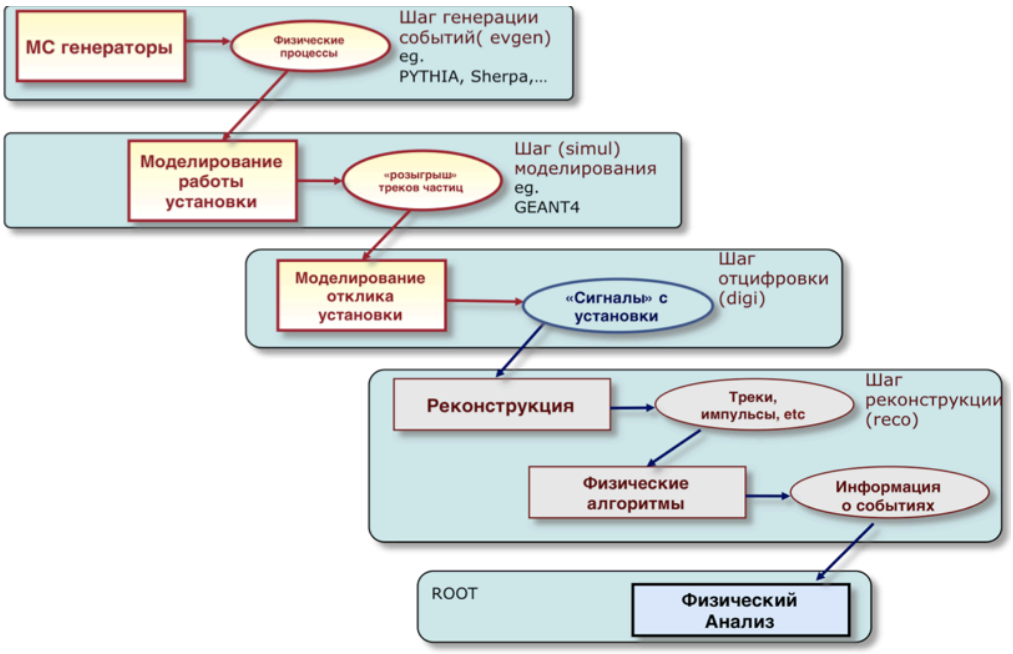


Рисунок 14 - Последовательность шагов обработки для моделируемых событий

Так моделирование событий (этап EVGEN) является наиболее компьютероемким (на рисунке 15 показан в процентах астрономическое процессорное время, затрачиваемое при обработке различных типов данных эксперимента ATLAS. Из рисунка следует, что этап Монте-Карло моделирования требует наибольшего ЦПУ ресурса).

Иерархическая модель данных. Была предложена следующая модель данных.

“*Событие*” - результат моделирования или полученное с физической установки в результате “слияния” информации со всех систем считывания для единичного столкновения ускорителя. *Событие* может быть “сырым” или реконструированным;

“*Файл*” - группа событий, как правило набранных или обработанных последовательно;

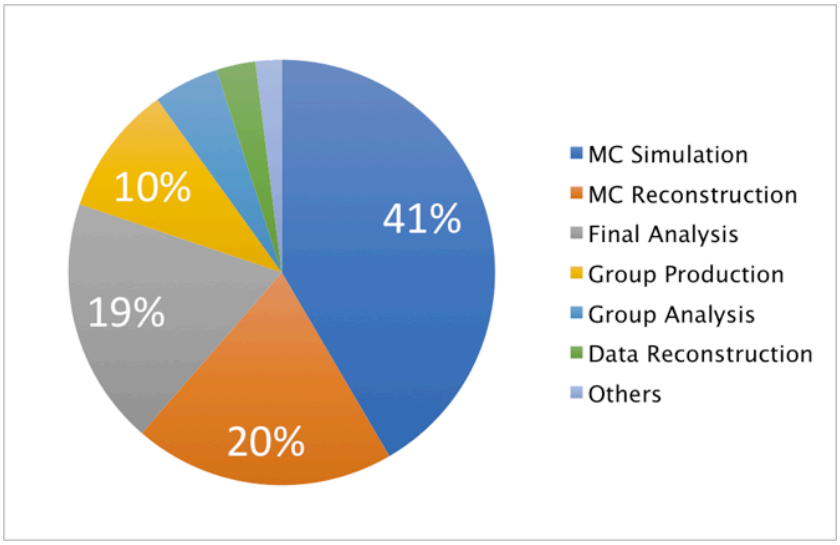


Рисунок 15 - Астрономическое время, затрачиваемое на различных этапах обработки и моделирования данных эксперимента ATLAS

“*Датасет*”. Для лучшей гранулярности и организации было введено понятие “набор данных” (от англ. dataset), состав датасета составляли файлы с событиями одного формата и созданные одной версии ПО, произведенные за определенный промежуток времени (один этап набора статистики при одинаковых условиях и параметрах отбора (*run*) при одном заполнение коллайдер – fill).

“*Контейнер*”. Контейнер содержит наборы данных (*датасеты)* одинакового формата, и созданные/обработанные одинаковой версией ПО, например, для работы ускорителя определенного периода (с одинаковой энергией и светимостью машины). Эта модель была применена для организации данных эксперимента ATLAS.

По модели MONARC все классы данных имели определенной количество копий и распределялись между центрами грид согласно MoU, заключенным ВЦ в рамках WLCG. При этом дисковый ресурс центров уровня T2 не мог быть использован для длительного (месяцы) или постоянного хранения данных. Такая концепция привела к тому, что к середине 2012 года дисковое пространство было заполнено данными формата ESD, в предположении, что именно эти данные наиболее востребованы. Диски T1 центров были переполнены, при значительном свободном дисковом пространстве в центрах уровня T2.

Классификация наборов данных и их номенклатура были выполнены в рамках эксперимента ATLAS. Данная работа была опубликована как препринт ATLAS [58] и до сих пор является основным документом, где описывается номенклатура данных эксперимента. В работе были сформулированы базовые определения классов данных, предложенные автором диссертации. После введения классификации данных, необходимо было определить их значимость и популярность. Так «сырые» события составляют отдельную группу, они являются наиболее ценными, их утрата по любой из причин, не может быть восполнена. Поэтому для «сырых данных» была выбрана следующая модель: полная копия “сырых” данных архивируются в ЦЕРН и их вторая копия распределяется и архивируется между центрами уровня T1 (т.о. всегда существуют две копии “сырых” данных). Одна копия “сырых” данных для последнего периода работы ускорителя (как правило 2 месяца) распределена между центрами T1 и T2 (выбор центров основан на стабильности их работы и имеющемся дисковом пространстве) и используется для изучения работы детектора и/или экспресс обработки. Для остальных классов данных было введено понятие популярности данных. Для этого были классифицированы методы доступа к данным: копирование за пределы распределенной системы обработки, чтение информации программами анализа индивидуальных ученых (при этом учитывались количество запросов, география и количество индивидуальных ученых), доступ к данным физическими группами. Центральная обработка и моделирование данных не

учитывались как доступ, но эта информация использовалась на следующем этапе для определения “температуры” данных (на рисунке 16 показано количество запросов пользователей на дополнительные копии данных). Была введена система весов, когда общее количество запросов нормировалось на количество ученых требовавших доступа к данным.

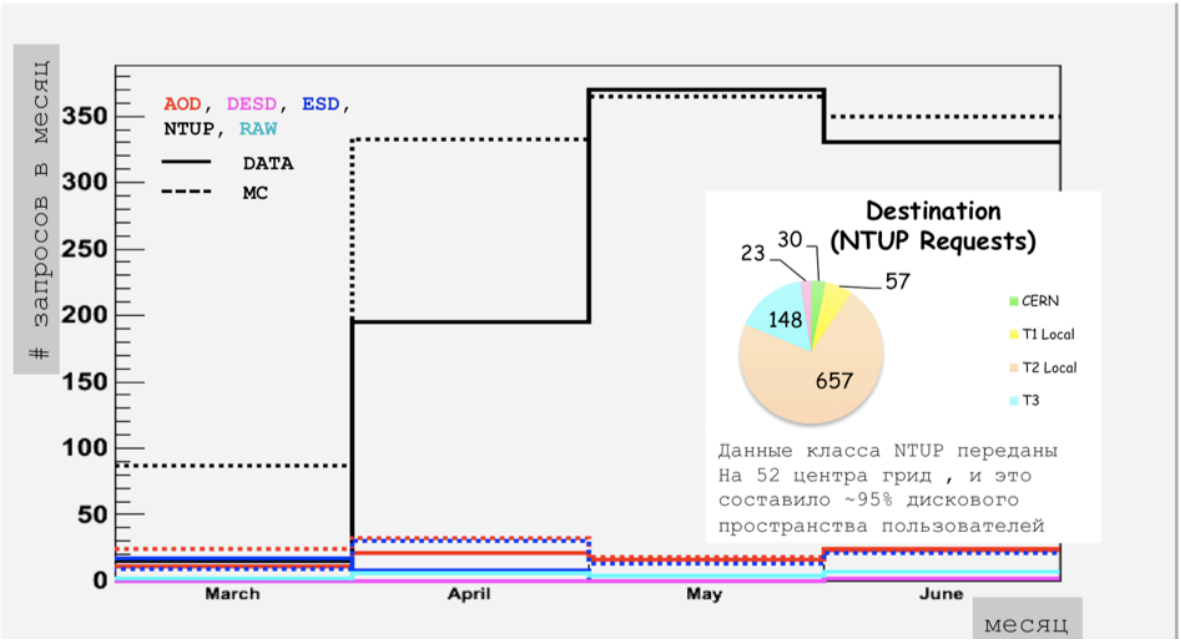


Рисунок 16 - Количество запросов пользователей на создание дополнительных копий наборов данных в распределенной инфраструктуре

Таким образом была проанализирована информация о популярности всех классов данных для реальных и моделированных событий. В результате анализа было определено, что наиболее популярными являются данные классов AOD и NTUP, а не ESD, и именно для них необходимо наибольшее количество копий, с учетом географического распределения пользователей. На рисунке 17 показано количество обращений задач анализа данных к различным классам данных в центрах уровня T1 и T2. Из графика видно, что форматы AOD и NTUP являются наиболее популярными для данного класса задач. Одновременно достаточно одной копии данных формата ESD, т.к. эти данные используются только группами, изучающими работу детектора, и в основном централизованно. Разработанная методика

определения популярности данных сохранилась до сих пор, в 2016/17 годах к методам вероятностного анализа [59] были добавлены методы “машинного обучения” для определения популярности данных в системах управления данными и управления потоком загрузки [60]. Следующим этапом стала работа по анализу популярности данных внутри класса, для этого была предложена термодинамическая модель данных, рассматриваемая в следующем разделе.

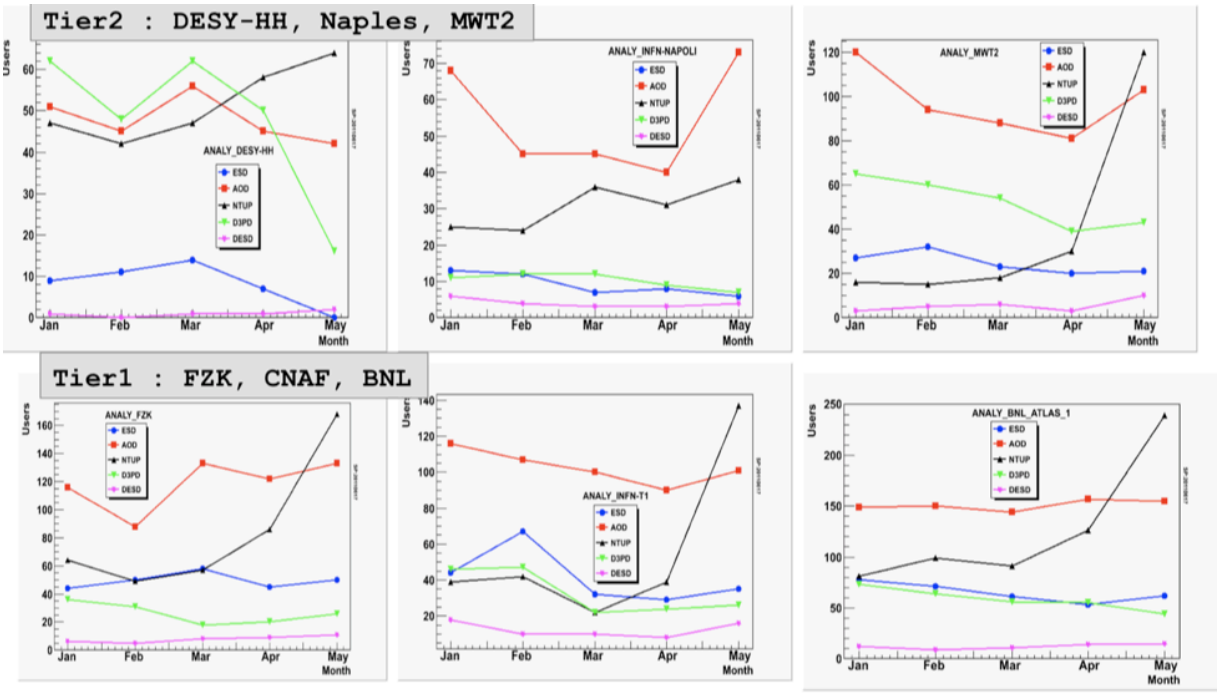


Рисунок 17 - Количество обращений задач анализа данных к различным классам данных в центрах уровня T1 и T2

1.4.2 Термодинамическая модель данных

Целью создания модели явились потребности экспериментов LHC определить методику управления данными (распределение наборов данных между центрами грид, удаление данных, архивирование “непопулярных данных”). Реализация методики привела к созданию автоматической системы управления данными для экспериментов в области ФВЭ, использующих грид инфраструктуру.

В модели под набором данных подразумеваются как моделируемые данные, так и полученные с экспериментальной установки. Единицей управления данными является датасет, как это было определено ранее.

Базовые предположения были сформулированы следующим образом:

- детальная классификация типов данных:
    - физические эксперименты имеют различные категории/классы данных - “сырые” данные, получаемые непосредственно с экспериментальных установок, (RAW), события, моделируемые методом Монте-Карло (RDO), и отцифрованные RDO событие, называемые “срабатывания” (HITS), реконструированные данные (ESD) и производные данные (AOD, DPD, NTUP, TAG), используемые для физического анализа;
- каждый класс данных может иметь различное время жизни (от “хранить вечно” для данных класса RAW, до нескольких месяцев, данные класса ESD);
- метод управления данными зависит от их класса;
- все дисковое пространство, принадлежащее эксперименту в рамках грид инфраструктуры (для экспериментов на LHC такой инфраструктурой является WLCG), рассматривается как единый ресурс;
    - это стало принципиальным отличием от модели MONARC, где при иерархии центров ресурсы уровней T0, T1 и T2 рассматривались изолированно;
- вводится понятие “центр хранения данных”, таким центром может быть центр уровня T0, T1 и T2, при условии его стабильной работы и наличия ресурса WAN для скоростной передачи данных;
    - скоростная передача данных была определена необходимостью подключения центра к выделенной сети LHCOPN или LHCONE, но не ограничена ими;

- “стабильная работа” определялась согласно результатам системы учета обращений к данным, принятой в грид инфраструктуре, с допустимым отклонением от параметров работы описанном в меморандуме о взаимопонимании (MoU) не более чем 1%, а также постоянным мониторированием стабильности работы WAN для каждого центра с использованием системы perfSONAR [61]. Информация о стабильности центра автоматически обновлялась в информационной системе каждые 12 часов.
- это было принципиальное изменение концепции MONARC, и первой попыткой начать эволюционный переход от иерархической модели к “смешанной” модели грид, кроме того большую роль в использовании ВЦ начинала играть стабильность и производительность WAN.
    - социологически это стало очень важным шагом по привлечению руководства T2 центров к работе над новой компьютерной моделью для LHC. При таком подходе роль наиболее стабильных T2 центров возрастала и их участие становилось более заметным;
- данные класса RAW распределяются для архивирования на основе Меморандума о взаимопонимании, в котором определены параметры каждого центра уровня T1 в рамках WLCG, полная версия данных хранится в ЦЕРН;
- первичная копия приведенных данных (классы: ESD, AOD,...) всегда хранится в центре, где она была произведена;
- удалению первичной копии данных должно предшествовать архивирование этой копии (за исключением централизованно удаляемых наборов данных в случае обнаружения ошибки при их создании, или уточнению калибровочных констант);

- удаление копий данных всегда осуществляется централизованно, действия протоколируются, протокол управления данными хранится в течение всего времени жизни эксперимента;
- “важность” данных и “популярность” данных не являются синонимами.
  - “важность” данных определяется физической программой эксперимента;
  - “популярность” данных “измеряется” на уровне набора данных (датасет), по количеству обращений к данным, с использованием средств обработки, анализа, репликации и т.д., с учетом частоты обращения, географии и временного интервала;

Была введена температурная шкала для всех данных, согласно показаниям шкалы состояние (температура: Т) данных могли быть :

- Горячей, Теплой, Холодной, Замороженной, Устаревшей (слово “устаревшие” было выбрано по этическим соображением, чтобы не называть значение температуры данных “мертвой”) .

«Горячие данные». Данные, широко используемые учеными и физическими группами эксперимента для проведения физического анализа и для исследования работы детектора. К таким данным всегда относятся, данные класса RAW для последнего периода набора данных (2-3 месяца), и приведенные данные последнего года. Для “горячих” приведенных данных всегда существует несколько копий. Копии размещаются в грид инфраструктуре согласно географии доступа к данным (например, все копии не могут быть размещены на одном континенте). Первичная реплика “горячих данных” не подлежит удалению. Дополнительные копии горячих данных создаются автоматически по мере необходимости (этот метод будет рассмотрен в следующем разделе).

“Теплые данные”. Данные, используемые отдельными физическими группами и учеными для физического анализа. Предполагается, что уже имеется достаточное количество копий этих данных в грид инфраструктуре. Дополнительные

копии можно запросить через интерфейс запроса передачи данных, запросы будут одобрены согласно стандартной процедуре утверждения передачи данных (подробно созданная система и ее реализация описаны в работе [62], эти разработки были проведены под руководством и непосредственном участии автора в рамках разработки новой компьютерной модели). Для “теплых” данных гарантировалось по крайней мере две копии. Т “теплых данных” всегда может повышена, количество копий может быть увеличено автоматически. При уменьшении количество копий для “теплых данных” первоначально удаляются данные на уровне Т1 и в ВЦ, где отмечен наименьшая частота доступа к ним. Копии “теплых данных” удаляются в случае запроса на свободное дисковое пространство, например, перед “началом” переобработки данных.

“Холодные данные”. Данные, используемые отдельными физиками или рабочими группами. Полная копия “холодных данных” распределена между центрами грид инфраструктуры, дополнительные копии могут быть созданы только на дисках, принадлежащих отдельным группам / пользователям, но не могут быть размещены на дисках, предназначенные для всего эксперимента. Т “холодных данных” всегда может быть повышена, при этом количество копий увеличивается автоматически. Массовое удаление данных (уменьшение количества копий) внутри инфраструктуры автоматически начинается при понижении Т с “горячая/теплая” до “холодной”.

“Замороженные данные”. Данные «практически» не используются. Сохраняется одна полная копия (в центре, где данные были произведены или повторно обработаны). Это может быть копия на диске или магнитной ленте. Дополнительные копии «замороженных» данных могут быть созданы только на дисках, принадлежащих отдельным группам / пользователям, но не могут быть размещены на дисках, предназначенные для всего эксперимента. Данные класса RAW являются исключительным случаем, для них всегда сохраняется две

архивируемые копии (причем одна из них находится в ЦЕРН, а вторая в центрах уровня T1);

«Устаревшие данные» Данные не могут быть использованы для физического анализа или других исследований в эксперименте, они подлежат удалению со всех сайтов и из всех каталогов. Примером “устаревших данных” являются приведенные данные произведенные при ошибке найденной в ПО, или с неправильными калибровками. Данные класса RAW - не могут быть признаны устаревшими;

Следующим этапом стала разработка концепции динамического увеличения количества копий датасетов в зависимости от “популярности” данных. Был реализован механизм автоматического контроля количества копий при изменении количества запросов. Количество запросов анализировалось ежедневно, количество копий рассчитывалось каждые три дня. Оба интервала были выбраны, исходя из оценки среднего времени ожидания заданий анализа данных и времени их выполнения. Эта метод получил название “Динамическое распределение данных”. В качестве места для новой копии выбирался сайт со свободным дисковым пространством и вычислительным ресурсом из числа “центров хранения данных”. Такой подход также привел к конкуренции между грид центрами WLCG, т.к. теперь уровень загрузки был напрямую связан со стабильной работой центра.

На рисунке 18 показано, как введение термодинамической модели позволило оптимально использовать дисковый ресурс и увеличить скорость набора данных в четыре раза, до 400 Гц, а также и провести переобработку всех данных. Это было достигнуто исключительно за счет введение понятия “популярность данных” и оптимального использования дискового пространства грид центров.

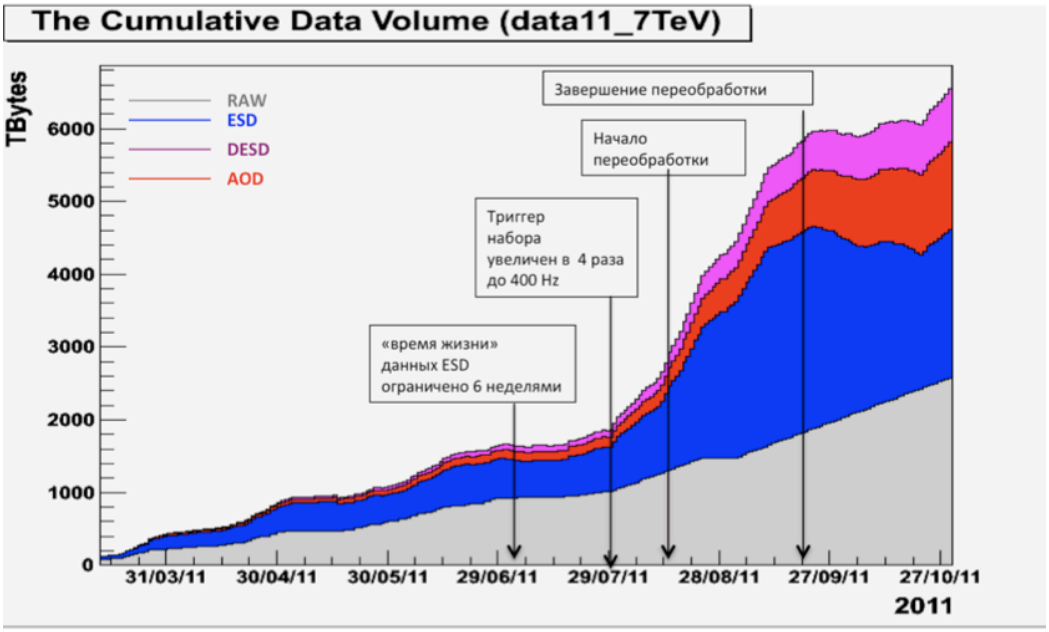


Рисунок 18 - Использование дискового ресурса ATLAS в 2011 году до и после введения термодинамической модели

1.4.3 Методика определения стабильности работы центров WLCG при создании «смешанной модели» грид инфраструктуры. Переход к «смешанной компьютерной модели» для экспериментов на Большом адронном коллайдере.

Одним из основных ограничений модели MONARC явилось предопределение функций центров грид на основе их классификации по уровню T0/T1/T2/T3. Вместе с тем оказалось, что многие центры уровня T2 имеют бОльшие мощности и часть центров по стабильности работы сравнима (или даже превосходят) центры уровня T1. Для перехода к «смешанной модели» необходимо было выделить наиболее мощные и стабильные центры и использовать их ресурс оптимальным образом. Были введены три независимые метрики:

- стабильность центра при обработке и анализе данных;
- стабильность центра при обмене данными с другими центрами;
- пропускная способность центра;

и четыре градации для классификации центров: альфа, бета, чарли, дельта (от английского A, B, C, D). На первом этапе все центры уровня T2 были причислены к группе ‘альфа’.

Стабильность центров при выполнении заданий обработки, анализа и моделирования. Были определены тестовые образцы заданий трех типов (это было сделано совместно с физическими группами, чтобы задания соответствовали реальным заданиям и имели с аналогичные требования по времени обработки событий, оперативной памяти и вводу/выводу информации) :

Для каждого из типов заданий были подготовлены тестовые наборы данных и их копии помещены во все центры уровня T2, для всех типов заданий были определены необходимые версии ПО (например, версия программы реконструкции событий). Версии ПО были доступны в каждом из центров. Ежедневно каждый центр получал запрос на выполнение тестовых заданий, при этом тип задания выбирался случайным образом. Результаты выполнения тестовых заданий хранились в базе данных, и статистика была доступна как компьютерным специалистам центров, так и представителям экспериментов LHC.

Стабильность центра при обмене данными с другими центрами. Были подготовлены тестовые наборы данных, эти наборы были распределены между всеми центрами уровня T2. Наборы отличались размерами файлов, который соответствовал размерам файлов, создаваемых при обработке, анализе и моделирование (в первом приближении, их можно разделить как файлы «большого размера» 2.5ГБ, «среднего» 1 ГБ и «малого» 0.5 ГБ). Четыре раза в день генерировался автоматический запрос на передачу данных между всеми центрами уровня T2, а также между всеми центрами уровня T2 к центрам уровня T1. По результатам тестовых передач создавалась матрица «все против всех» и по ней определялись центры уровня T2, имеющие стабильную передачу данных со всеми центрами уровня T1, и при наличии 100% выполнения заданий, описанных в предыдущем параграфе, эти центры используются также как центры уровня T1, за

исключением архивирования данных (такие центры получили название T2D) . Созданная матрица также используется системой управления загрузкой, при определении наилучшей комбинации центров для обработки данных (это будет рассмотрено детально в третьей главе диссертации в разделе «всемирное облако»).

Пропускная способность центра и стабильность его подключения к WAN. Как было изложено ранее модель MONARC была статической, и в качестве вычислительного ресурса в ней учитывались только вычислительные мощности центров WLCG и их дисковый ресурс. При переходе к «смешанной модели» необходимо было учитывать также ресурс WAN. Для этого инфраструктура каждого центра была дополнена системой мониторирования perfSONAR, которая позволяла постоянно мониторировать состояние WAN при передаче данных. Для всех пар сайтов (пункт отправки / пункт назначения) собирается следующая информация:

- количество файлов, переданных за последний час;
- количество файлов в очереди на передачу;
- средняя пропускная способность согласно метрикам FTS (File Transfer System) для последнего часа, дня и недели;
- информация от системы perfSONAR: задержки при передаче, число потерянных пакетов, пропускная способность.

Эта информация собирается специальным сервисом NWS (Network Weather Service) и хранится в специальном хранилище, агрегированная информация хранится в информационной системе. ИС запрашивает информацию от NWS каждые 15 минут. В третьей главе будет рассмотрено как информация о WAN используется для выбора вычислительного ресурса при выполнении заданий обработки, анализа или моделирования данных. На рисунке 19 представлена матрица, показывающая стабильность работы центров первого уровня при передаче данных между собой, а на рисунке 20 результаты передачи данных в центры, отобранные в результате созданной методики для “постоянного” хранения данных.

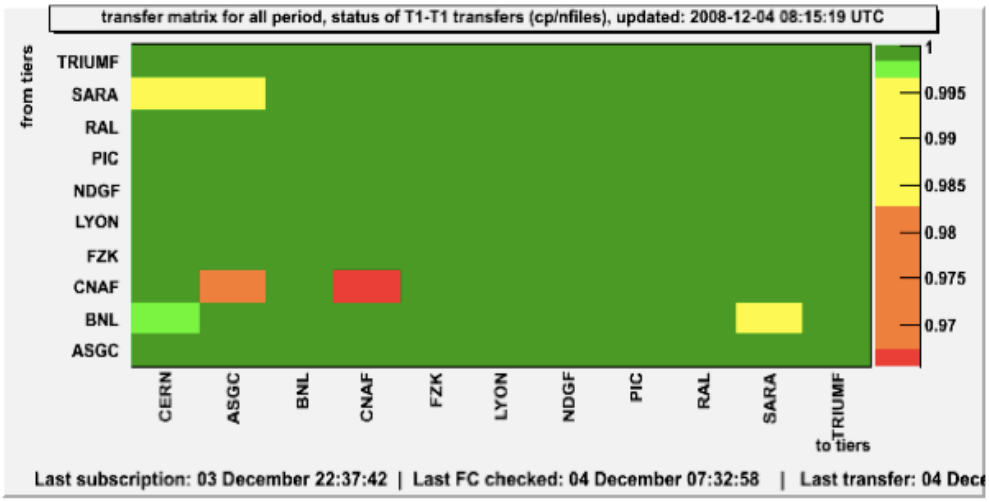


Рисунок 19 - Матрица эффективности работы центров T1 при передаче данных

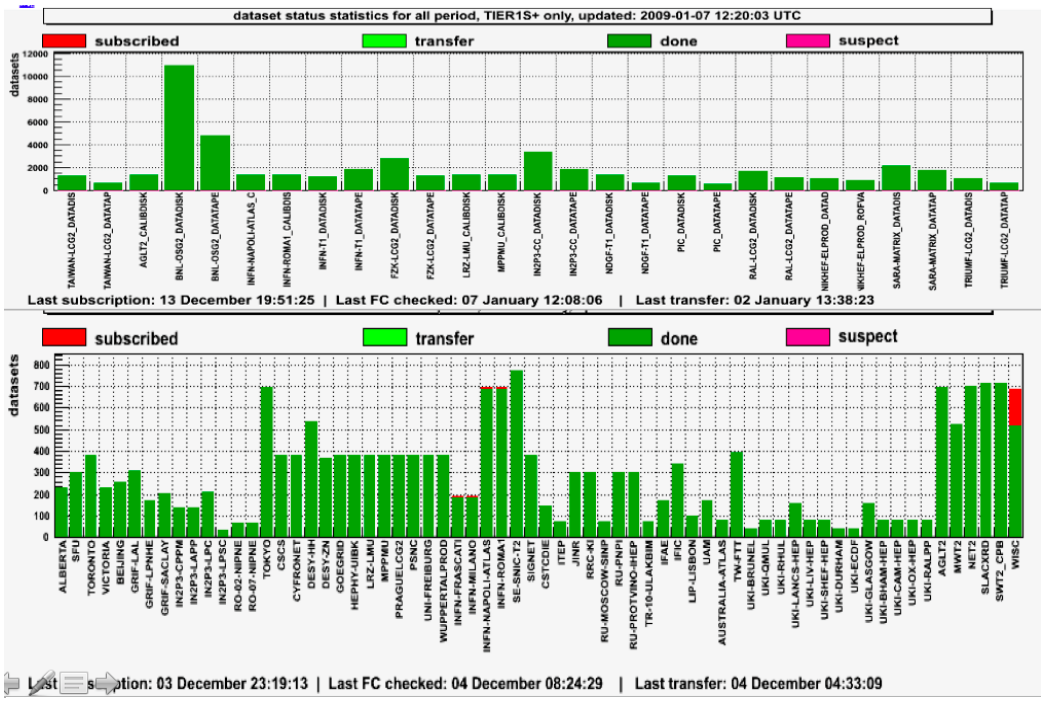


Рисунок 20- Результаты передачи данных в центры, отобранные в результате созданной методики для “постоянного” хранения данных

DDM Sonar						perfSONAR						
AvgBRS (MB/s)	EvS	AvgBRM (MB/s)	EvM	AvgBRL (MB/s)	EvL	MinThr (MB/s)	AvgThr (MB/s)	MaxThr (MB/s)	MinPL	AvgPL	MaxPL	FAX xrdcp rate
1.05+/-0.19	10	7.46+/-1.48	11	12.54+/-6.72	519	12.4	34.7	56.9	0.0	0.0	2.0	n/a
0.85+/-0.04	10	9.97+/-4.20	602	26.48+/-13.48	10	0.6	0.8	1.1	0.0	0.0	1.0	3.93
0.42+/-0.06	10	0.89+/-0.11	10	0.00+/-0.00	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
0.39+/-0.06	10	1.02+/-0.04	10	0.00+/-0.00	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
0.58+/-0.07	10	2.91+/-0.82	10	0.00+/-0.00	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
0.48+/-0.06	10	2.45+/-0.65	10	3.18+/-0.79	10	n/a	n/a	n/a	n/a	n/a	n/a	n/a
0.12+/-0.39	465	4.13+/-1.44	1575	4.59+/-1.68	3803	164.2	172.3	180.3	0.0	0.0	0.0	n/a
2.10+/-1.88	4920	8.76+/-6.32	10075	14.05+/-23.55	4006	0.3	0.3	0.3	0.0	0.0	0.0	0.72
0.47+/-0.11	5	1.23+/-0.39	9	0.00+/-0.00	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
0.37+/-0.11	10	1.14+/-0.20	5	2.53+/-0.15	10	n/a	n/a	n/a	n/a	n/a	n/a	n/a
0.67+/-0.54	10	7.53+/-3.81	10	0.00+/-0.00	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
0.56+/-0.38	10	5.95+/-2.64	10	50.52+/-9.11	10	n/a	n/a	n/a	n/a	n/a	n/a	n/a
0.94+/-0.08	10	5.41+/-1.33	10	0.00+/-0.00	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
0.55+/-0.25	10	4.95+/-1.63	10	21.09+/-9.01	10	n/a	n/a	n/a	n/a	n/a	n/a	n/a
1.13+/-0.11	10	7.17+/-1.44	510	0.00+/-0.00	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a
0.82+/-0.33	10	6.90+/-1.82	10	30.36+/-11.35	10	n/a	n/a	n/a	n/a	n/a	n/a	5.55
1.14+/-0.09	10	6.50+/-2.41	10	0.00+/-0.00	0	n/a	n/a	n/a	n/a	n/a	n/a	n/a

Рисунок 21 - Результаты тестирования глобальной сети между центрами грид для двух типов тестов : perfsonar и тестов передачи данных ATLAS

На рисунке 21 показаны результаты тестирования глобальной сети между центрами грид для двух типов тестов perfsonar и тестов передачи данных ATLAS. На рисунке 22 показано как изменилось количество центров второго уровня, используемых для хранения и обработки данных, на тех же “правах”, что и центры первого уровня после применения методики определения стабильности центров. Как отмечалось ранее, для большинства центров T2 - это дало шанс существенно расширить свои функциональные возможности и предоставить гораздо большие ресурсы для “виртуальной организации”. Многие центры (например, ОИЯИ, ИФВЭ НИЦ КИ) предоставляют свои ресурсы более, чем одной коллаборации, и изменение режима их использования экспериментов ATLAS привело к изменению статуса центров в экспериментах ALICE и CMS. Кроме того, введения параметра WAN, привело к необходимости для центров учитывать пропускную способность глобальной вычислительной сети и, в конечно итоге, созданию WAN известной ныне, как

LHCONE. На рисунке 22 показано как изменилось количество центров, используемых для передачи данных “все-ко-всем” (центры T2D) и как увеличилось число “стабильных центров” за 6 месяцев после использования методики определения стабильности центров.

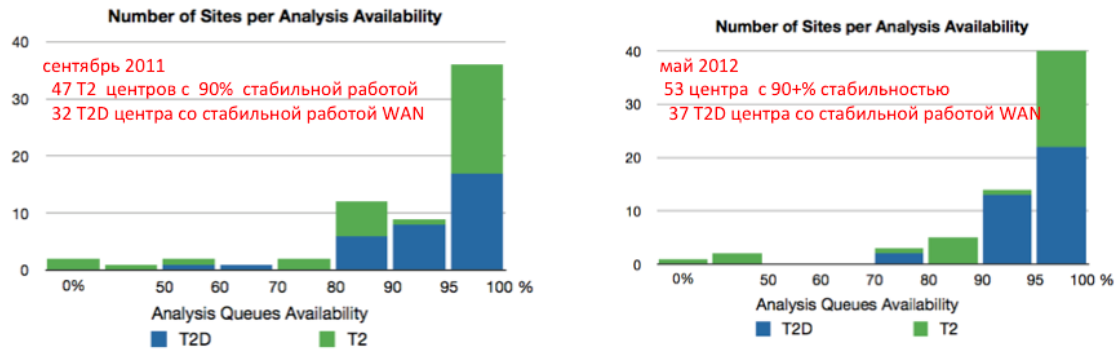


Рисунок 22 - Количество центров уровня T2, используемых для хранения данных

Следует отметить, что переход к «смешанной модели» потребовал разработки концепции и создания новой информационной системы хранения информации [63]. Концепция такой системы и ее архитектура были предложены автором диссертации и реализована совместно с А.В.Анисенковым изначально для эксперимента ATLAS, система получила названия AGIS – ATLAS Grid Information System, и позволила аккумулировать информацию о центрах WLCG, а также дополнить ее информацией собираемой в результате проверки стабильности работы центров.

Разработанная концепция ИС оказалась гибкой и динамичной, что позволило на следующем этапе создания модели компьютинга дополнить ее информацией о суперкомпьютерных центрах и центрах облачных вычислений. Дальнейшее развитие ИС и ее использование в экспериментах COMPASS и CMS, привело к разработке второго поколения ИС, известной как CRIC – Computing Resource Information Catalog.

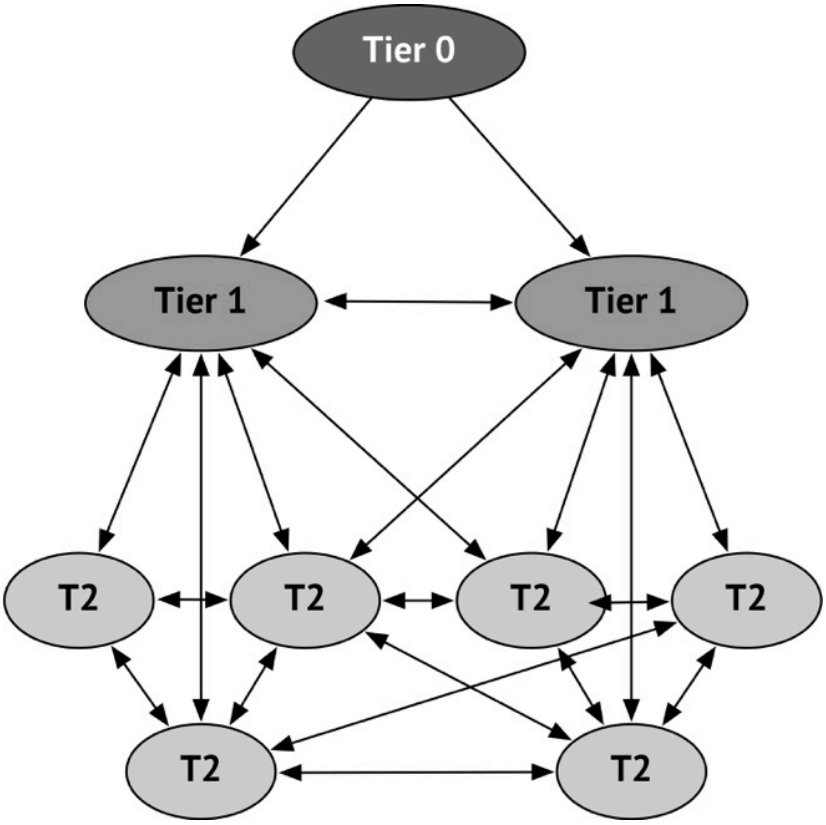


Рисунок 23 - “Смешанная компьютерная модель» для экспериментов LHC

Таким образом на первом этапе развития компьютерной модели был совершен переход от иерархической модели к «смешанной компьютерной модели» (рисунок 23), была «нарушена» иерархия и предопределение функций центров грид, предложенная в модели MONARC. Были созданы необходимые программные средства (система запросов на передачу данных, первая версия принципиально новой информационной системы), исследована работа и роль WAN при создании вычислительной инфраструктуры, разработана и реализована методика определения стабильной работы центров WLCG, разработана и реализована методика определения популярности данных и метод динамического распределения данных между центрами WLCG. Все это создало предпосылки к разработке новой компьютерной модели и созданию гетерогенной компьютерной киберинфраструктуры для следующего этапа работы LHC и нового поколения экспериментов в области физики частиц.

1.4.4 Метод динамического распределения данных с использованием информации о популярности данных

Слоган, описывающий модель MONARC звучал как: “задачи идут к данным” (т.е. задачи анализа, обработки, моделирования выполняются в центрах, где находятся исходные данные). Такой подход приводил не только к задержки с выполнением задач, но и искусственному увеличению количества копий данных. На рисунке 24 показано как после начала работы коллайдера резко возросло количество данных на сайтах T2 за счет многочисленных копий.

Подробно ограничения модели MONARC рассмотрены в разделе 1.3, здесь мы рассмотрим, как переход к “смешанной компьютерной модели”, введение термодинамической модели и методики выбора наиболее стабильных центров уровня T2 позволили более эффективно использовать ресурсы WLCG. На первом этапе были определены наиболее популярные форматы данных для задач физического анализа, ими оказались данные форматов AOD и NTUP, задачей второго этапа стала разработка метода динамического увеличения копий наиболее популярных наборов данных, и использования дискового пространства центров второго уровня для кэширования дополнительных копий.

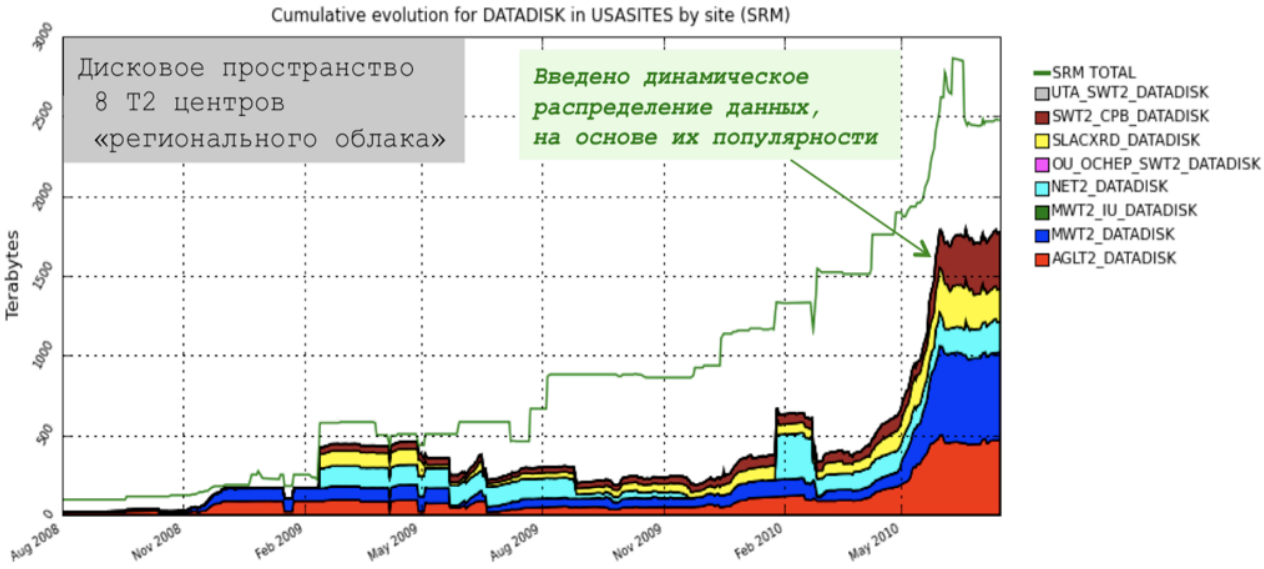


Рисунок 24 - Рост объема данных после начала работы LHC

Рассмотрим более подробно предложенный алгоритм:

- статическое распределение данных для центров Т2 прекращается;
- если задача пользователя обращается к набору данных (датасет) и нет копии датасет в центрах уровня Т2, то первая такая задача выполняется в центре первого уровня (Т1), где всегда есть копия данных, одновременно автоматически посылается запрос в систему управления данными для создания дополнительной копии набора данных на одном из центров уровня Т2;
- таким образом для первого обращения к данным не существовало задержки с выполнением задачи пользователя, а время передачи данных и создание дополнительной копии датасет занимало несколько часов.
    - метод выбора Т2 была основана на основе нескольких метрик (методика управления потоками заданий и работы брокера задач подробно рассмотрены в главе 3) :
        - учитывалось свободное дисковое пространство;
        - количество задач в очереди на выполнение к данному сайту;
        - планируемая время остановки сайта;
        - количество файлов в очереди на передачу к данному сайту;
            - общее количество копий данных основано на статистике обращения к ним задач пользователей и увеличивается логарифмически по мере роста количества обращений, т.е 10, 100, 1000,... обращений соответствуют 1,2,3,... дополнительным копиям данных;
- данные форматов, не используемых для физического анализа (за исключение формата EVNT - входные события для этапа моделирования), например, датасеты формата RAW, исключается из рассмотрения;

На рисунке 25 показано насколько эффективно используются наборы данных и их дополнительные копии.

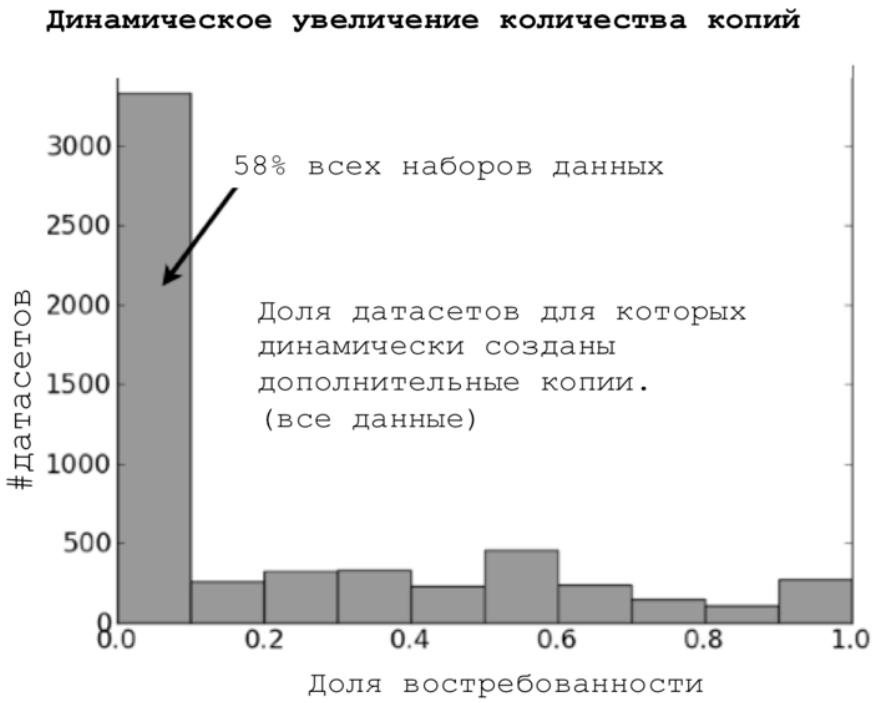


Рисунок 25 - Популярность данных в эксперименте ATLAS, частота использования наборов данных

Из графика видно, что в 58% случаев было достаточно только одной основной копии. Доля востребованности набора данных вычислялась как:

$$accessFraction = dynamicReplicaAccess/totalDatasetAccess$$

где :

- *dynamicReplicaAccess* - количество задач анализа обратившихся к дополнительному набору данных;
- *totalDatasetAccess* - общее количество обращений задач анализа к наборы данных;

На рисунке 26 показано какие классы данных наиболее популярны для динамического увеличение количества копий, а также демонстрация популярности копий, повторное (и более) использование копий. Графики подтверждают правильность выбранной методики и из них видна популярность данных форматов NTUP и AOD, используемых для физического анализа.

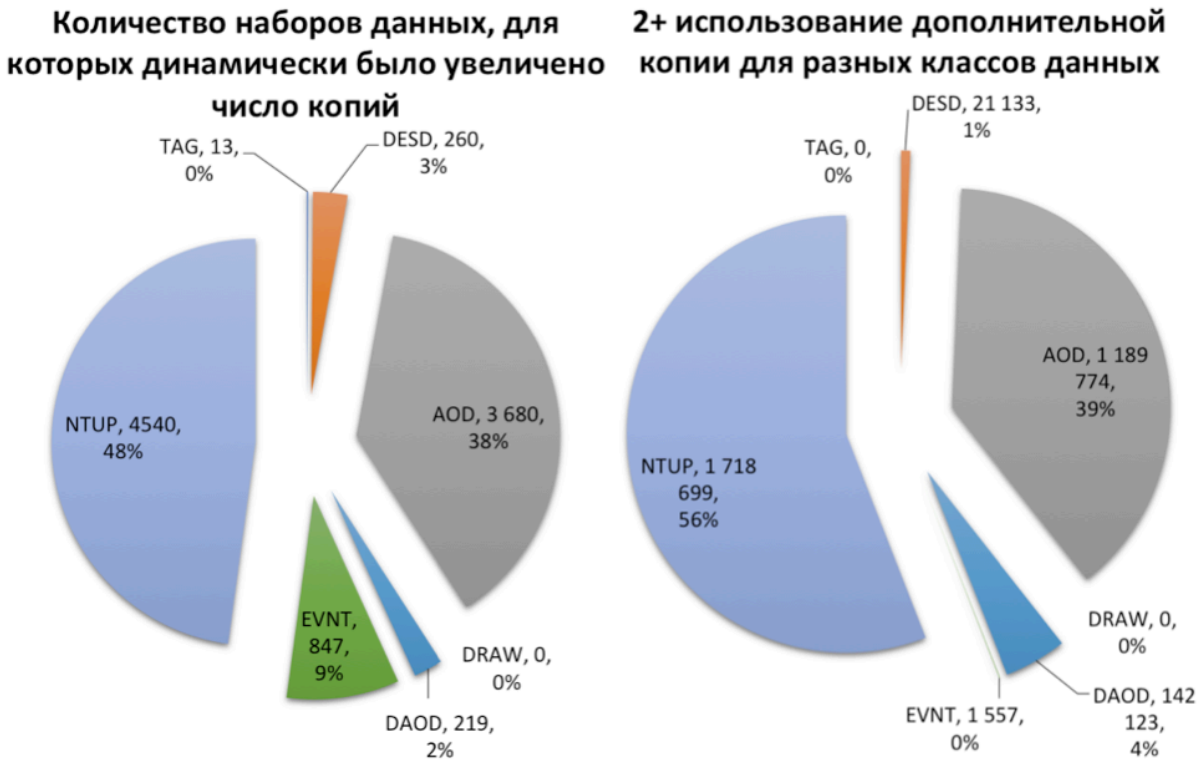


Рисунок 26 - Выбор данных для динамического увеличения наборов, популярность второй и последующих копий наборов данных в зависимости от класса данных

Важной особенностью явилось введение понятия “кэширование данных”, а также реализация термодинамической модели и понимание, что копии данных на дисках должны иметь “время жизни”, и по истечении интереса к набору данных, датасет должен автоматически архивироваться и “мигрировать” на ленту. На рисунке 27 показано как популярность данных менялась со временем, из графика видно, что “популярность” данных резко уменьшается через примерно 45 дней. Это привело к созданию специального сервиса в системе управления данных (deletion service) [65] для удаления копий датасетов не востребованных в физическом анализе. Таким образом, был сделан важный шаг в развитии методики управления данными и переход от заранее планируемого распределения данных на сайтах грид, к динамическому распределению данных.

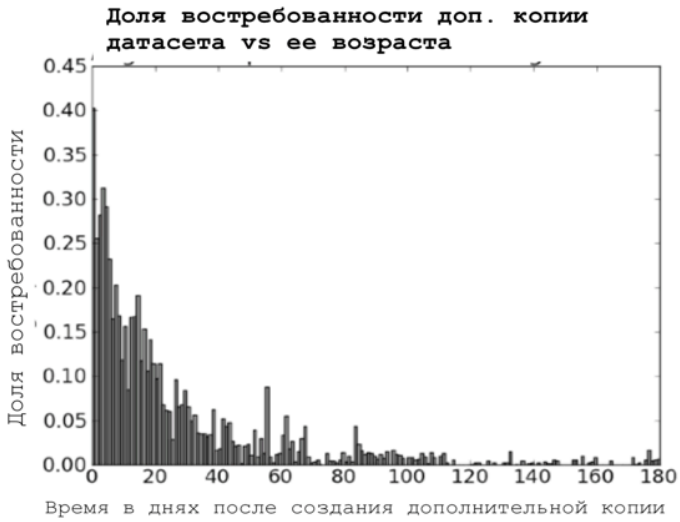


Рисунок 27 - Изменение популярности набора данных в зависимости от времени

Следует отметить, что у реализация данного метода не могла быть отложена до момента плановой остановки коллайдера, и его реализация проводилась непосредственно в период набора и обработки данных. Переход к динамической системе распределения данных позволил использовать слоган: “данные и задачи анализа идут туда, где есть свободные ресурс”.

Выводы к первой главе.

Разработана концепция и новая компьютерная модель для распределенной обработки данных. Обоснован переход от иерархической компьютерной модели обработки данных к “смешанной” компьютерной модели. Разработаны методика определения популярности классов научных данных и методика определения популярности наборов научных данных. На основе этих методик предложена и реализована метод динамического распределения данных между центрами грид инфраструктуры. Рассмотрены основные ограничения компьютерной модели на

первом этапе работы коллайдера LHC, фундаментальными ограничениями модели были :

- иерархия ВЦ и статический характер связки 1:T1-n:T2, когда любой сбой в работе центра первого уровня (T1) практически останавливал работу всех связанных с ним центров второго уровня (T2), в результате эксперименты лишались мощностей до 10 центров одновременно.
- недооценка роли глобальной вычислительной сети (WAN).

Обоснован вывод о необходимости учитывать ресурс WAN наряду с дисковым и компьютерным ресурсам. В главе обоснована возможность отказа от модели MONARC. Разработан метод оценки надежности и стабильности работы центров уровня T2, что позволяет использовать их ресурс для хранения данных, а сами центры для (пере)обработки данных.

Результаты исследований, изложенных в первой главе, подтверждают следующее защищаемое положение: Разработаны методы предсказания популярности классов данных и наборов данных, а также модель динамического управления данными в распределенной среде для сверхбольших объемов данных.

Глава 2. Требования к вычислительной инфраструктуре для обработки, моделирования и анализа данных. Роль суперкомпьютеров для приложений физики высоких энергий и ядерной физики

В данной главе рассмотрены требования к вычислительной инфраструктуре на втором и последующих этапах работы Большого адронного коллайдера. Обоснована необходимость развития компьютерной модели в области физики частиц в целом, и для экспериментов на LHC, в частности. Обоснован переход от однородной среды грид к использованию гетерогенной вычислительной среды, включающей в ресурсы облачных вычислений, суперкомпьютеры и грид-инфраструктуру. В главе рассмотрены вопросы использования суперкомпьютеров для приложений ФВЭ и ЯФ и интеграция суперкомпьютеров с системой высокопропускной обработки данных (грид).

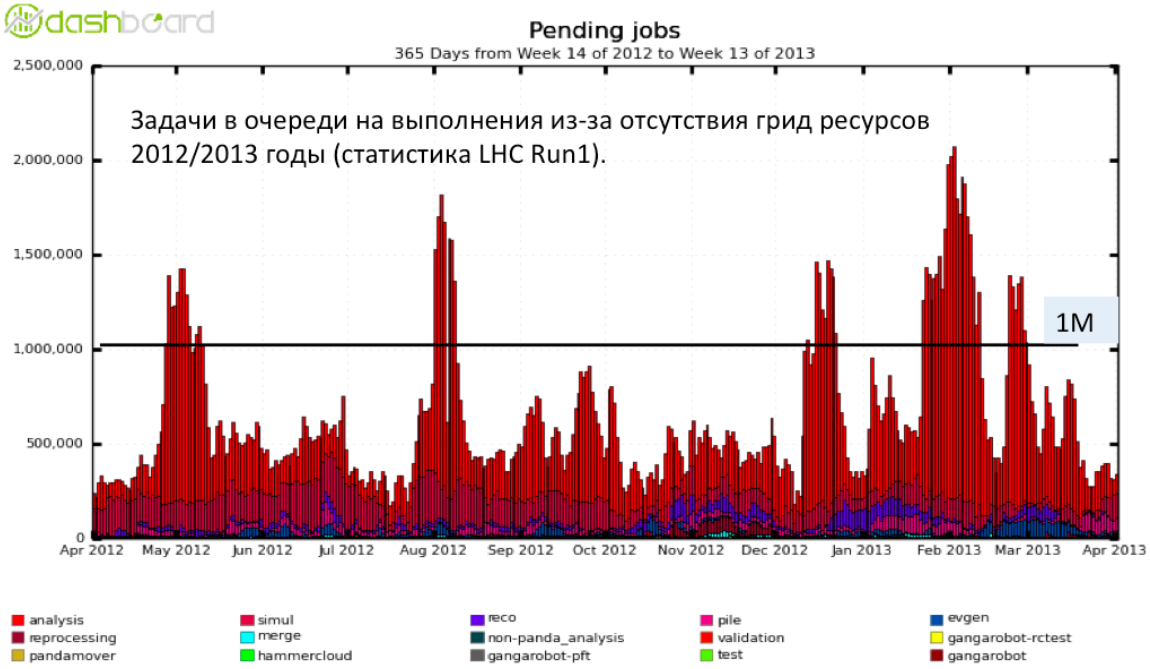


Рисунок 28 - Количество задач в очереди на выполнение из-за отсутствия вычислительного ресурса

К началу второго этапа работы LHC стало очевидно, что имеющийся компьютерный ресурс использован полностью. На рисунке 28 показано количество задач, ожидавших выполнения из-за отсутствия грид ресурсов. Хорошо видно, что в случае пиковых нагрузок, как правило предшествующих основным научным конференциям и этапам массовой переобработки данных, очередь могла достигать 1.5M задач. Беспрецедентная производительность LHC на втором этапе его работы и увеличение объемов данных требуют больших компьютерных мощностей, чем может предоставить консорциум WLCG (на рисунке 29 приведены графики интегральной светимости в 2014-2016 годах. Из рисунка видно, что в 2016 году светимость была на 60% больше запланированной, зеленая и серая кривые, соответственно). На графиках 30, 31 показаны зависимости времени обработки события и размера события в формате AOD для различных значений показателя “множественности” ускорителя.

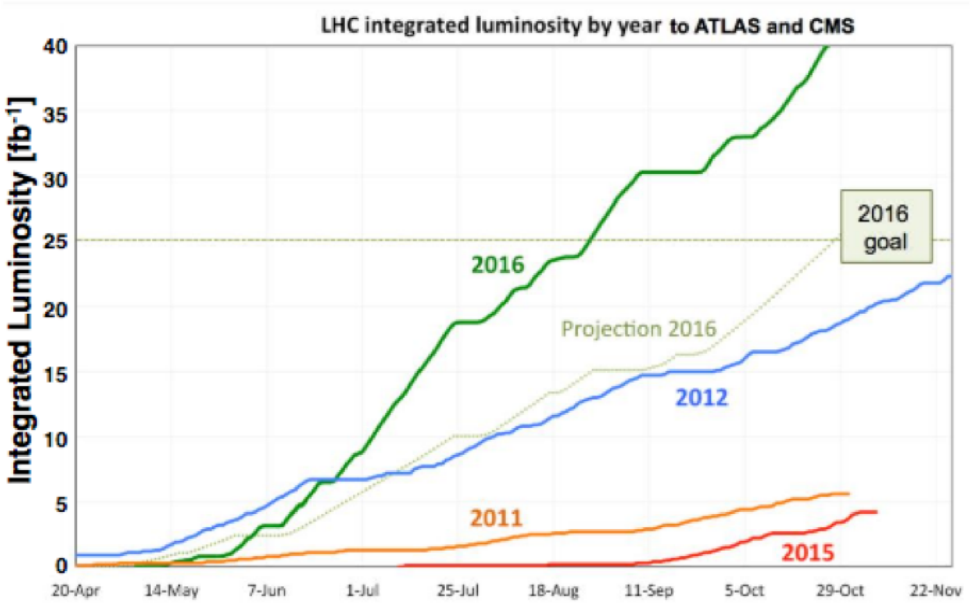


Рисунок 29 - Интегрированная светимость LHC в 2014, 2015, 2016 годах

Возрастающая “множественность”, связанная с ростом энергии пучков и светимости ускорителя, ведет к увеличению размера события и времени необходимого для его обработки. Ожидаемое количество данных для третьего и

четвертого этапа работы LHC (этап суперLHC или этап работы с высокой светимостью) показано на рисунке 32. Все эти факторы потребовали поиска дополнительных вычислительных ресурсов и пересмотра концепции использования только гомогенной вычислительной среды (грид).

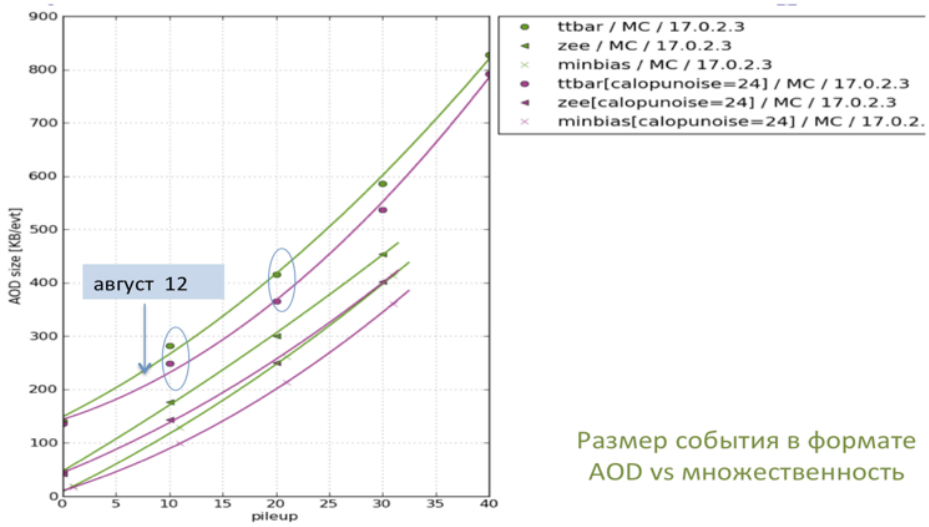


Рисунок 30 - Время реконструкции события в зависимости от показателя множественности

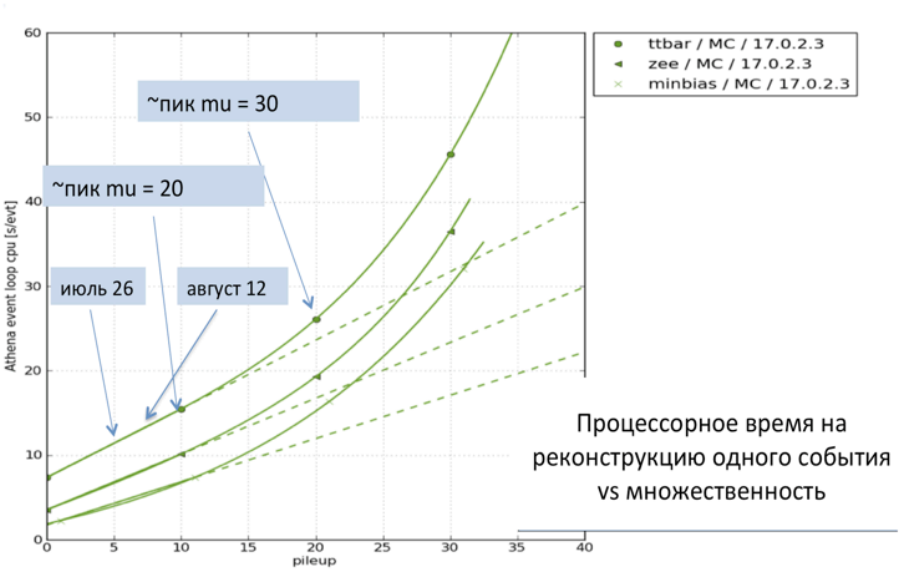


Рисунок 31 - Размер события в формате AOD в зависимости от показателя множественности

В силу финансовых и технических причин было невозможно увеличить вычислительный и дисковый ресурс центров грид в несколько раз или создать новые центры. Финансирующие организации согласились в лучшем случае сохранить существующий фонд финансирование ИТ для экспериментов на LHC. Технически, создание новых центров и их интеграция в общую инфраструктуру, потребовали бы годы. Кроме того, использование гомогенной киберинфраструктуры не выглядело более как единственное правильное решение.

Конечно, оставалась возможность консервации существовавшей компьютерной модели (только грид инфраструктура) в пределах доступного финансирования. Ценой такого решения стало бы уменьшение количества набираемых данных и невозможность обрабатывать данные в течение года, и как результат замедление в получении фундаментальных знаний об окружающем нас мире. Кроме того, появился ресурс, которого не было в начале века, а именно ресурс, предоставляемый крупными коммерческими ИТ компаниями (Google, Amazon, Яндекс). Рассмотрим, как эти факторы повлияли на эволюцию компьютерной модели в области физики частиц.

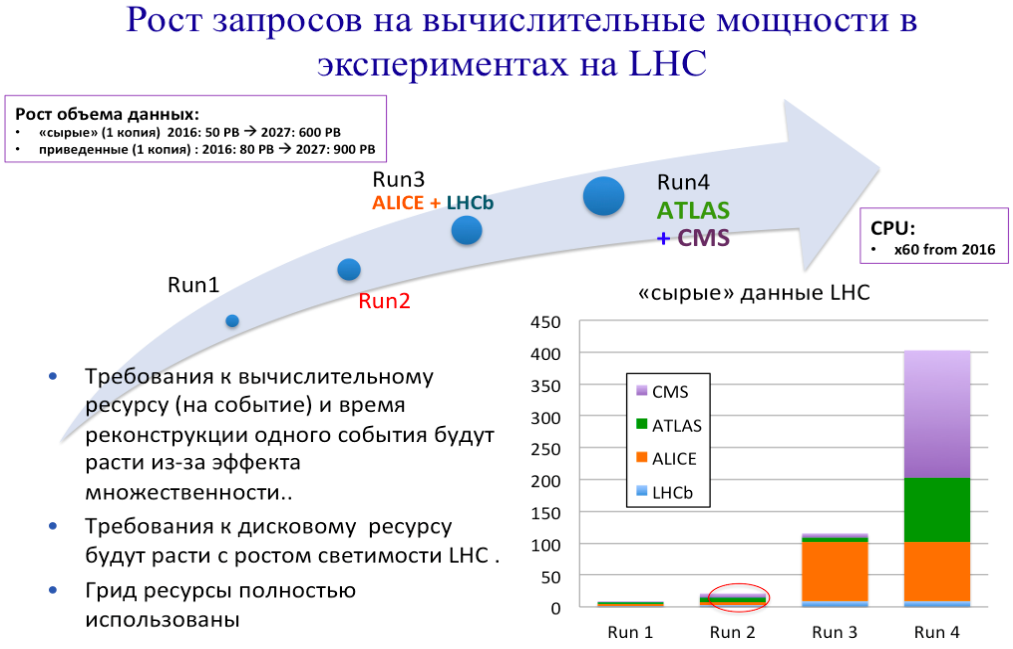


Рисунок 32 - Рост запросов на вычислительные мощности экспериментов LHC и ожидаемый объем данных для этапа высокой светимости LHC

Первый руководитель и организатор проекта WLCG Др. Лес Робертсон в своем докладе на конференции по компьютингу в области физики высоких энергий и ядерной физики в Праге [66] задал не риторический вопрос: «Нами построен грид. Что дальше?» Он так же постулировал, что вызовы, стоящие перед научным сообществом, требуют изменений в концепции обработки и анализа данных. В частности, Др. Робертсон сказал, что пока мы строили грид, компьютерный ландшафт изменился и надо быть готовым адаптироваться к новым реалиям. В некотором смысле, это выступление стало его напутствием, т.к. в следующем году он покинул пост руководителя WLCG.

Работы по развитию новой концепции и методов обработки данных были начаты в ЦЕРН, университетах и национальных лабораториях США (BNL, ФермиЛаб, Университеты Ратгерс и Техаса), Европе (университет Осло, DESY), России (ОИЯИ). Одним из лидеров таких исследований стала Лаборатория «Технологии Больших Данных для экспериментов в области мегасайенс» НИЦ «Курчатовский институт». Действительно, с момента начала проектов EGEE и globus ландшафт компьютерных ресурсов к началу второго десятилетия XXI века изменился. Он стал огромен и разнообразен, сложен и разнороден, способен к решению многих задач, и скорее выглядит как большой архипелаг, чем группа материков. Наряду с центрами коллективного пользования, которые подразумевают использование ресурса группами ученых, часто работающих в разных областях наук, узковедомственными и/или узкоспециализированными ВЦ, существуют суперкомпьютеры, грид консорциумы, коммерческие и академические ресурсы облачных вычислений, университетские вычислительные кластеры. Такова тенденция организации вычислительных мощностей (киберинфраструктуры) во всем мире. В настоящее время распределенная киберинфраструктура и ее составляющие используются в лучшем случае индивидуально, а чаще как изолированные ресурсы. Аристотель утверждал, что «целое больше, чем сумма его частей» [67], поэтому федеративная организация распределенной киберинфраструктуры позволит

использовать компьютерные ресурсы более эффективно, что будет выгодно как «владельцам» ресурса, так и пользователям. Проблема интеграции разнородных ресурсов и создание единой федеративной киберинфраструктуры стала основополагающей идеей при разработке новой компьютерной модели для экспериментов в области ФВЭ и ЯФ.

2.1 Общие проблемы создания федеративной киберинфраструктуры

Существующие проблемы создания федеративной распределенной киберинфраструктуры (ФРКИ) характеризуются:

- недостатком блоков, из которых можно построить такую федерацию;
- точечными решениями, не выходящими за пределы немедленного использования и конкретной задачи (и/или центра), другими словами, это, как правило, прикладные решения с низким уровнем абстракции для интерфейсов и модулей программного обеспечения;
- вопросы интеграции распределенной киберинфраструктуры рассматриваются только после создания вычислительной инфраструктуры, а не на этапе разработки архитектуры системы;
- использованием характеристик дискового и вычислительного ресурса как параметров, определяющих мощность вычислительного комплекса, без учета мощности глобальных компьютерных сетей, скорости передачи данных и возможного использования удаленного доступа к ним. (например, при создании грид инфраструктуры и программного обеспечения для экспериментов на Большом адронном коллайдере, не была предусмотрена возможность использовать ресурс Университетских кластеров, суперкомпьютерных центров и центров облачных вычислений, а также удаленный доступ к данным).

Таким образом, требуется решить все четыре проблемы и обеспечить доступ к информации о параметрах ФРКИ пакетам управления данными и загрузкой системы в динамическом режиме.

При исследовании ФРКИ и создании архитектуры необходимо следовать следующим принципиальным подходам:

- единый метод и уровень абстракции управления ресурсами;
- общая система управления загрузкой и данными в гетерогенной компьютерной среде;
- интегрируемые и развиваемые средства для управления программным обеспечением ФРКИ.

Все вместе это составляет проблему фундаментального характера. Отсутствие ее адекватного решения в настоящее время приводит к экономическим и функциональным потерям. Логика развития компьютерной модели экспериментов на LHC, в частности для эксперимента ATLAS, привела к выводу о необходимости создания системы для обработки данных в гетерогенной среде, федеративного устройства вычислительных ресурсов и создания системы управления данными и загрузкой в распределенной вычислительной среде. И хотя начальная мотивация была связана с экспериментами на Большом адронном коллайдере, а также будущими мегапроектами, такими как NICA, FAIR и XFEL, но как количественные, так и качественные требования не являются специфическими для физики, а типичны

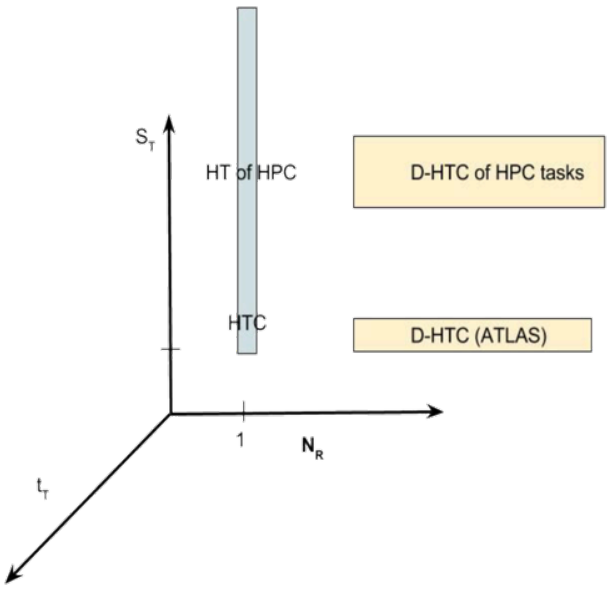


Рисунок 33 - Конвергенция (1) HPC и HTC ресурсов

для научных приложений в областях наук, требующих хранения, анализа, обработки и управления данными в мультипета- и эксабайтном диапазоне.

2.2 Вопросы конвергенции высокопропускных и высокопроизводительных вычислений. Роль приложений физики высоких энергий и ядерной физики для суперкомпьютеров

В данном разделе мы рассмотрим какое место занимает высокопропускной компьютинг (HTC - от англ. High Throughput Computing) и его использование. Научные задачи ФВЭ и ЯФ являются прекрасным примером HTC приложений.
Высокопропускные вычисления имеют следующие характеристики:

- рабочее (выполняемое) задание состоит из нескольких задач;
- каждое из заданий является частью одной компании (запроса, цепочки заданий, как будет показано далее в главе 3 о разработке системы управления потоками заданий). Типичным примером запроса является моделирование физических процессов методом Монте-Карло (рисунок 13), состоящее из цепочки последовательных заданий: генерация событий, отцифровка, моделирование, реконструкция, создание объектов, используемых для физического анализа.

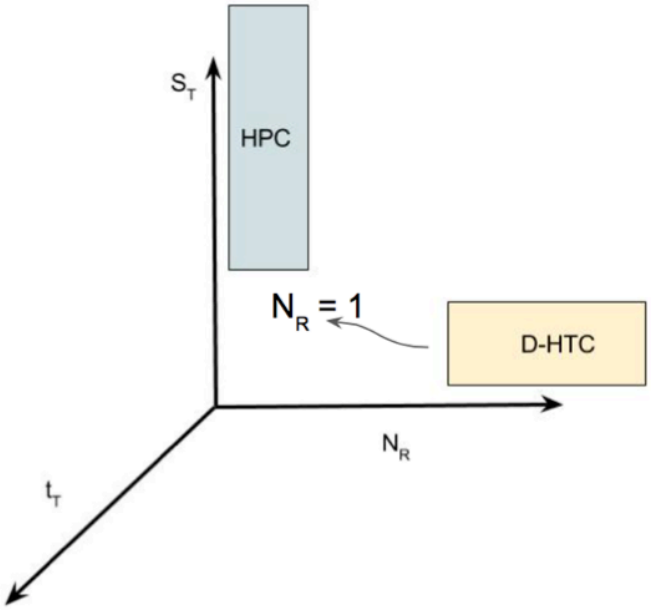


Рисунок 34 - Конвергенция (2) HPC и HTC ресурсов

- основной характеристикой является количество выполненных заданий;

Одновременно HTC может характеризоваться временным

параметром: время выполнения заданий, и иметь дополнительный параметр, назовем его параллелизм - это количество одновременно выполняемых задач в предоставленной вычислительной среде (например, количество задач виртуальной организации выполняемых ежедневно в среде WLCG). В HTC практически не используются параллельно выполняемые (и связанные между собой задачи). На рисунке 33 представлены параметры, которые требуют оценки:

Nt - число заданий

Nr - объем вычислительного ресурса

St - размер задания (количество узлов необходимых для его выполнения)

Tt - время выполнения задания

Nj - число задач, для выполнения заданное планировщиком заданий;

Для HTC традиционно это можно описать как : $N_T > 1$, $N_T \geq N_R$, $N_R = 1$, а с учетом распределенной модели обработки данных (D-HTC), как $S_T = 1$, $N_R > 1$ ($t_T < 12h$), а в применении HTC приложений для суперкомпьютеров (высокоскоростного компьютинга : HPC - High Performance Computing)

$S_T > 1$, $N_R > 1$.

Работы по использованию суперкомпьютеров для HTC приложений, например для приложений вычислительной химии с попыткой ввести параллелизм на уровне задач [68], для O(1000) задач, были не очень успешны, и привели к решению о выполнении задач последовательно на суперкомпьютере, но важным посланием является понимание важности модели выполнения заданий, и роль планировщика заданий, определяющего какие задания, где и когда выполняются. На рисунке 34 показано как может быть совершен переход от HTC к HPC.

Соответственно параметры, рассмотренные ранее $N_R > 1$, $S_T = 1 \rightarrow N_R = 1$ (HPC), $S_T > 1$, а модель обработки на СК меняется соответственно от D-HTC к HTC заданий HPC (HT-HPC).

Реализация такого подхода потребовало изменений и введения абстракции на нескольких уровнях модели обработки данных. Основными решениями явилось

понятие пилотных заданий (фундаментальное понятие для грид/НТС, поздняя “привязка” ресурсов к выполняемым заданиям) и создание нового поколения системы управления загрузкой (WMS - Workload Management System). Таким образом, существует четыре базовых уровня (рисунок 35):

- L4 - научные приложения
- L3 - система управления загрузкой
- L2 - система управления выполнением задач
- L1 - вычислительный ресурс

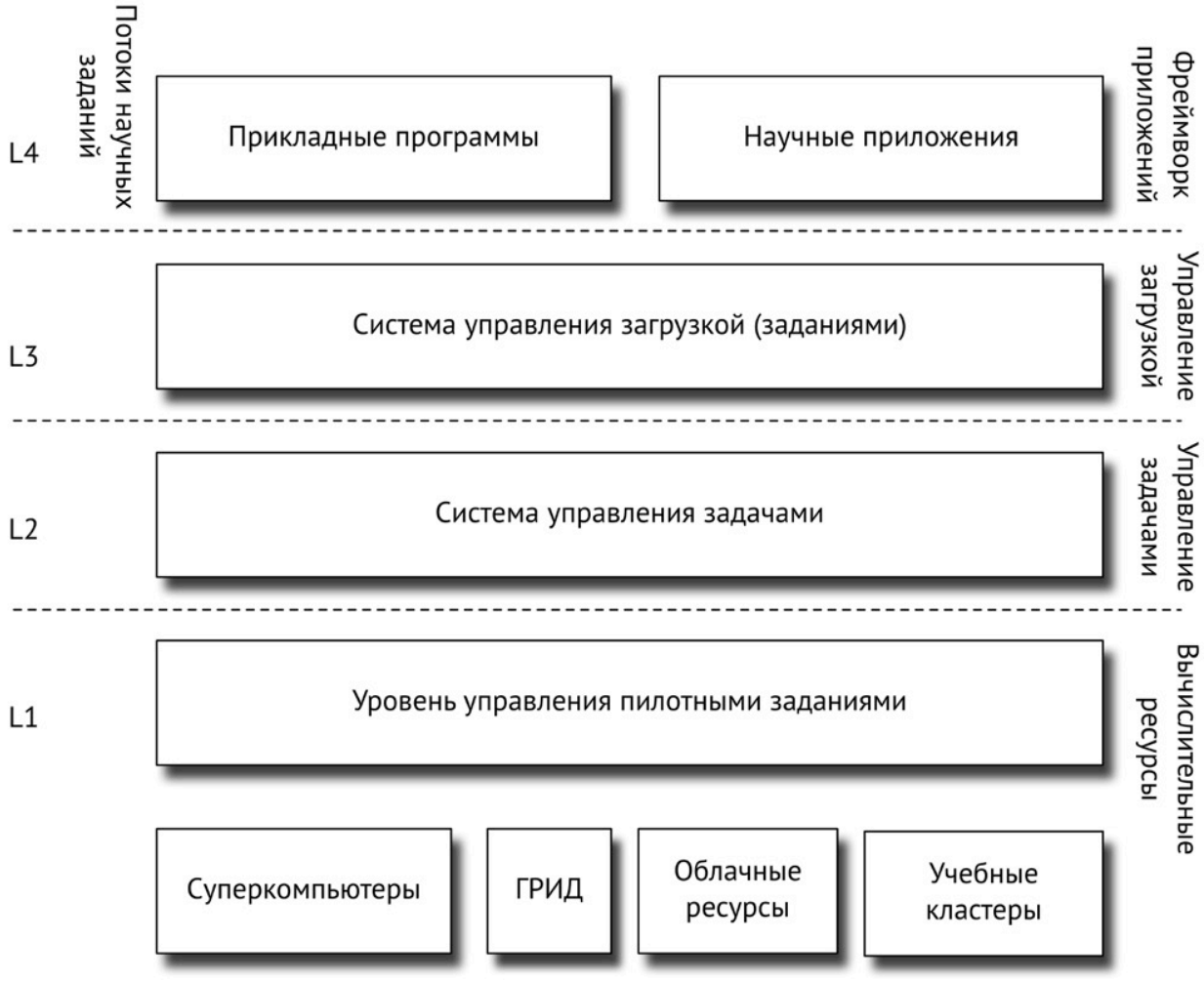


Рисунок 35 - Схема интеграции системы управления загрузкой и вычислительных мощностей

Для интеграции всех возможных вычислительных ресурсов и их оптимального использования, необходимо разработать систему управления выполнением задач (этот вопрос подробно рассмотрен в главе 3).

Рассмотрим типичное научное приложение физики высоких энергий на примере заданий обработки и анализа данных эксперимента ATLAS. С точки зрения потребностей в обработке и анализе данных, ATLAS является ярким примером почему необходима федеративная организация вычислительных мощностей и использование всего доступного вычислительного ресурса. Типичные вычислительные потребности эксперимента характеризуются следующими величинами: два миллиона выполненных задач, три миллиона используемых ЦПУ часов в сутки.

- наиболее интенсивное ATLAS-"задание" - это приложение для обработки данных, использующее до 2 Гбайт памяти на одном ядре в течение примерно 12 часов, обрабатывающее максимально нескольких гигабайт входных данных и производящее выходные данные такого же объема;
- непрерывная глобальная передача данных на уровне до 30 ГБ/сек суммарно.
- требуется шесть миллионов ядро-часов (core-hours) для первого шага обработки одного петабайта данных ATLAS, полученных от одного миллиарда актов столкновений на БАК.
- совокупный поток заданий ATLAS использует более 250 000 ЦПУ-ядер, с пиковой производительностью приблизительно 0.32 петафлопса (такую производительность обеспечивают суперкомпьютер №40 из списка Top500 [69])

Есть по крайней мере два дополнительных параметра, заслуживающих упоминания:

(a) загрузка мощностей не является статичной во времени, и зависит от многих причин. При хорошо определенном среднем значении в 1.5 миллиона задач в день, существуют сильные временные флуктуации, когда потребность

в ресурсах возрастает в 10 и более раз в течение короткого (дни) промежутка времени, что соответствует классическим примерам систем с ограничением в передаче данных и доступа к распределенному вычислительному ресурсу;

(б) стабильная во времени потребность в компьютерных мощностях (т.н. steady state demand).

Даже в отсутствии пиковых нагрузок ресурс предоставляемый консорциумом WLCG недостаточен. Пример ATLAS не единичен, другие эксперименты на LHC: ALICE, CMS и LHCb, сталкиваются со схожей проблемой.

Необходимо также отметить, что следующее поколение программного обеспечения экспериментов в области физики высоких энергий и ядерной физики, будут гораздо более комплексным, сложным и неоднородным, поэтому пост-Хиггс эра (исследование свойств новой частицы и возможно поиск второй и третьей частицы *a la* Хиггс) потребуют в будущем гораздо большего вычислительного ресурса, в дополнение к тому что количество данных, набираемых ежегодно, будет расти (как это было рассмотрено ранее).

Более того «загрузка» суперкомпьютеров приложениями ФВЭ и ЯФ может быть не только потребностью экспериментов, но и иметь сильный экономический аргумент для «владельцев» подобных машин. Хотя точная цифра загрузки суперкомпьютерных центров широко не афишируется, будет возможным предположить, что она не превышает 90% (автор использует информацию последних суперкомпьютерных конференций (SC14, SC16 2014, 2016 гг) и технические данные некоторых центров в России, США и Европе). Таким образом около 10% ресурса могут быть использованы в фоновом режиме (backfill), без изменения существующего портфеля задач, что повысит эффективность и процент использования суперкомпьютеров. Т.е. более гибкое и эффективное использование суперкомпьютерного ресурса возможно за счет приложений ФВЭ и ЯФ, когда такой

ресурс доступен в суперкомпьютерном центре и не используется для специальных приложений.

Важно понимать, что подобный подход представляет интерес для многих научных приложений (за пределами экспериментов на LHC) и для многих научных дисциплин.

2.3 Роль суперкомпьютеров для приложений физики высоких энергий и ядерной физики

Научные приоритеты в области физики высоких энергий и ядерной физики представляют собой проблемы управления и работы с “большими данными”, требующие современных вычислительных подходов и, следовательно, служат проводниками новых идей по созданию интегрированной компьютерной и информационной инфраструктуры. Для ФВЭ эти приоритеты включают исследования свойств бозона Хиггса в попытке лучше понять происхождение массы элементарных частиц и поиска новых законов физики. Для ЯФ эти приоритеты включают исследования свойств кварк-глюонной плазмы. Во избежание потенциальных проблем, связанных с недостатком в существующих и будущих ресурсов, предоставляемых консорциумом WLCG, эксперименты на LHC (также как будущие эксперименты на FAIR, NICA, LSST) начали рассматривать суперкомпьютеры, как возможный дополнительный вычислительный ресурс, который позволил бы не уменьшать скорость набора данных, а также вести моделирование физических процессов в необходимых объемах, как отмечалось ранее время выполнения задач моделирования методом Монте-Карло (рисунок 14 [70]) занимает до 42% всех вычислительных ресурсов. Первый этап работы LHC (2009-2013) и первые годы второй фазы работы LHC (Run2 2015-2018 гг.) убедили физиков, что их ПО нуждается в фундаментальном переосмыслении. Возможность использовать особенности суперкомпьютеров, такие, как параллельные вычисления

и графические процессоры, должно привести к созданию нового поколения ПО для физических экспериментов. Разработка ПО для выполнения на СК - должно стать задачей для ФВЭ и ЯФ, и эта задача должна быть выполнена до начала третьего этапа работы коллайдера.

После успеха в обнаружении новой частицы, ATLAS и CMS проводят более точные измерения и исследования свойств частицы, необходимые для дальнейших открытий, которые станут возможными при гораздо более высоких энергиях работы LHC. Одновременно потребность в моделировании и анализе будет превосходить ожидаемую мощность вычислительных мощностей WLCG, ценой отказа от использования СК будет сокращение диапазона и точности физических исследований. Даже сегодня важные задачи анализа физических данных, которые требуют больших наборов моделируемых событий откладываются на месяцы, т.к. существующие ресурсы WLCG полностью заняты. Кроме того, некоторые физические процессы, представляющие интерес для LHC, практически невозможно промоделировать на традиционных грид ресурсах из-за чрезвычайно высоких вычислительных требований. Суперкомпьютеры предлагают уникальную возможность промоделировать и создать такие наборы данных посредством массивного распараллеливания. Кроме того, сгенерированные события могут представлять интерес для физиков теоретиков по всему миру, и не являются ориентированы только на научные интересы специфические для LHC экспериментов. По мере роста энергии и светимости коллайдера отсутствие адекватного вычислительного ресурса приведет к еще большему отставанию выполнения физической программы экспериментов.

Поиск новых открытий в фундаментальной физике требуют сравнения между результатами экспериментов и предсказаниями новых теорий, наборами реальных событий и соответствующими наборами событий Монте-Карло для различных теоретических моделей. Кроме того, для получения статистических выводов требуется массовое моделирование всех известных физических процессов. Из-за

необходимости сравнения с моделированными событиями, физическая программа LHC экспериментов часто ограничена не их возможностями к набору данных и проведению измерений, а способностью проанализировать набранные данные, из-за отсутствия соответствующих моделируемых данных. Кроме того, существуют ограничения на общее количество событий, их сложность, а в некоторых случаях ограничения на полное моделирования теоретических моделей. Задачи моделирование в ФВЭ и ЯФ являются хорошими кандидатами для работы на СК в режиме фоновой загрузки. Моделирование эксперимента состоит из ряда последовательных шагов (рисунок 13), из которых шаг генерации событий (*'evgen')* является наиболее ЦПУ-затратным с минимальными требованиями к вводу/выводу, дальнейшее "разбиение" этого шага на десятки тысяч "шажков" по генерации отдельных событий, позволит выполнить код программ на многих возможных аппаратных архитектурах, и СК являются наиболее вероятными кандидатами из-за своей вычислительной мощности и минимальной коммуникационной нагрузки между задачами выполняемыми на рабочих узлах.

Полностью оптимизированное использование существующих и новых суперкомпьютерных мощностей для приложениями ФВЭ и ЯФ является долгосрочной задачей, требующей работы в течение нескольких лет.

Таким образом, приложения ФВЭ и ЯФ требуют не только больших объемов вычислений, но и возможностей, которые могут предоставить только суперкомпьютеры. Вклад суперкомпьютеров порядка 100 миллионов или более ЦПУ-часов в год становятся важным и ценным дополнением к имеющимся ресурсам WLCG. Следует отметить, что выполнение приложений ФВЭ и ЯФ может быть близким к идеальному «заполнению пустот», когда они использует свободный ресурс в дополнение к классическим приложениям, выполняемых на суперкомпьютерах, таким как, моделирование климата или теоретические расчеты в квантовой хромодинамике.

Интеграция суперкомпьютеров и ресурсов облачных вычислений потребовали разработки новой архитектуры системы управления потоками заданий, которая могла бы работать с динамически изменяющимся вычислительными ресурсами, и использовать мощности, которые доступны в течение относительно коротких периодов времени. Одновременно необходимо было расширить компьютерную модель и ввести понятие ВЦ без “дискового элемента”, потому что ни СК центры, ни центры облачных вычислений не предоставляют дискового ресурса для постоянного хранения данных (в данном случае в масштабах необходимых для экспериментов на LHC - сотни петабайт).

Выводы ко второй главе.

Выводы ко второй главе формулируются следующим образом. Обоснована значимость разработки новой архитектуры системы управления потоками заданий и программного обеспечения для ее реализации. Обоснованы принципы распределенной системы для обработки данных, которая могла бы работать с динамически изменяющимися вычислительными ресурсами и использовать мощности доступные в течение относительно коротких временных интервалов. Также показана необходимость расширения компьютерной модели и введения понятия ВЦ без “дискового элемента”, поскольку ни суперкомпьютерные центры, ни центры облачных вычислений не предоставляют дисковый ресурс для постоянного хранения данных (речь идет о дисковом ресурсе масштаба сотен Пбайт, необходимых для экспериментов на LHC). Определена роль научных приложений ФВЭ и ЯФ для суперкомпьютеров.

Изложенное подтверждает следующее защищаемое положение:

Разработанная компьютерная модель современного физического эксперимента позволяет использовать гетерогенные вычислительные мощности в рамках единой вычислительной инфраструктуры

Глава 3. Разработка концепции, методов и архитектуры системы управления потоками заданий в распределенной гетерогенной компьютерной среде

Эта глава посвящена разработке концепции, методов и архитектуры системы управления заданиями в гетерогенной компьютерной среде. В главе сформулированы требования к системе, проведен анализ классов научных приложений в ФВЭ и ЯФ, предложена логическая модель данных системы распределенной обработки. Рассмотрены вопросы разделения вычислительного ресурса между различными потоками заданий для обработки, моделирования и анализа данных. В этом же главе рассмотрены принципы построения системы для глобальной распределенной обработки данных, ее уровни, функции и взаимодействие компонент системы между собой, а также взаимодействие системы для обработки данных с внешними системами, такими как система управления данными и информационная система. На основе требований к архитектуре и анализа ее основных функций, предложена архитектура системы для обработки данных в распределенной среде, способная обрабатывать данные эксабайтного диапазона. Последний раздел главы посвящен реализации системы управления потоками заданий эксперимента ATLAS на основе разработанных принципов и методов. Подробно рассмотрены характеристики созданной системы. В этом разделе также обоснованы принципы подсистемы мониторинга и ее компоненты.

Как обсуждалось во второй главе, при создании федеративной распределенной киберинфраструктуры существуют функциональные трудности связанные с управлением загрузкой и описанием ресурсов (проблема описания вычислительных ресурсов была решена в созданной информационной системе AGIS), другая часть проблем связана с идентификацией пользователей, протоколам обмена информацией и определением политики использования и разделения ресурса. В последнее время

есть успешные попытки синхронизировать и гармонизировать вторую группу проблем, но в этом есть смысл, если существуют все узловые функциональные блоки для создания ФРКИ.

Для решения проблемы управления загрузкой в гетерогенной среде необходимо разработать систему управления потоком заданий нового поколения (WMS) и модели выполнения заданий для динамически определяемых федеративных гетерогенных ресурсов. Такая система должна работать независимо от типа инфраструктуры, ее неоднородности и с учетом параметров, определяющих динамику возможного изменения ресурса. В целом предлагаемая модель выполнения заданий и управления загрузкой имеет следующие основные особенности :

- интеграция информации о выполняемом задании и ресурсах;
- стратегия выполнения основывается на последовательности решений, используемой для выполнения данного задания, которая может измениться при условии изменения инфраструктуры и/или типа задания.
- такой подход позволяет интегрировать рабочую нагрузку и ресурсы: а) оценить потребности необходимые для выполнения задания (рабочей нагрузки), б) оценить возможности ресурса, выработать стратегию выполнения и начать выполнение задания. Схематично базовые уровни системы и их взаимодействие показаны на рисунке 36.

Таким образом новое поколение системы управления загрузкой должно было а) абстрагировать задание от управления ресурсами; б) иметь достаточно высокую гранулярность для выбора ресурса наилучшим образом соответствующего заданию; в) проводить жесткий контроль выполнения заданий; г) обеспечивать создание инфраструктуры управления заданиями без явного управления ресурсами.

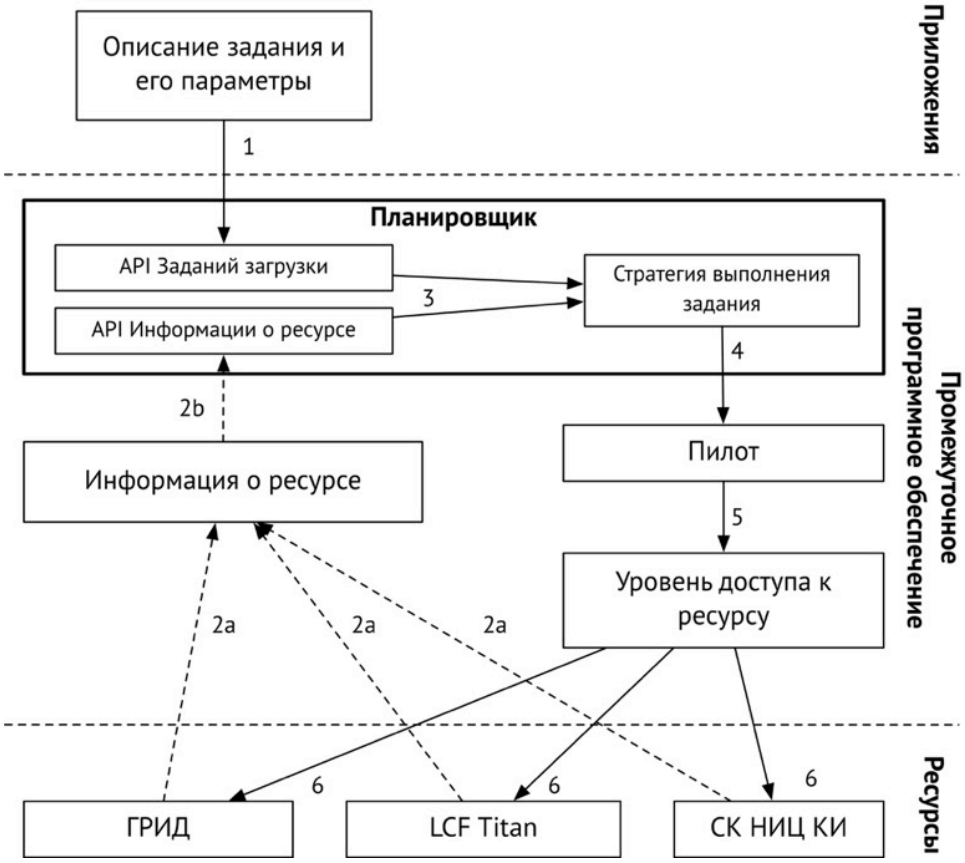


Рисунок 36 – Этапы выполнения задания и взаимодействия с гетерогенными вычислительными ресурсами

3.1 Классификация типов заданий современного эксперимента в области физики высоких энергий и ядерной физики

Моделирование методом Монте-Карло. Моделирование физических процессов и детекторов методом Монте-Карло является первым этапом при написании технического документа для любого эксперимента. Стандартная последовательность преобразований в задачах Монте-Карло физического эксперимента рассмотрена в работе [71] (опубликованной в соавторстве с коллегами из эксперимента ATLAS) и представляет из себя цепочку логически связанных заданий для следующие этапов их выполнения (шаги обработки данных):

- генерация событий (шаги evgen,evnt);
- моделирование (шаг simul);

- реконструкция (шаг recon);
- создание объектов для последующего анализа данных (шаг AOD).

Этапы генерации событий и моделирования требуют значительного вычислительного ресурса и отличаются от последующих этапов сравнительно невысокими требованиями по обмену информацией между памятью и диском на рабочем узле. На рисунке 14 показано время выполнения для различных типов заданий, из рисунка следует, что время, затраченное на моделирование и генерацию событий составляет 42% от общих вычислительных потребностей современного эксперимента. В запросах на выполнение заданий последовательность выполнения может начинаться с любого шага. Например, с шага реконструкции, либо выборочно могут быть использованы события, сгенерированные ранее. В рамках одного физического эксперимента или “виртуальной организации”, в терминах грид, все запросы на моделирование поступают и выполняются централизовано через одно “входное окно”, приоритеты определяются физической программой эксперимента. Сводный запрос составляется на основании запросов групп, занимающихся исследованиями по различным направлениям (стандартная модель, исследование свойств бозона Хиггса, суперсимметрия и т.д.). Преимуществом данного класса заданий является предсказуемость требуемого вычислительного ресурса и, как правило, времени выполнения задания. Данный класс потоков заданий выполняется централизованно для всего эксперимента.

Обработка и переобработка данных эксперимента. Обработка данных для системы отбора событий триггера “высшего” уровня (триггер HLT). Централизованная обработка данных эксперимента выполняется в два шага:

- реконструкция (шаг recon);
- создание объектов для последующего анализа данных (шаг AOD).

Как правило, физические эксперименты стараются провести первую обработку данных в течение 24-48 часов после окончания набора (при наличии калибровок и

вычислительных мощностей), т.е. это непрерывный поток заданий во время работы ускорителя. Переобработка данных проводится для уточненных калибровок (и/или с уточненной информацией о параметрах работы детектора), а также при изменении версии программного обеспечения (в среднем 1.5 переобработки для каждого года работы ускорителя). Переобработка данных проводится централизовано в течение 1-2 месяцев. Обработка данных для системы отбора событий “высшего уровня” (от англ. HLT - High Level Trigger), отличается только версией ПО, и требованием получения конечного результата в течение нескольких часов, что накладывает дополнительные требования на систему управления загрузкой. Во всех перечисленных случаях последовательность шагов обработки не отличается от “рутинной” реконструкции событий при шаге *recon*. Программы реконструкции, используемые для реальных и моделируемых событий одинаковы, особенностью для данного класса заданий, является разветвленная иерархия, когда создаются объекты промежуточных форматов (например, ESD - Event Summary Data), имеющие время жизни несколько месяцев, при этом результаты обработки одной задачи могут быть полностью или частично входными данными для нескольких новых задач. Класс заданий выполняется централизованно для всего эксперимента.

Обработка, фильтрация и анализ данных, проводимые физическими группами. Задания обработки для отдельных физических групп имеют ту же последовательность, что и при обработке данных эксперимента, но могут отличаться версиями программного обеспечения. Задания фильтрации отличаются высокими требованиями к вводу/выводу, и в результате работы создают выборку событий, используемую для физического анализа, проводимого группами и отдельными учеными. Последовательность задач анализа физических данных, чаще всего имеет простую структуру и состоит из одной задачи для каждого набора данных. Входные параметры задачи хранятся в файле текстового формата. Особенностью анализа, проводимого физическими группами, является большое количество пользователей, что требует продуманной системы аутентификации и авторизации при управлении

заданиями и доступа к данным. Класс заданий выполняется централизованно для каждой группы. Как правило, количество групп соответствует количеству научных тем, по которым ведутся исследования, и варьируется от 10 до 30, в зависимости от сложности эксперимента и его программы. (Обработка “поездом” описанная ниже позволила сократить количество вариаций и выполнять централизовано данный класс заданий). В результате этого типа обработки создаются данные в формате DAOD (от английского Derived Analysis Object Data) и данные в табличном формате NTUP (от английского ntuples)

Физический анализ данных. Наиболее важный и конечный шаг всего процесса получения физического результата. Подразумевается, что на этом этапе ученые используют наборы данных, созданные либо на шаге реконструкции, либо в результате выполнения заданий физических групп. В реальности эта группа заданий наиболее разнородна и непредсказуема, как по набору шагов, так и используемых версий ПО. Задания данной группы децентрализованы, запросы на анализ физических данных поступает в среднем от 1000 ученых ежемесячно, а количество заданий составляет до 50% от всех выполняемых системой заданий.

3.2 Модель данных

После проведения анализа классов заданий необходимо было определить базовые компоненты системы и построить логическую модель данных. Были определены следующие сущности:

Запрос (Request) - верхний уровень абстракции, объединяющий задания одного класса. Типичным примером может быть *запрос* коллаборации на (пере)обработку всех данных для определенного периода работы установки, или компания по моделированию детектора и физических процессов для этапа работы ускорителя с новыми параметрами (энергия, светимость, множественность событий). Таким запросом может также быть запрос физической группы на специфический

анализ данных, например, поиск редкого распада в одном из каналов. Каждый запрос имеет статус. Статус запроса отражает его текущее состояние – подготовлен, в процессе выполнения, выполнен. Изменения статуса запроса выделено в отдельную сущность для возможности хранения истории изменений состояния, а также для удобства мониторирования и учета работы (аккаунтинга).

Список входных параметров (input list) - включают в себя параметры для запуска задач генерации событий, входные наборы данных и атрибуты, относящиеся к ним, например, приоритет и комментарий.

Шаблон шага обработки данных (step template) - заранее определенный набор параметров (включая информацию о форматах выходных данных, версии ПО, ...), который содержит всю необходимую информацию для запуска задач при данном шаге обработки (recon, AOD), моделирования (evnt, simul) или анализа.

Исполняемый шаг обработки (step execution) – иерархически зависимые «инициализированные» шаблоны шага. Исполняемый шаг транслируется в выполняемые задания и хранит их текущее состояние, как и состояние всего шага в целом.

Вертикальный срез (slice) - комбинация из исполняемых шагов обработки, каждый *запрос* состоит из одного или нескольких *срезов*.

Задание (task) - сущность для передачи параметров в систему запуска задач. Имя задания формируется автоматически исходя из информации в *запросе, шаге,...* Каждое задание имеет уникальный идентификатор. Задание может иметь входные данные, результатом выполнения *задания* является набор файлов, организованный как датасет. Имя датасета наследуется из имени *задания,* с учетом выходных форматов, определяемых в *шаге* выполнения задания, и версии ПО, используемого для данного шага. Задание имеет состояния, описывающие ход его подготовки и выполнения.

Задача (job) - каждое задание состоит из одной или многих задач (до 10 тысяч). Задача является единицей измерения работы системы управления загрузкой.

Задача выполняется на единичном вычислительном элементе грид (см главу 1), для суперкомпьютеров задача выполняется на рабочем узле СК. Задача может иметь входные данные (файлы) и записывает результат работы в выходные файлы. Задачи имеют состояния, описывающие ход их подготовки и выполнения. Каждая задача имеет уникальный идентификатор.

Система хранит информацию о всех запросах, заданиях и задачах, созданных и выполненных в системе, а также метаинформацию о ходе выполнения каждой из сущностей.

Пилотная задача (pilot job) - некоторый шаблон задачи, выполняемый на СЕ, запрашивающий реальную задачу, при наличии определенных условий (например, свободного ресурса, наличия версии ПО,…). Пилот следит за выполнением задачи на СЕ, отвечает за передачу результатов выполнения задачи на элемент хранения и удаляет информацию о выполнении задачи на рабочем узле, после ее выполнения. Пилотные задачи имеют состояния, описывающие ход их выполнения. Задача не может быть выполнена (а более точно быть направлена для выполнения на СЕ), если нет информации о пилотной задаче, успешно работающей на данном вычислительном элементе.

Центральной сущностью рабочего процесса является запрос. Пользователь создает запрос, определяет его параметры и набор входных данных. К каждому набору входных данных пользователь задает последовательный набор шаблонов шагов выполнения, который транслируется системой в последовательность выполняемых заданий.

Основной особенностью созданной модели данных является логическое разделение этапов обработки(“запроса”) на “срезы”, «шаги», «задания» и «задачи». «Шаги» являются шаблонами, из которых после инициализации всех параметров создаются «задания», которые, в свою очередь, транслируются в «задачи» и непосредственно отвечают за обработку данных на вычислительных ресурсах.

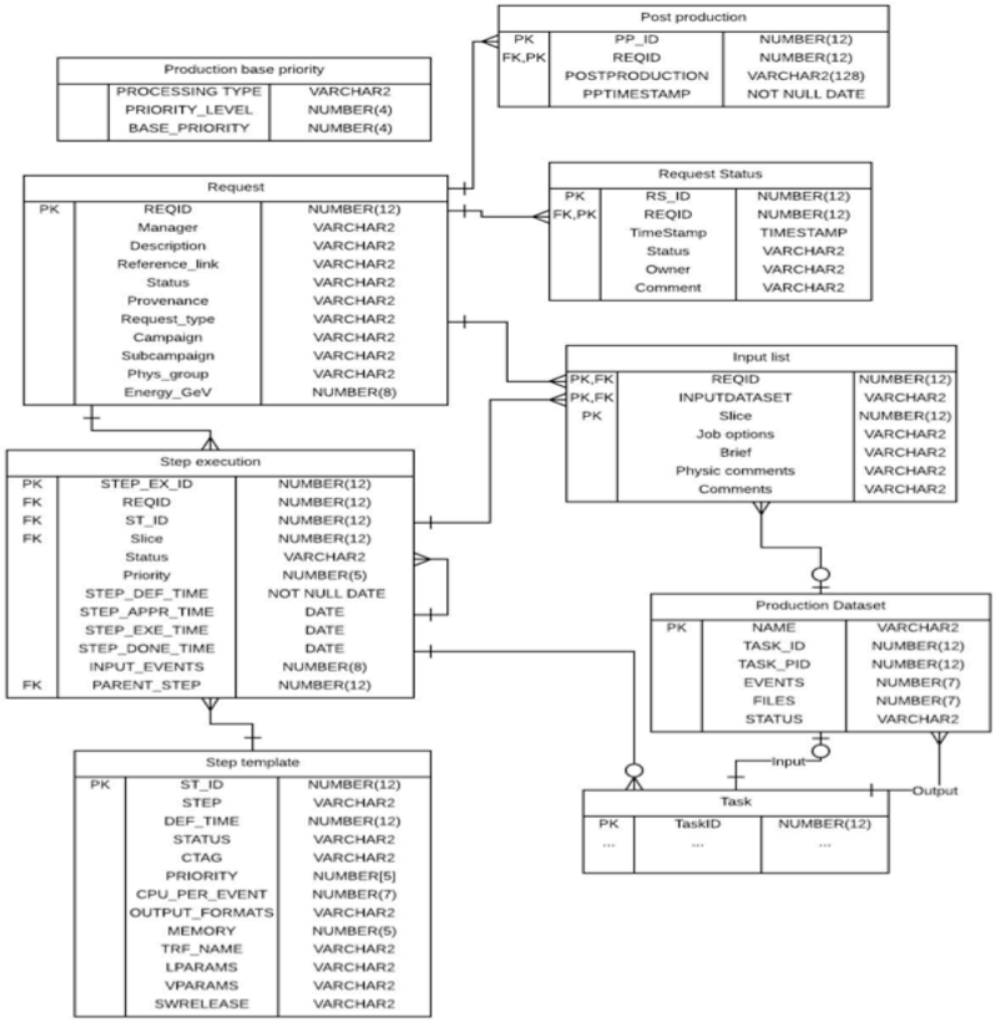


Рисунок 37 - Логическая схема данных

Набор «шагов» также является шаблоном, что позволило реализовать сложные процессы обработки, моделирования и анализа данных физического эксперимента (как будет показано далее данная модель была успешно применена для решения задач биоинформатики). Логическая модель данных приведена на рисунке 37. В данной модели также использовались, ранее предложенные автором, понятия “датасет” и “контейнер” для организации данных [58]. Файлы одного формата, произведенные одним “заданием”, были организованы, как единый набор (“датасет”), который является единицей управления данными. Наборы (“датасеты”), произведенные разными заданиями, но имеющие одинаковые метаданные (версия ПО, используемого для обработки, энергия ускорителя, калибровочные константы)

помещаются в один контейнер. Контейнер может использоваться как “входной” набор данных для задания.

3.3 Новые методы организации поточной обработки данных. Обработка данных “поездом” и “постоянная” обработка данных

Созданная модель данных, введение понятий “датасет” и “контейнер”, позволила реализовать новые подходы по созданию потоков обработки данных физическими группами. Опыт обработки данных физическими группами во время первого этапа работы LHC требовал упорядоченного подхода, это диктовалось как ограничениями, связанными с имеющимися вычислительными ресурсами (ресурс грид был практически полностью использован), так и тем, что алгоритмы обработки данных отдельными группами были схожи между собой и различались только на конечном этапе фильтрации событий [72]. На рисунке 38 приведена статистика обработки для 1400 наборов данных (шкала абсцисс) различными физическими группами (шкала ординат). Цветом показано количество заданий для (пере)обработки данных. Из графика следует, что группы во многих случаях работают с одинаковыми наборами данных и количество попыток обработать один набор группами JetMet и SuSy достигло 14 раз. Созданная модель и методы обработки, рассмотренные ниже, позволили существенно снизить общее (в целом по эксперименту, или виртуальной организации) время обработки, уменьшить количество ошибок при создании запросов на выполнение заданий, уменьшить общее число выполняемых заданий и уменьшить общее время необходимое для получения научных результатов. Двумя такими методами стали создание запросов на «постоянную обработку» и запросов на обработку данных «поездом».

“Постоянная обработка”. Как было рассмотрено в разделе 3.1, входными данными для класса заданий, выполняемых физическими группами, являются данные произведенные централизовано после шага реконструкции.

Классическая методика состояла в том, что только после завершения этого шага начинается выполнение заданий физических групп. Это требовало "отслеживания" выполнения шага реконструкции со стороны многих групп. Новая модель позволила существенно упростить весь процесс. Пользователь (в данном случае физическая группа) заранее определяет "шаблон шагов" выполнения "*запроса*", где "прописан" набор шагов обработки, и поток входных данных. Поток входных данных содержится в контейнере. После чего запрос поступает на выполнение в систему распределенной обработки данных. На момент поступления запроса "контейнер" может не содержать данных или содержать только часть из них. По мере заполнения контейнера данными, автоматически определяются задания обработки и/или анализа, согласно предопределенным шагам шаблона. Реализация такого метода показала свою эффективность для запуска заданий отдельными физическими группами, где входным потоком являются данные полученные после первичной обработки, это позволило быстро начать этап анализа новых данных, с задержкой менее 36 часов, после их получения.

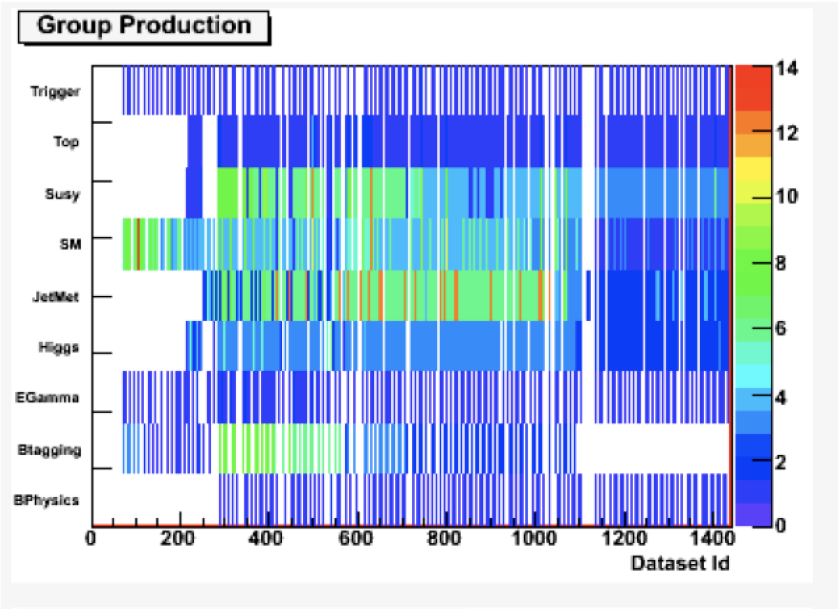


Рисунок 38 - Статистика обработки наборов данных физическими группами эксперимента ATLAS

Обработка данных «поездом». "Обработка поездом" позволяет в одном запросе запустить задания с ПО для нескольких физических групп, для одного и того же набора данных. Каждая группа определяет ПО, т.о. "поезд" имеет столько "вагонов" сколько различных версий ПО было определено.

Но при формировании заданий одинаковые версии ПО составляют единое задание (“физические группы” попадают в “один вагон”). Сложность работы с такими запросами является большое количество возможных типов обработки для большого количества версий программного обеспечения. Основываясь на выбранной модели данных реализация данного метода позволила существенно автоматизировать процесс создания подобных задач.

Рисунок 39 - Время задержки при начале обработки данных физическими группами

Следует также отметить, что также был реализован подход “постоянный поезд”, когда в качестве входных данных использовался контейнер, описанный в параграфе “постоянная обработка”. В таком случае общее время обработки оптимизируется, поскольку задачи задания n с входными данными, произведенными заданием m, могут начаться, как только будут произведены первые файлы, т.е. еще до полного завершения задания m.

Результат реализация новых методов обработки показан на рисунке 39, более чем в 75% случаев физические группы начали обработку в течение 24 часов после того как данные были доступны на грид сайтах.

3.4 Архитектура системы управления загрузкой и глобальной обработки данных физического эксперимента

Принятие концепции грид и модели распределенной обработки данных, потребовало создания нового поколения систем управления загрузкой (систем управления потоками заданий) и пересмотра парадигмы как вычислительный ресурс используется для различных классов физический задач. Рассмотрим каким

требованиям должна отвечать система управления загрузкой современного физического эксперимента:

- сотни вычислительных центров, распределенных по всему миру, для конечного пользователя должны “выглядеть” как единый ВЦ;
- система должна обеспечивать доступ и выполнение заданий на O(100) ВЦ, для O(1000) пользователей и O(1000000) задач в день;
- вычислительные центры могут быть представлены не только центрами грид, но и суперкомпьютерными центрами и центрами облачных вычислений, при этом все центры рассматриваются как равноправные участники. Весь набор центров составляет единый пул вычислительных ресурсов. Отметим, что этот подход стал возможен после перехода к описанной в первой главе диссертации “смешанной” компьютерной модели и обеспечил создание гетерогенной киберинфраструктуры, интегрировав дополнительные вычислительные ресурсы с грид ресурсами.
- очередь на выполнение пользовательских заданий в распределенной среде должна быть единой, сравнимой по функциям с очередью пакетной обработки на локальном ВЦ. Все участники эксперимента (“виртуальной организации”) должны иметь доступ к ресурсам VO через единую систему запуска заданий или, на более высоком уровне, через систему “*запросов*”;
- ошибки в работе вычислительных центров и задержки, связанные с распределенным характером обработки должны быть минимизированы. Для этого необходимо использовать “позднюю привязку” реально выполняемой задачи к вычислительному ресурсу, используя концепцию “пилотных задач”;
- сложность и разнообразие промежуточного программного обеспечения (ППО) грид должны быть “скрыты” от пользователя. Для этого необходимо выполнение следующих условий : а) система управления загрузкой “знает” о ППО и взаимодействует с ним (в обоих направлениях), конечный пользователь

взаимодействует только с системой управления загрузкой; б) механизмы автоматизация управления загрузкой “скрыты” от пользователя;

- изменения и эволюция ППО не должны менять пользовательский интерфейс управления заданиями;
- система должна быть адаптируема к изменениям в аппаратном и программном обеспечении вычислительных центров, и эти изменения не должны быть видимы для пользователя;
- единая система управления загрузкой должна использоваться для всех классов задач физического эксперимента, таких как моделирование, реконструкция, физический анализ, а также для потоков заданий, генерируемых экспериментом, физическими группами, отдельными учеными;
- система должна обладать высокой степенью автоматизации в части обнаружения и “исправления” ошибок, связанных со сбоями в работе распределенной инфраструктуры;
- мониторирование и контроль должны быть частью системы управления загрузкой;
- задания должны использовать ресурс, выделенный для работы виртуальной организации, согласно единой системе приоритетов, пользовательских квот, квот для классов задач.

Архитектура системы должна была разработана таким образом, чтобы обеспечить непрерывный и оптимальный доступ научного сообщества к вычислительными ресурсам. Это должно быть достигнуто за счет использования расширяемой многоуровневой архитектуры. На рисунке 40 схематично представлены уровни системы управления потоком заданий.

Архитектура имеет три основных уровня абстракции

- DEFT (Database Engine For Tasks) - подсистема верхнего уровня. DEFT принимает запросы на выполнение заданий (через специальный пользовательский интерфейс и или из подготовленных списков в формате

удобном для пользователя: текстовый файл, документ google или excel). DEFT обрабатывает запросы и отвечает за формирования шагов обработки, заданий, входных данных и параметров;

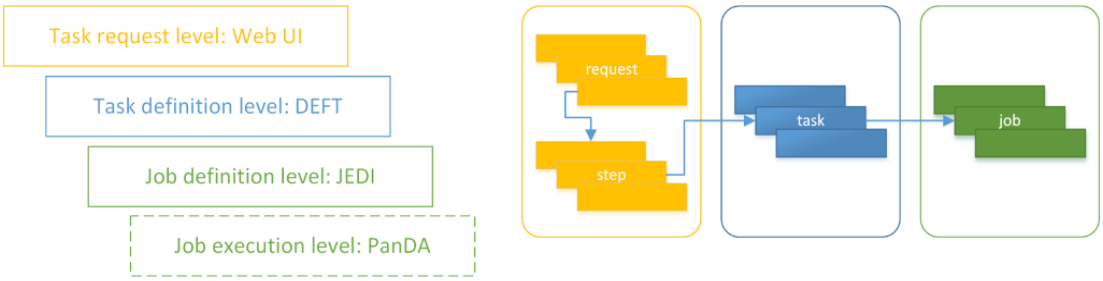


Рисунок 40 - Логические уровни системы управления потоками заданий

– JEDI (Job Execution and Definition Interface) - подсистема среднего уровня, использующая описания заданий, подготовленные DEFT. JEDI динамически определяет количество задач для каждого задания и отвечает за запуск и выполнение отдельных заданий;
– PanDA - основной «движок» системы, подсистема выполнения задач. PanDA определяет какой ресурс и в какой момент будет использован каждой из задач, получает информацию от пилотных заданий и информационной системы, управляет ходом выполнения задач.

Информация о выполняемых заданиях находится под контролем JEDI и хранится в рабочей базе данных (БД). Программы мониторирования и учета (аккаунтинга) используют копию БД JEDI, что гарантирует отсутствие нежелательного доступа к рабочей БД, включая запросы мониторирования, требующие агрегации информации из различных таблиц БД, что может привести к снижению производительности системы в целом. Были определены основные архитектурные компоненты (подсистемы) системы управления загрузкой (PanDA WMS) и их функции. Каждая подсистема должна иметь общие компоненты и настраиваемые уровни. Специализированные уровни должны быть конфигурируемы. Основные компоненты системы управления загрузкой, их

взаимодействие между собой, взаимодействие с внешними системами (хранения и доступ к метаинформации, управления данными DDM, информационная система), а также вычислительными ресурсами показаны на рисунке 41. Система должна обеспечивать управления загрузкой с учетом особенностей трех реализаций грид в рамках WLCG: проект EGEE/EGI, проект NorduGrid, проект OSG, а также использовать дополнительные ресурсы, в том числе университетские кластеры, ресурсы облачных вычислений и суперкомпьютеры.

Рассмотрим основные компоненты уровней системы управления загрузкой и их функции:

Уровень DEFT (Database Engine For Tasks). Принимает пользовательские запросы, проверяет их корректность и формализует запросы. DEFT формирует последовательность выполнения заданий для каждого запроса. DEFT получает необходимую информацию от систем управления данными и системы хранения метаинформации и имеет следующие компоненты:

– **Интерфейсы контроля и управления DEFT**

Интерфейс программирования приложений (API) должен быть реализован в виде отдельного веб-сервиса, поддерживающего протокол RESTful. В API должны быть реализованы основные сервисы для мониторинга и управления объектами системы. Соответствующий веб-сервис должны обеспечивать аутентификацию с помощью специальных цифровых ключей для разграничения доступа к системе и журналирования действий пользователей. Система должна иметь механизм отложенного выполнения запросов и автоматического восстановления после сбоя. Необходимо иметь возможность управления *запросами, срезами,* отдельными заданиями в системе. Важной характеристикой является стабильность и скорость работы API. Интерфейс должен быть интегрирован с системой обработки ошибок для оперативного отслеживания возможных проблем при выполнении запросов к системе.

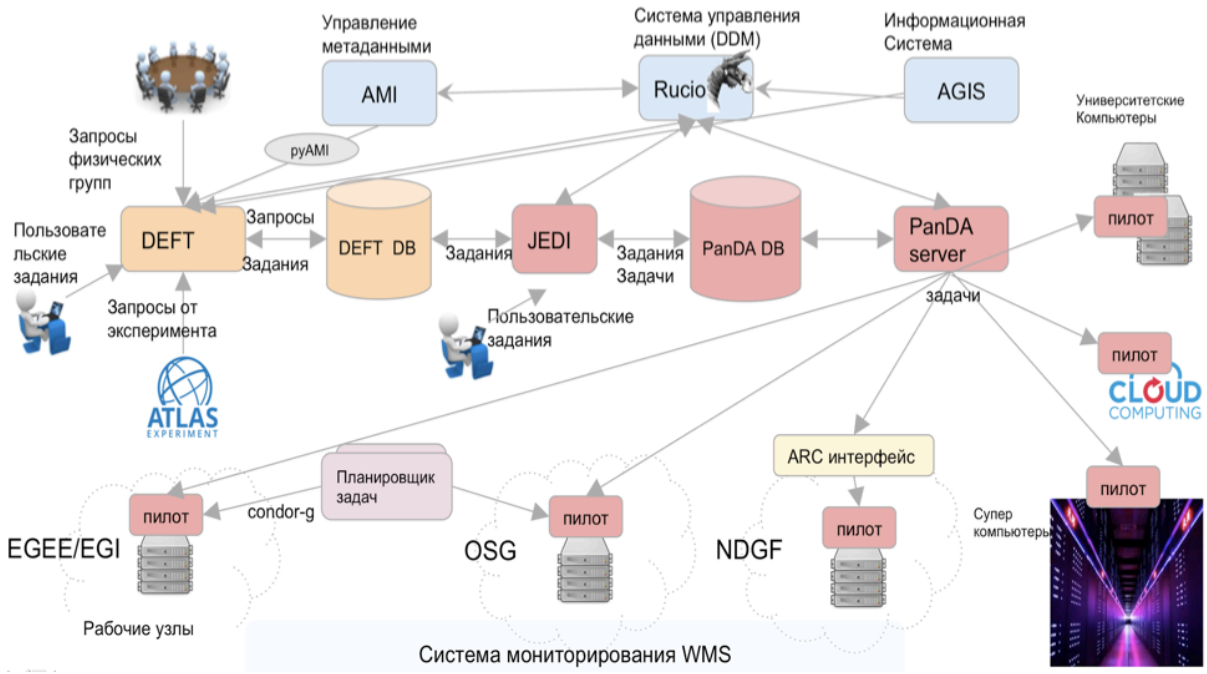


Рисунок 41 - Схема управления потоком заданий и основные компоненты системы для распределенной обработки данных

- **Интерфейс управления запросами, шагами выполнения, заданиями, задачами.** Данная группа интерфейсов позволяет контролировать выполнения запросов, заданий, задач, например, (при)останавливать их выполнение, изменять параметры задания, перенаправлять задания между ВЦ.
- **Интерфейсы управления потоками заданий**. Управления специальными потоками, такими как “постоянная обработка”, “обработка поездом”.
- **База данных.** База данных запросов, шагов выполнения и заданий в масштабе всей системы, которая хранит всестороннюю статическую и динамическую информацию и метаинформацию обо всех запросах, “шагах” и заданиях определенных, выполняемых и выполненных в системе, в том числе историю их выполнения, текущее состояние и статус, ошибки, возникшие при этом. Информация о заданиях в момент их выполнения синхронизуется с БД следующих уровней (JEDI, PanDA).

Уровень JEDI (Job Execution and Definition Interface). Принимает формализованные описания заданий от DEFT, определяет ресурс для выполнения задания, определяет количество задач и «разбивает» задание на задачи. JEDI проверяет информацию о данных (через систему управления данными, DDM). JEDI работает с “рабочими очередями”, описанными в ИС. JEDI и PanDA используют общую базу данных для хранения информации о состоянии заданий и задач.

Уровень PanDA (Production and Distributed Analysis). PanDA - мотор всей системы и наиболее сложная ее часть, она должна включать в себя следующие компоненты,

- **сервер.** Сервер является основой системы управления загрузкой и должен быть создан как общий сервис WMS;
- **база данных.** База данных заданий и задач в масштабе всей системы, которая хранит всестороннюю статическую и динамическую информацию и метаинформацию обо всех заданиях и задачах, определенных, выполняемых и выполненных в системе, в том числе историю их выполнения и ошибки, возникшие при этом.
- **пилот**. Пилотные задачи используются для сбора информации о состоянии вычислительных ресурсов. Рабочие задачи передаются успешно активированным и проверенным пилотам сервером WMS на основе критериев выбора ресурса. «Поздняя привязка» рабочих задач к месту выполнения предотвращает задержки и отказы, и максимизирует гибкость выделения ресурсов для задачи на основе динамического состояния обрабатывающих ресурсов и приоритетов задач. Пилот - также основной 'изолирующий слой' для WMS, инкапсулирующий сложные неоднородные среды и интерфейсы грид и средств, с которыми взаимодействует WMS. Пилотные задания для анализа способны переключить свои идентификационные данные на рабочем узле на данные пользователя

запустившего задание, используя инструмент grid, если правила компьютерной безопасности сайта (ВЦ) того требуют.

- **система распределения заданий (брокер)**. WMS брокер – это интеллектуальный модуль, выбор ресурса происходит на основе типа и приоритета задания, наличия программного обеспечения, входных данных и их местоположения, статистики задания в реальном времени, и доступного ЦПУ, ресурсов хранения, связи ВЦ с “внешним миром” (пропускная способность WAN). Это - ключевой компонент автоматизации потока операций WMS.
- **диспетчер WMS**. Диспетчер - компонент в сервере WMS, который получает запросы на задачи от пилотов и диспетчеризирует задачи, используя информацию о заданиях в очереди(ях) к данному ВЦ, их приоритетов, квот, политики распределения ресурсов и стратегии повторного запуска задачи (например, n-кратный запуск задачи на рабочем узле в случае отсутствия фатальной ошибки).
- **автоматическая фабрика пилотных задач (АФП)**. АФП - независимая от WMS подсистема, которая управляет поставкой пилотных задач к рабочим узлам (CE). Пилот, запущенный на рабочем узле, связывается с диспетчером и получает доступную задачу, для задания, выполняемого на данном ВЦ.

Важным свойством этой схемы псевдо-интерактивного анализа, где важна минимальная задержка от запуска задачи до начала ее выполнения, является то, что диспетчеризация пилотных заданий обеспечивает устранение любых задержек в системе планирования при запуске самого пилотного задания. Механизм пилотных заданий изолирует рабочие задания от сбоев в работе грид и систем пакетной обработки (рабочие задачи предоставляются на сайт только после успешного запуска пилотной задачи).

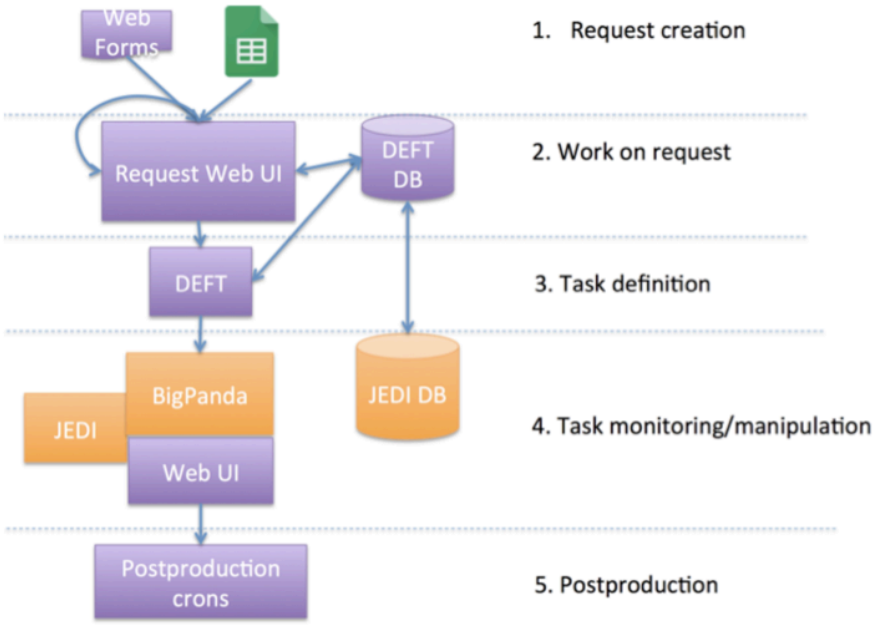


Рисунок 42 - Схема взаимодействия различных уровней WMS

На рисунке 42 схематически показано взаимодействие между уровнями WMS, на рисунке 43 показано взаимодействие между уровнями DEFT и JEDI при обработке запроса и выполнения задания. Эта система получила название ***megaPanDA***.

Информационная система, система мониторирования, интерфейсы контроля и управления WMS, система компьютерной безопасности и аутентификации работают со всеми уровнями, их основные функции:

Информационная система (ИС). База данных, хранящая информацию о вычислительных центрах и очередях в масштабе всей распределенной киберинфраструктуры. ИС хранит статическую и динамическую информацию, используемую WMS (особенностью ИС AGIS, описанной ранее, стало ее использование как системой управления загрузкой (WMS), так и системой управления данными (DDM). ИС должна быть конфигурируема и иметь возможность изменения информации в “ручном режиме”.

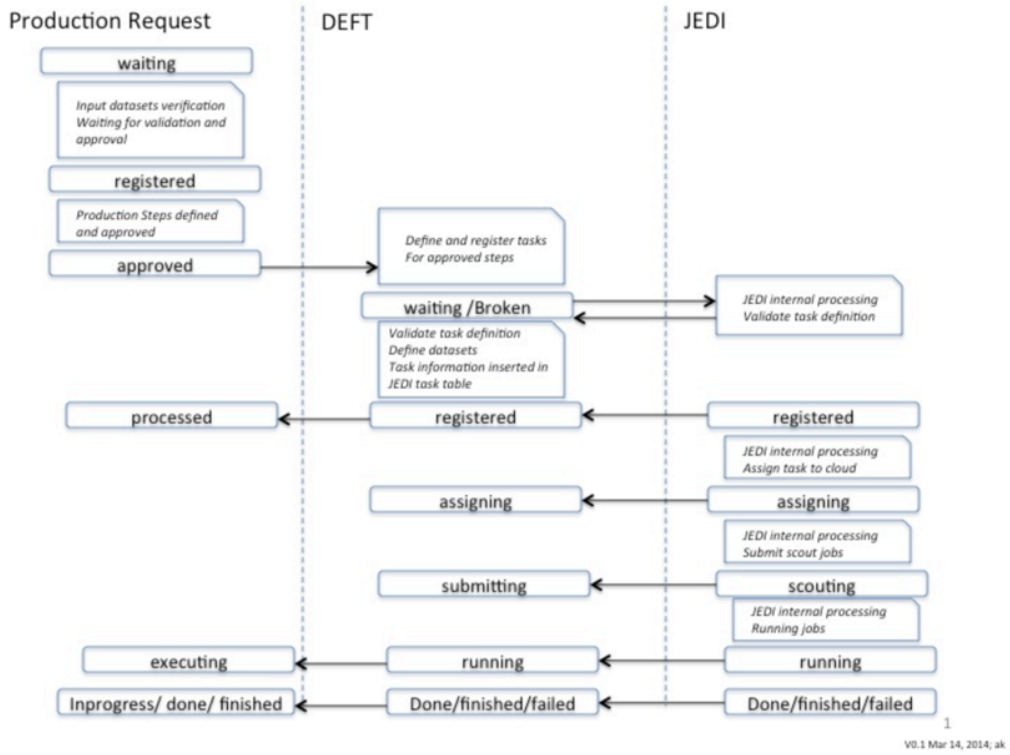


Рисунок 43 - Схема взаимодействия уровня DEFT и JEDI

Через ИС происходит контроль вычислительных ресурсов и параметров отдельных очередей. Содержимое базы данных ИС - это информационный кэш, агрегирующий и интегрирующий данные из отдельных информационных систем грид, систем управления данными и других источников. Пилотные задачи запрашивают информацию от ИС, чтобы сконфигурировать задачу в соответствии с параметрами очереди, в которую задачу направил WMS брокер.

Интерфейсы контроля и управления системы управления потоком заданий

Интерфейс запуска заданий и задач. Пользовательский интерфейс, который обеспечивает интеграцию с разнообразными средствами для запуска заданий и задач в систему управления загрузкой. Интерфейс используется как при отладке и настройке WMS, так и предоставляет возможность для пользователя запустить задачу (или задание) и определить ее параметры

(например, пользователь хочет осуществить брокеровку задачи вручную, указав определенный ВЦ, в качестве места выполнения задачи).

Интерфейс управления системой. Управление потоком операций. Управление использованием ресурсов для пользователя, групп пользователей и регулирование квот использования ресурсов. Управление может быть как глобальным (на уровне всей системы), так и локальным (на уровне одного из ВЦ или определенной очереди).

Интерфейс управления запросами, шагами выполнения, заданиями, задачами. Данная группа интерфейсов позволяет контролировать выполнения запросов, заданий, задач, например, (при)останавливать их выполнение, изменять параметры задания, перенаправлять здания между ВЦ.

Интерфейсы управления потоками заданий. Управления специальными потоками, такими как “постоянная обработка”, “обработка поездом”.

Система мониторирования. Всесторонний мониторинг заданий (и задач), для всех классов заданий: задания всего эксперимента, задания отдельных групп и отдельных ученых. Система мониторирования а) предоставляет подробную информацию о запросах, заданиях, задачах и сайтах для диагностики их состояния и возможных проблем; б) отображает информацию об использовании процессорного времени, квотах, правильности работы и производительности подсистем megaPanDA и используемых вычислительных средств.

Система аутентификации и компьютерной безопасности. WMS должна быть интегрирована с соответствующими системами безопасности грид.

Управление данными после завершения работы задания и/или завершения запроса (PostProduction). Этот уровень необходим для управления состоянием заданий и запросов после завершения работы заданий. Например, при повторной обработке экспериментальных данных с уточненными значениями калибровок детектора или при повторении реконструкции моделируемых данных с

новой версии ПО. В обоих случаях данные произведенные ранее могут быть признаны “устаревшими” (как это обсуждалось в разделе 1.4.2). В таком случае статус заданий также должен быть изменен, а данные удалены со всех элементов хранения VO. Этот уровень может быть реализован как набор программ-агентов и через пользовательский интерфейс. Например, запрос может быть направлен не только на “производство” данных, но и на их удаление.

Более детально архитектура системы для распределенной обработки данных и взаимодействие между различными ее подсистемами показано на рисунке 44.

3.5 Методика управления потоками заданий и задач

Методика управление потоком зданий и задач на первом этапе работы LHC. Разработка методики распределения ресурса в рамках модели MONARC была основана на иерархии центров, и статическим характером организации групп центров грид. Все центры были организованы как набор “региональных облаков”, когда каждому центру уровня T1 соответствовало от 2 до 14 центров уровня T2 (схематично это показано на рисунке 45. Понятие «облако» было введено до появления облачных ресурсов, и в данном контексте соответствует группе из одного T1 и нескольких T2 центров. Подробно модель MONARC описана в первой главе диссертации). Рассмотрим ограничения накладываемые на систему обработки, при такой организации вычислительных ресурсов.

Выбор “облака” для выполнения задания происходил в два этапа. На первом этапе брокер WMS должен был выбрать из 11 “облаков” наилучшее для выполнения всего задания, выбор происходил по совокупности следующих параметров:

- наличие входных данных для выполнения задания (если это требуется);
- свободное дисковое пространство на сайте T1 (т.к. согласно компьютерной модели результаты работы задания должны быть переданы на хранение на сайт T1);
- свободные вычислительные ресурсы в облаке;

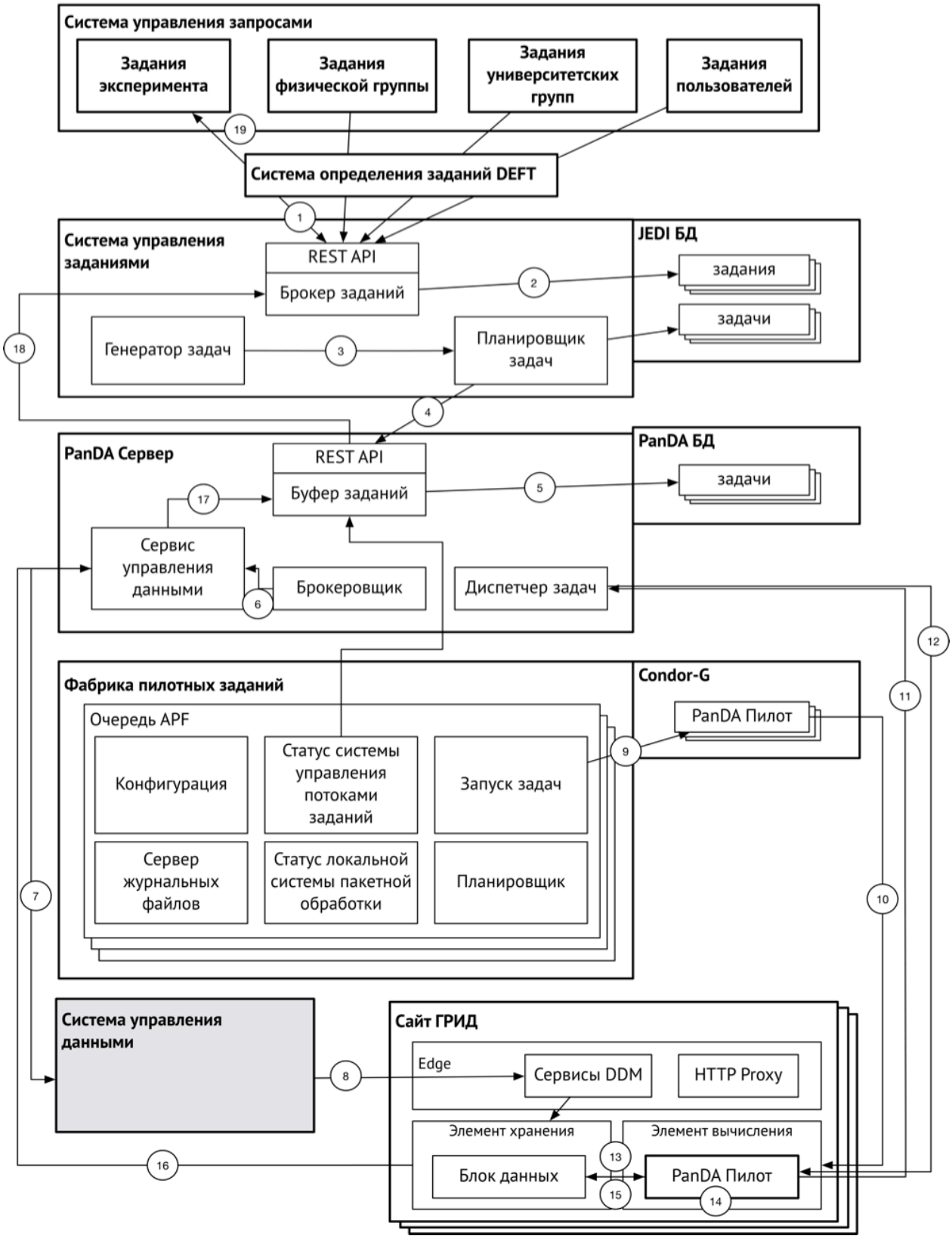


Рисунок 44 – Архитектура системы для распределенной обработки данных megaPanDA

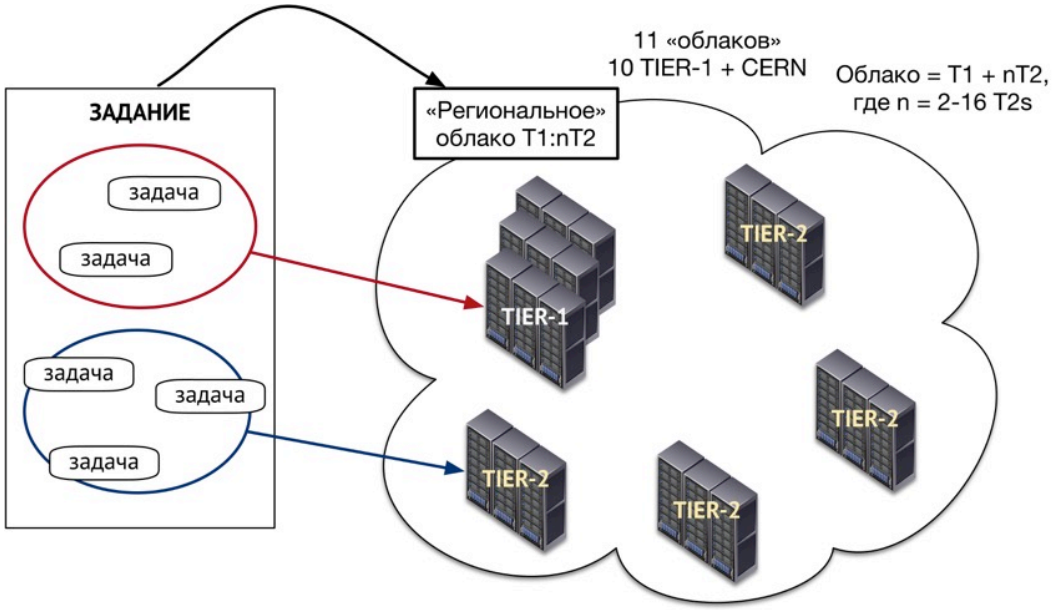


Рисунок 45 – Организация центров эксперимента ATLAS на первом этапе работы LHC

- планируемая плановая остановка T1;
- число заданий в очереди на выполнения в данном облаке и их приоритет.

На втором этапе происходила брокеровка задач и выбор сайтов в рамках “облака” для их выполнения. Выбор сайта для каждой задачи, учитывал следующие параметры:

- данные находятся на ленте или задача требует интенсивного ввода/вывода информации (например, фильтрация событий);
- оценка требуемого задачей ЦПУ ресурса;
- наличие необходимой версии ПО эксперимента;
- размер свободного дискового пространства на элементе хранения (SE), размер свободного дискового пространства для временного хранения, размер оперативной памяти на рабочих узлах Tx;
- текущая загрузка Tx, которая определялась как отношение:

Occupancy = Nr/Nq, где

- Nr - число, число выполняемых на сайте задач
- Nq - число задач ожидающих выполнения, находящихся в очереди

- планируемая остановка сайта для профилактики.

Такая модель работала, но брокеровка в два этапа приводила к тому, что при наличии свободного вычислительного ресурса в центрах T2, задание ожидали выполнения, т.к. не было “облака” со свободным T1. Методика управления загрузкой “наследовала” все проблемы типичные для модели MONARC:

- использование вычислительных и дисковых ресурсов было неоптимальным;
    - более того, использование ресурса центров второго уровня зависело от стабильности и размеров центра T1;
    - создавалось несколько промежуточных копий данных, чтобы обеспечить выбор из нескольких облаков. Так количество копий данных формата EVNT (наборы сгенерированных событий) достигало 7, а для наборов AOD количество копий достигало 8;
- выполнение заданий с высоким приоритетом могло “зависнуть” в “облаке” с центром первого уровня, у которого было недостаточно вычислительного и/или дискового ресурса;
- передача данных (рисунок 9) вводила дополнительную задержку, т.к. существовало дополнительное ограничение, связанное с периметром каждого облака и для успешной передачи требовалось наличие свободного дискового ресурса на двух центрах T1.

Но главная проблема состояла в том, что внутри грид вычислительные ресурсы рассматривались как отдельные группы (T1 и ассоциированные с ним центры второго уровня: T2). Существовала статическая связка T1:nT2, когда до 15 центров могли стать недоступны для выполнения задач при остановке T1 или одного из его сервисов, например, сетевого оборудования. Развитие глобальных вычислительных сетей, многократное увеличение их пропускной способности в последнее десятилетие, создание термодинамической модели, методики определения

стабильности центров, создание принципиально новой информационной системы (AGIS), использование информации о пропускной способности WAN, включение WAN (наряду с вычислительным и дисковым ресурсами) в общий компьютерный ресурс центров и как результат переход к “смешанной” компьютерной модели, позволили перейти от 11 “облаков” со строгой “привязкой” Т2 центров к Т1 центру, к совершенно новой конфигурации, названной “всемирным облаком” и новому подходу при управлении потоками заданий и при обработке данных.

“Смешанная” компьютерная модель и “всемирное облако”. Новый подход должен был обеспечить более гибкое и динамичное использование имеющегося ресурса, устранить искусственные границы модели MONARC, и как результат, более эффективно использовать вычислительный ресурс. Единицей выполнения является задание, все ресурсы рассматриваются как общий ресурс. Для выполнения каждого задания система создает “облако”. В облако может входить от одного до всех центров. Облако состоит из “ядра” и “спутников”. Любой центр (Т1 и Т2) может стать ядром (это зависит от размеров центра, его стабильности и пропускной способности WAN). Информация о том, что центр может быть ядром “всемирного облака” хранится в информационной системе AGIS. В новой модели брокеровщик заданий выбирает “ядро” для выполнения задания и необходимое количество спутников. Результат выполнения задания (выходные данные) передаются на хранение в сайт-ядро. Сайты-спутники выбираются из всех имеющихся центров (Т1 и Т2), на основании критериев, описанных ранее, и с учетом информации о состоянии WAN между ядром и спутником. Эта работа выполняется программой конфигуратором, получающей информацию от систем управления данными, ИС и WAN (рисунок 46).

Рассмотрим более подробно методику выбора сайта-ядра для “всемирного облака”. Сайт-ядро должен удовлетворять всем требованиям, перечисленным ниже :

- сайт не планирует остановку в ближайшие 72 часа;

– сайт имеет 5Тбайт (или более) свободного дискового пространства, включая оценку необходимого дискового пространства для всех заданий (и задач), находящихся в очереди на выполнение к данному сайту;

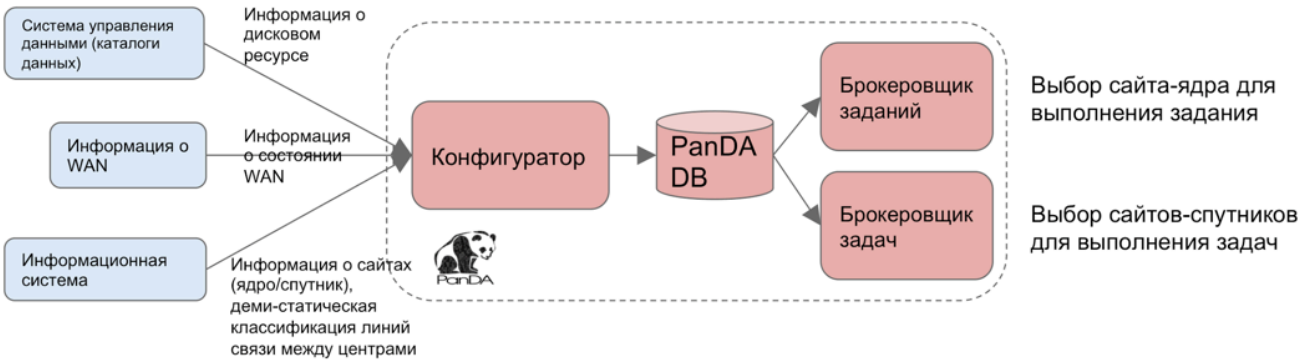


Рисунок 46 - Логическая схема работы конфигуратора “всемирного облака”

– число файлов, которые находятся в очереди на передачу на сайт-ядро не должно превышать двух тысяч (чтобы избежать задержки в выполнение заданий из-за проблем с DDM/FTS);
– не менее 10% входных данных (при их наличии) в Тбайтах и не менее 10% входных файлов находится на сайте-ядро (при размере входных данных более чем 1Гбайт и 100 файлов);

Все параметры (5Тбайт, 72 часа, 2К файлов) были выбраны исходя из оценки наиболее вероятных размеров датасетов и времени выполнения заданий. Рассмотрим более подробно как введение понятия “всемирное облако” позволяет динамично использовать компьютерные ресурсы.

Динамическая модель управления потоками заданий и задач. Последовательность управления потоком заданий показана на рисунке 47. После поступления описания задания в уровень JEDI (описанный в разделе 3.4) необходимо разработать методику выбора наилучшего сайта-ядро и формирования облака из имеющегося глобального вычислительного ресурса.

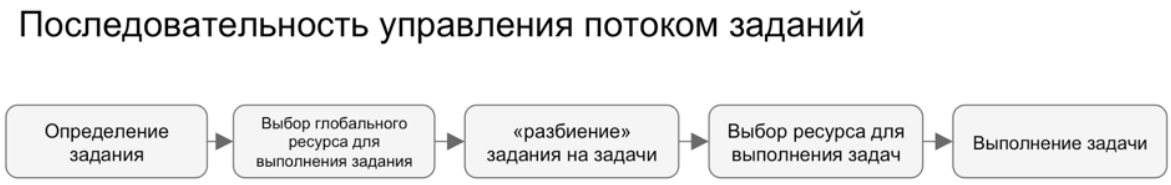


Рисунок 47 - Последовательность управления потоком заданий

Методика выбора сайта-ядра. Из всего набора сайтов, имеющих роль “ядро” в ИС, выбираются, удовлетворяющие требованиями перечисленным выше, остальные сайты не рассматриваются. Для всех потенциальных сайтов ядер рассчитывается величина, называемая *totalRW* (total Remainng Work - общая работа, которая должна быть выполнена всеми сайтами). *totalRW* является суммой *RW* для каждого из потенциальных сайт-ядер, согласно формуле :

RW – величина remaining work вычисляется согласно формуле :

- $RW = (nEvents - nEventsUsed)$x *cpuTime*
  - *nEvents* - общее число событий для обработки/моделирования/анализа в заданиях;
  - *nEventsUsed* - число событий уже обработанных;
  - *cpuTime* - ожидаемое время выполнения задачи (для данного задания), рассчитываемое по первым 5 закончившимся задачам в единицах HS06 (HEPSpec06) и учитывающего мощность каждого сайта. Информации о вычислительной мощности сайтов хранятся в ИС;
- $totalRW = \Sigma RW\ (i = 0,..n)$

  где n – число сайтов, которые могут быть ядрами

После чего рассчитывается вес для каждого потенциального сайта-ядро, как:

$$weight = \frac{1}{totalRW} \times \frac{sizeA}{sizeT} \times TW \times \frac{spaceF}{spaceT}$$

где :

- *sizeA* - объем входных данных, имеющийся на сайте-ядро;
- *sizeT* - общий объем входных данных;
- *TW* - 0.001, если входные данные не имеют копии на диске, и 1, если есть копия на диске;
- *spaceF* - свободное дисковое пространство на сайте;
- *spaceT* - общее дисковое пространство на сайте.

После выбора сайта-ядро (сайт, имеющий наибольший вес), необходимо сформировать “всемирное облако”, добавив сайты-спутники. На первом этапе происходит отбор кандидатов, исходя из следующих параметров:

- информация о доступности сайта и доступности его дисков (SE), например, сайт может быть исключен из списка, если его SE помечен, как нестабильный;
- наличие версии ПО необходимой для выполнения задачи;
- число памяти на ядро (в сравнение с требованиями для данного задания);
- свободное дисковое пространство (не менее 200 Гбайт);
- количество запросов на передачу данных к/от данного сайта, если количество запросов превышает заданный предел, то сайт исключается из списка;
- *T1weight* - величина связана с приоритетом задания, при *T1wieght=1* - все задачи должны быть выполнены на сайте-ядро;
- *WANweight* - величина, пропорциональная пропускной способности между сайтом-ядро и спутником и учитывающая количество файлов в очереди на передачу между сайтом-ядром и сайтом-спутником;

Для потенциальных сайтов вычисляется:

$$manyAssigned = max(1, min(2, assigned/activated))$$

- *assigned* - число задач в очереди к данному сайту;
- *activated* - число пилотных задач, выполняемых на данном сайте;

$$weight = (running/total)/manyAssigned$$

- *running* - число реальных задач, выполняемых на сайте;
- *total* - общее число задач в задании;

После чего вес нормируется согласно информации о входных данных и состоянии WAN.

$$weight = weight * (\frac{availableSize + totalSize}{totalSize} * (\frac{nMissingFiles}{100} + 1))$$

- *availableSize* - размер свободного дискового пространства;
- *totalSize* - общий размер дискового пространства;
- *nMissingFiles* - число файлов за пределами сайта.

$$weight = weight * WANweight$$

Методика расчета веса определяющего качество WAN между сайтом-ядро и сайтом-спутник. Для расчета используется информация из ИС AGIS и системы NWS и вычисляются 2 параметра : queueWeight и throughputWeight (рисунки 48, 49), что позволяет определить качество связи между парой ядро/саттелит.

$$WANweight = 0.5 * queueuWeight + 0.5 * throughputWeight$$

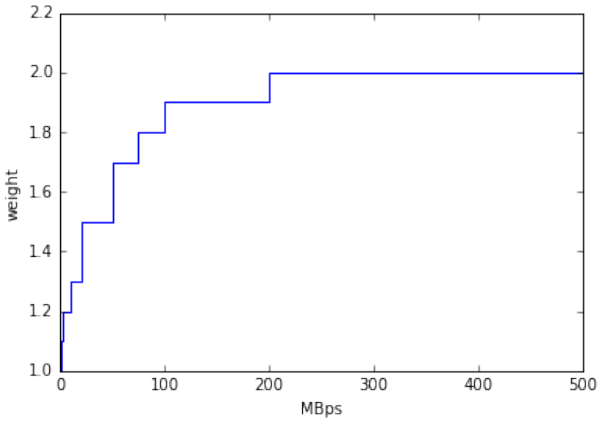


Рисунок 48 - График queueWeight при выборе сайта-спутник

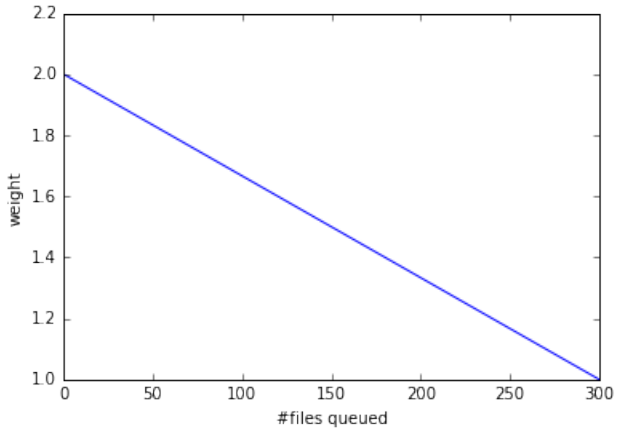


Рисунок 49- График throughputWeight при выборе сайта-спутник

В результате из 10 наилучших сайтов выбираются те, которые вместе с сайтом-ядром составляет “облако” для выполнения задания. На рисунке 50 показано

какие сайты были выбраны как наилучшие для формирования “всемирного облака” с сайтом-ядро (AGLT2 - ATLAS Great Lakes, T2 центр в штате Мичиган, США). Наряду с очевидными кандидатами, такими как BNL (Brookhaven National Laboratory T1 центр в штате Нью-Йорк, США) и MWT2 (Middle West T2, центр в штате Иллинойс, США), в 10 лучших сайтов-спутников вошли сайты в Европе (Франция, Великобритания, Польша).

Изменение роли центров и добавление функции “сайт-ядро” было постепенным, к сентябрю 2016 года более 20% от всех сайтов уровня T2 имели такой статус.

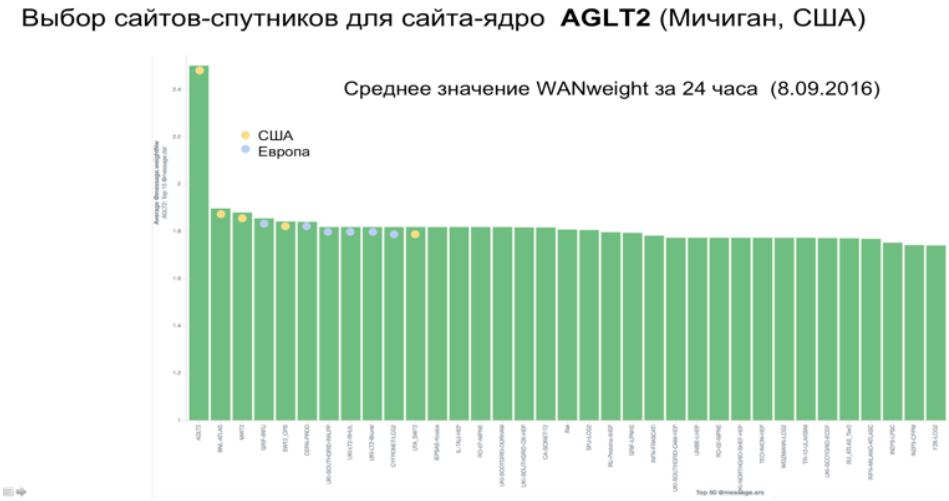


Рисунок 50 - 10 лучших сайтов-спутников для сайта-ядро AGLT2

3.6 Методика распределения вычислительного ресурса между различными потоками заданий физического эксперимента

В предыдущих разделах данной главы мы определили основные классы заданий физического эксперимента и методику распределения заданий и задач между сайтами, а также ввели понятие “всемирное облако”. В данном разделе мы рассмотрим методику разделения ресурсов между различными потоками заданий

внутри одной “виртуальной организации” (эксперимента). Таких потоков может быть три:

1. Потоки заданий, выполняемые WMS для всего эксперимента (“виртуальной организации”)
    a. Обработка и (пере)обработка данных
        - С учетом этапа работы коллайдера/ускорителя в режиме протон/протонных или тяжелоионных столкновений (эксперименты ALICE, ATLAS, CMS на LHC), или программы эксперимента в конкретном году, например мюонной или адронной (эксперимент COMPASS на SPS). Количество ресурсов, выделяемых экспериментом (виртуальной организацией) может быть различно и зависеть от его научных приоритетов.
    b. Моделирование методом Монте-Карло
        - Кампании (и подкомпании) могут частично перекрываться, и приоритет для различных (под)компаний моделирования может быть разным;
    c. Создание приведенных данных для физического анализа;
    d. Обработка данных для специализированных систем, например для системы “триггера высокого уровня”, такие потоки могут иметь специальные требования, например они должны быть выполнены в течение 12 часов;
    e. Специальные случаи, например “обработка поездом” для всех или нескольких физических групп;
2. Потоки заданий, выполняемые WMS для отдельных физических групп
    a. Создание данных для физического анализа, проводимого физической группой

b. Анализ данных

3. Потоки заданий, выполняемые WMS для отдельных пользователей

a. Физический анализ данных

Использование методики “обработки поездом” позволяет объединить потоки 1). и 2.a). Важность разделения вычислительного ресурса между различными потоками заданий определяется в первую очередь научными приоритетами эксперимента. Установление квоты для задач анализа, не гарантирует разделение ресурса в силу различий в производительности и стабильности различных ВЦ, выполняемом коде, способах доступа и наличия входных данных. Каждая задача независимо от типа *потока* выполняется в несколько этапов (рисунок 51).

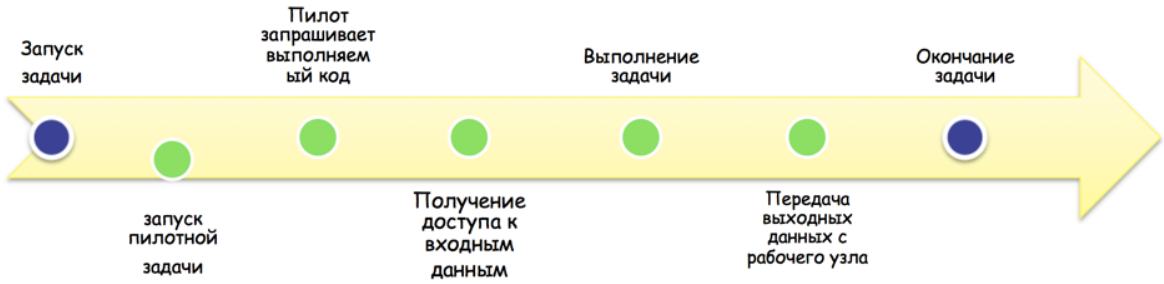


Рисунок 51 - Шаги выполнения задач

Введем следующие определения:

$$\text{время работы задачи}: Te = Tf - Ts$$

$$\text{время ожидания}: Twait = Tp - Ts$$

время выполнения задачи: $Trun$ время выполнения кода

где :

- Ts - время запуска задачи
- Tf - время окончания задачи
- Tp - время запуска пилотной задачи.

На рисунке 52 показано распределение для значений *Twait* и *Trun* на одном из наиболее стабильных ВЦ (T1 BNL, ~25% вычислительных ресурсов эксперимента ATLAS).

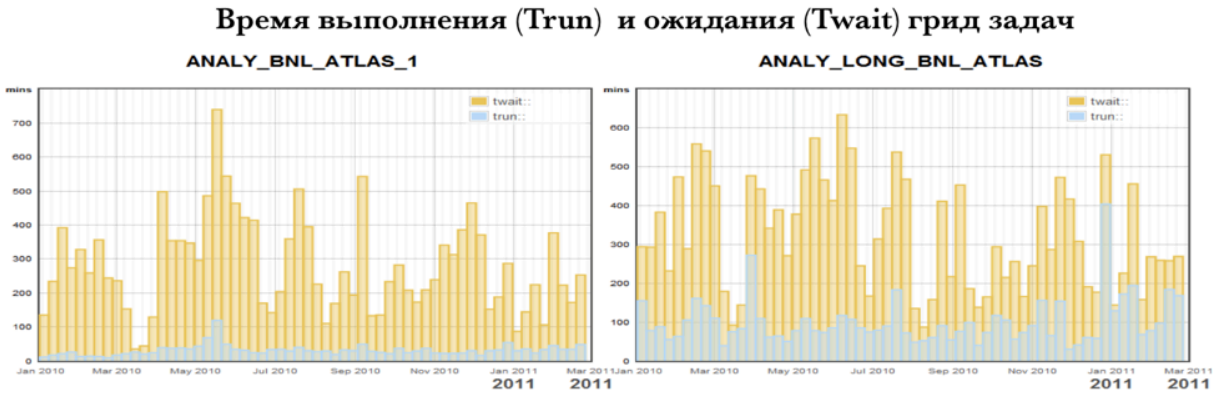


Рисунок 52 - Распределения времени ожидания и выполнения задач для T2 и T1 центров

Из распределения видно, что время выполнения задачи существенно меньше времени ожидания. Для двух разных очередей $Trun$ составляет 55 и 100 минут соответственно, среднее значение времени ожидания ($Twait$) составляет около пяти часов. Из этого следует, что в течение дня ученый может совершить 2-3 попытки для выполнения своего кода и получения физического результата. Это привело к тому, что в первые годы большинство ученых с недоверием относились к инфраструктуре грид, и предпочитали локальные вычислительные мощности.

Для решения этой проблемы необходимо было разработать методику распределения вычислительных мощностей, между различными потоками заданий “виртуальной организации”, а также способ реализации методики, который гарантировал бы “справедливое” разделение ресурса согласно долям, установленным в соответствии с научными приоритетами эксперимента. Такая система получила название “fair share” (справедливое разделение).

Предположим, что имеющийся вычислительный ресурс может быть классифицирован по типу исполняемых задач: HIMEM (для задач с требованием более nГбайт оперативной памяти на ядро (для современных приложений ФВЭ и ЯФ n >= 2 Гбайт), LONG (для задач, требующих 24 часа и более на выполнение), SCORE

(single core, задачи, требующие весь “рабочий узел”), и т.д. Тогда у нас есть матрица ресурсов и потоков, которые должны их использовать. Важно отметить, что в данном подходе все вычислительные ресурсы рассматриваются как равноправные и не учитывается иерархия введенная MONARC (T0,Tn), также важным является факт, что суперкомпьютерный центр (или коммерческие ресурсы “облачных вычислений”) рассматривается как возможный ресурс, и его использование определяется или ограничивается только его возможностями по выполнению потоков задач определенного типа (например “генерация событий”, шаг *evgen,* требующий большого процессорного времени и минимального ввода/вывода), а в случае коммерческих ресурсов их стоимостью.

Все ресурсы должны быть разделены между обработкой/моделированием данных (“производство данных”) и их анализом (уровень L1), и далее между различными потоками по “производству данных” (моделирование, (пере)обработка, триггер высокого уровня, проверкой версий ПО,... - уровень L2), введение следующего уровня (L3) - позволяет ввести дальнейшее разделение ресурса, например между кампаниями по моделированию данных (например, для предполагаемой энергии ускорителя 13 ТэВ или 14 ТэВ, в случае LHC), обработкой данных для протон-протонных (pp) или тяжелоионных (HI) типов столкновений (схематично это показано на рисунке 53).

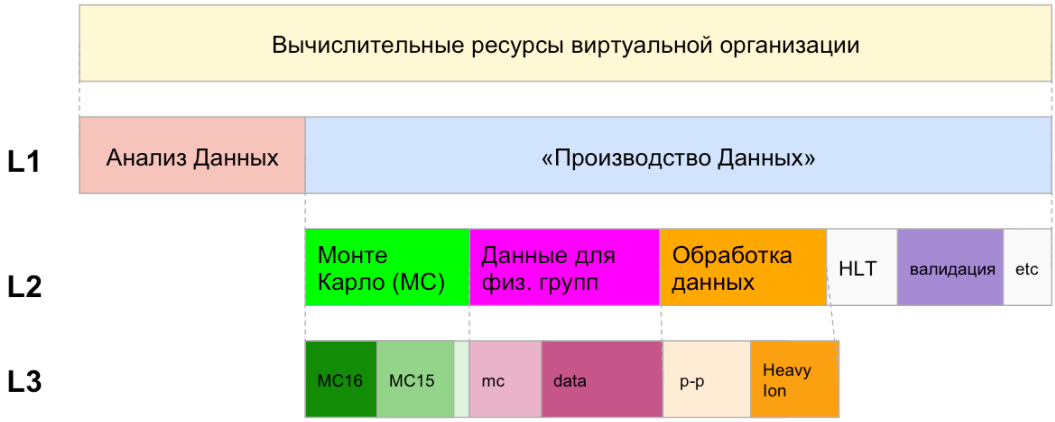


Рисунок 53 - Распределения вычислительного ресурса для различных классов заданий

Для реализации данной модели была предложена и реализована методика иерархического распределения долей, позволяющая динамически разделять весь имеющийся вычислительный ресурс между различными потоками заданий.

Базовые определения модели :

n : типы потоков заданий (Монте-Карло, потоки заданий физических групп, обработка данных, HLT, …)

m : типы вычислительных ресурсов (SCORE, MCORE, HIMEM, LONG, …)

R^S : $n \times m$ матрица, где *(a,r)* элемент матрицы соответствует числу задач класса *(a)*, выполняемых на вычислительном ресурсе *(r)*, в очереди выполнения задач *(s)*

Q^S: $n \times m$ матрица, где *(a,r)* элемент соответствует числу ядер выделенных для задач класса *(a)* и вычислительного ресурса типа *(r)* в очереди выполнения задач *(s)*

A : n-мерный вектор, где элемент *(a)* - выделенный ЦПУ ресурс для задач класса *(a)*

C : n-мерный вектор, где элемент *(a)* - число ядер используемых для выполнения задач класса *(a)*

"*рабочая очередь*" (WQ) – метаочередь, она определяется для каждой возможной комбинации вида деятельности (потока заданий) и вычислительного ресурса, т.о. общее число очередей составит $n \times m$ Такое разделение и гранулярность необходимы, при различных требованиях к использованию вычислительных ресурсов, например, при использовании одной и той же очереди задачей SCORE с низким приоритетом и задачей MCORE высокого приоритета приведет к тому, что выполнение задачи SCORE будет ждать окончания выполнения задачи MCORE, даже, если есть свободные ресурсы SCORE. С другой стороны, важно различать задачи по виду деятельности (это объясняет необходимость иметь двухмерность для очередей);

"очередь выполнения заданий" WMS (RQ) - ресурсно-ориентированная очередь, существуют две возможности, как RQ могут быть определены, (1) - очередь

для каждого типа ресурса для каждого ВЦ, например XYZ_SCORE, XYZ_HIMEM, и др. (для центра XYZ), (2) - определить единую очередь XYZ_RQ, которая может быть использована для задач с требованиями любого класса ресурсов.

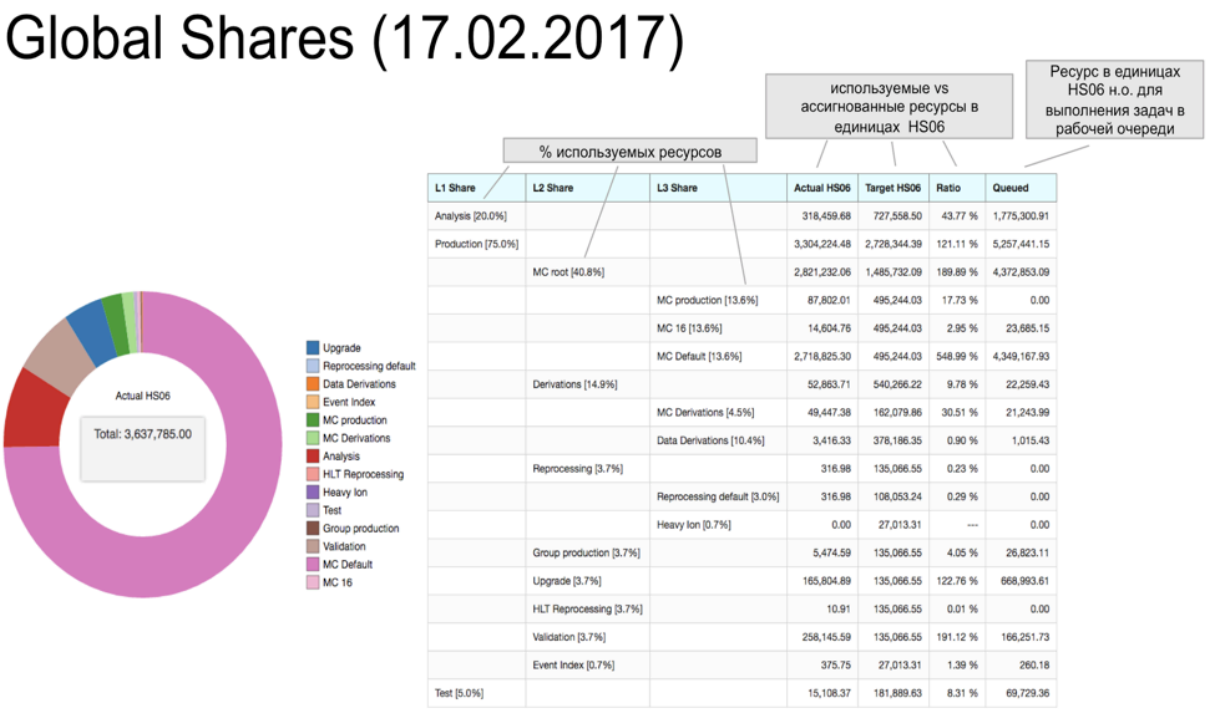
Таким образом, WMS динамически определяет количество задач необходимых для выполнения каждого задания, и "направляет" задачи в соответствующие центры. Количество задач в задании для каждой очереди происходит пока не будет выполнено следующее условие : $\sum_S Q_{a,r}^S > 2 \times \sum_S R_{a,r}^S$, и как результат : $Q^S \simeq 2 \times R^S$, и "направление" задач пилоту для выполнения на вычислительном ресурсе определяется уравнением : $C = (\sum_S R^S) \times \vec{1} \sim A$ Расхождение между A и C рассчитывается для каждого направления задачи пилоту (т.е. для выполнения на рабочем узле) для всех типов потоков заданий, если они имеют одну или более выполняемых задач в RQ, после чего выбирается задача для того потока, для которого существует наибольшая разница и число используемых ядер меньше зарезервированного. Т.е. выбирается задача потока заданий соответствующего индексу минимального ненулевого элемента:

$$D = (C \odot ((Q^S \times \vec{1}) > \vec{1})) \oslash A.$$

Данная модель соответствует реальным потребностям физического эксперимента, и случаи, когда, задания типа MC требуют только ресурсы класса MCORE, а задания типа "обработка данных" требуют только ресурсы класса SCORE, не рассматривается как не имеющие реального применения. Реализация данной модели для экспериментов ATLAS и COMPASS получила название "global shares", и была успешно реализована для обработки данных. На рисунке 54 представлена страница с информацией подсистемы мониторирования системы управления потоками заданий, показывающая распределение ресурсов между различными потоками заданий, согласно приоритетам определенным физической программой эксперимента.

На рисунке 55 представлен график, показывающий время ожидания в минутах до начала выполнения заданий анализа (светло-зеленый столбцы) и заданий

“производства данных” (фиолетовые столбцы). Как видно из графика среднее время ожидания для заданий анализа составляет менее трех часов после введения методики динамического распределения ресурса (по сравнению с пятью часами (рисунок 51)).



L1 Share	L2 Share	L3 Share	Actual HS06	Target HS06	Ratio	Queued
Analysis [20.0%]			318,459.68	727,558.50	43.77 %	1,775,300.91
Production [75.0%]			3,304,224.48	2,728,344.39	121.11 %	5,257,441.15
	MC root [40.8%]		2,821,232.06	1,485,732.09	189.89 %	4,372,853.09
		MC production [13.6%]	87,802.01	495,244.03	17.73 %	0.00
		MC 16 [13.6%]	14,604.76	495,244.03	2.95 %	23,685.15
		MC Default [13.6%]	2,718,825.30	495,244.03	548.99 %	4,349,167.93
	Derivations [14.9%]		52,863.71	540,266.22	9.78 %	22,259.43
		MC Derivations [4.5%]	49,447.38	162,079.86	30.51 %	21,243.99
		Data Derivations [10.4%]	3,416.33	378,186.35	0.90 %	1,015.43
	Reprocessing [3.7%]		316.98	135,066.55	0.23 %	0.00
		Reprocessing default [3.0%]	316.98	108,053.24	0.29 %	0.00
		Heavy Ion [0.7%]	0.00	27,013.31	---	0.00
	Group production [3.7%]		5,474.59	135,066.55	4.05 %	26,823.11
	Upgrade [3.7%]		165,804.89	135,066.55	122.76 %	668,993.61
	HLT Reprocessing [3.7%]		10.91	135,066.55	0.01 %	0.00
	Validation [3.7%]		258,145.59	135,066.55	191.12 %	166,251.73
	Event Index [0.7%]		375.75	27,013.31	1.39 %	260.18
Test [5.0%]			15,108.37	181,889.63	8.31 %	69,729.36

Рисунок 54 - Мониторирование распределения вычислительного ресурса

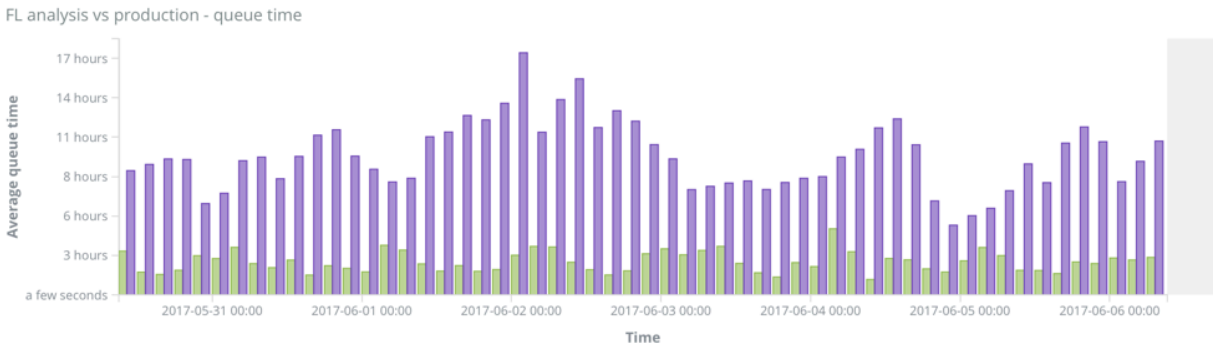


Рисунок 55 - Распределение времени ожидания для T1 и T2 центров после введения методики динамического распределения вычислительного ресурса

3.7 Создание системы обработки, моделирования и анализа данных эксперимента ATLAS

Эксперимент и международное сотрудничество ATLAS. ATLAS (от англ. A Toroidal LHC ApparatuS) является одной из двух установок общего назначения (вторым таким детектором является установка CMS). В 2013 году ученые ATLAS и CMS открыли, предсказанную в 1964 году частицу - бозон Хиггса. Научная программа ATLAS имеет следующие основные направления исследований:

- использование бозона Хиггса как инструмент для новых открытий;
- поиск темной материи;
- поиск новой физики частиц, взаимодействий, физических законов.

Научное сообщество ATLAS включает более 3000 ученых из 40 стран. ATLAS – самая большая в мире научная установка в области физики высоких энергий и ядерной физики. Она размещена в шахте на глубине 100 метров и имеет около 45 метров в длину, более 25 метров в высоту (высота семиэтажного дома), её вес составляет около 7000 тонн (общий вид установки и ее основные компоненты приведены на рисунке 56). Основными детекторами установки являются: внутренний детектор, электромагнитный и адронный калориметры, мюонный спектрометр. Установка имеет более 150 миллионов каналов считывания. Поток информации с установки составляет 1 Пбайт/сек, отбор интересных событий для последующей обработки и анализа производится трехуровневой системой триггера на вход триггера первого уровня события поступают с частотой 40 МГц, на последней стадии отбора (триггер высокого уровня) для дальнейшей обработки и анализа отбирается примерно 1000 событий в секунду.

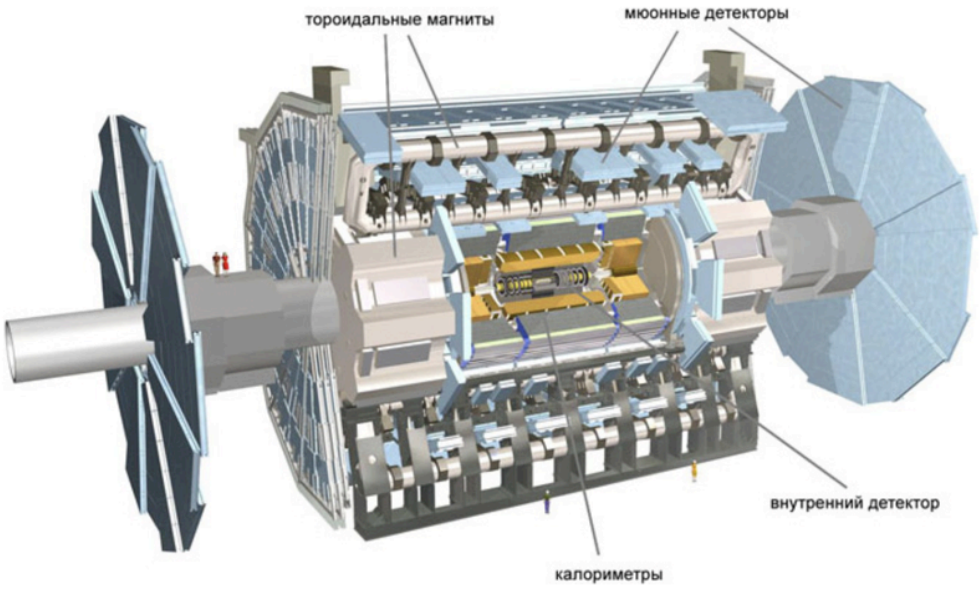


Рисунок 56 - Общий вид детектора ALTAS

Размер “сырого” события составляет 1.5 Мбайта, в 2016 году было набрано O(10^7) событий, и одновременно смоделировано 4 миллиарда событий. Физические явления, исследуемые в эксперименте, представляют собой очень редкие физические процессы, на рисунке 57 представлен график сечений (по оси ординат в логарифмическом масштабе), из графика видно, что 10^{11} столкновений необходимы для того, чтобы найти среди них 10 бозонов Хиггс.

Вероятность растет логарифмически с ростом энергии (ось ординат). Теоретические предсказания (линии) хорошо согласуются с измерениями (маркеры). Общий (все форматы для моделируемых и реальных данных) объем управляемых данных ATLAS на начало 2017 года составил 260 Пбайт, и представлен на рисунке 58, из графика видна корреляция роста объема данных с периодами работы коллайдера: первая фаза работы (2010/2013 годы), вторая фаза работы (2015/2017 годы), для сравнения все письменное наследие человечества на всех языках мира составляет 50 Петабайт.

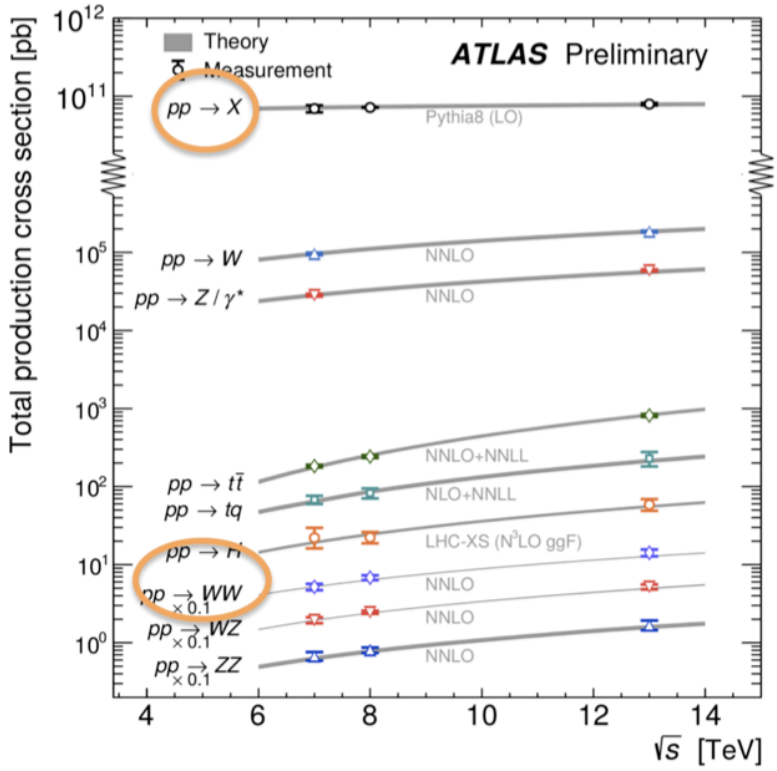


Рисунок 57 - График сечений (протон/протонные столкновения на LHC)

Рисунок 58 - Объем управляемых данных эксперимента ATLAS

В таблице 3 приведены объемы информации для различных групп в эпоху “Больших данных” ([73, 74]), и если компьютерные мощности WLCG значительно уступают имеющимся мощностям гигантов ИТ индустрии (так Amazon имеет более

40М ЦПУ-ядер, Google имеет ~1М серверов, т.е. примерно 20М ядер), то по объемам информации ФВЭ и ЯФ является заметным игроком на “поле BigData”. Поэтому задача создания систем обработки и анализа данных для физических установок класса мегасайенс привлекают интерес со стороны многих ИТ компаний. Рассмотрим как была реализована глобальная система распределенной обработки данных эксперимента ATLAS.

Таблица 3 – Объемы информации для различных групп в эпоху Больших данных

Источник информации	Размер в ед. измерения	Комментарий	Объем информации
Библиотека конгресса США			~200 Тбайт
Одно электронное сообщение (e-mail)	~1 Кбайт	В год 30 триллионов писем, без учета спам рассылки	30Пбайт*N копий
Электронная фотография	~2Мбайт	500 миллиардов фотографий в год, 25 миллиардов фотографий в Facebook	1 Эбайт
LHC	~2Мбайта/ событие	4 эксперимента, “сырые” и приведенные события	700 Пбайт
WWW		25 миллиардов страниц, 1 триллион документов	~1 Эбайт
Youtube		Ежегодно	15 Пбайт
Blue ray диски	~25Гбайт	Ежегодно 100М штук	2.5 Эбайт

3.7.1 Система обработки, моделирования и анализа данных эксперимента ATLAS

Для моделирования, обработки и анализа данных экспериментов класса мегасайенс требуется слаженная работа гетерогенных вычислительных ресурсов. В частности, эксперимент ATLAS использует ресурсы 250 ВЦ по всему миру, а также мощности суперкомпьютерных центров, национальные, академические и коммерческие ресурсы “облачных вычислений”. Разработанные методы и описанные выше подходы позволили создать систему обработки и анализа данных эксперимента ATLAS на LHC. Система была успешно реализована для управления вычислительными ресурсами ATLAS в конце 2013 года, а после тонкой настройки, в начале 2014 года она была принята, как основная система управления заданиями эксперимента. Система получила название ProdSys2-megaPanDA. (ProdSys2 - от англ. Production System 2 поколения, megaPanDA - Production and Distributed Analysis for megascience). Все модули системы управления потоками заданий были отлажены и протестированы на реальных задачах эксперимента, дополнительная проверка и отладка были проведены для экспериментов ALICE и COMPASS. Эксперимент ALICE использует данную систему для управления потоками заданий на суперкомпьютере Titan, эксперимент COMPASS использует данную систему для обработки данных в ЦЕРН и на суперкомпьютерных мощностях университета Иллинойс Урбана Шампань.

ProdSys2 - является необходимым уровнем абстракции, скрывающим от пользователя сложности работы, связанные с запуском и выполнением задач на рабочем узле. В целом работу системы обработки и анализа данных можно описать как процесс преобразования входных, обычно не структурированных данных эксперимента (или пользователя), в набор параметров для выполнения задач на вычислительных системах. Система имеет три уровня (DEFT, JEDI, PanDA) и

отвечает всем требованиям, предъявляемым к системам управления загрузкой, подробно рассмотренных в главе 3.

Система управляет всеми потоками заданий физического эксперимента ATLAS:

- потоки заданий, выполняемые WMS для всего эксперимента (“виртуальной организации”);
    - обработка и (пере)обработка данных;
    - моделирование методом Монте-Карло;
    - создание приведенных данных для физического анализа;
    - обработка данных для триггера высокого уровня;
    - специальные случаи, например “обработка поездом” для всех или нескольких физических групп;
- потоки заданий, выполняемые 20 физическими группами эксперимента
    - создание данных для физического анализа, проводимого физической группой;
    - анализ данных;
- потоки заданий, выполняемые отдельными пользователями;
    - физический анализ данных;

В системе реализованы “обработка поездом” и “постоянная обработка” (рассмотренные ранее), а также специальные потоки заданий для проверки кода и валидации физических результатов. Специальным случаем является поток заданий *‘event index*’, когда для каждого обработанного события в базу данных записывается краткая информация о нем. В системе реализована система “справедливых долей” и “всемирного облака”, система имеет подсистему мониторирования с большим и разнообразным набором меню для контроля, управления и мониторирования потоками заданий.

Ниже приведены графики, демонстрирующие работу и производительность системы. На рисунке 59 показано количество задач для различных потоков заданий, выполняющих одновременно. Из графика наглядно видно, что система выполняет до 300 тысяч задач в день одновременно.

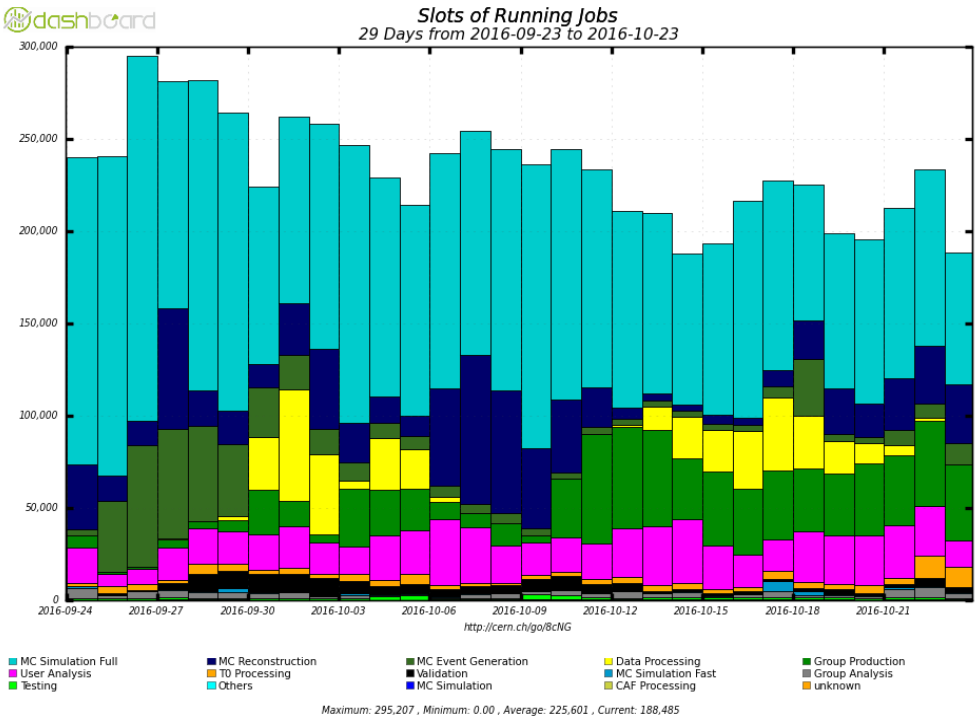


Рисунок 59 - Количество задач для различных потоков заданий, выполняющихся одновременно

На рисунке 60 показано количество задач, выполненных ежедневно с января 2016 года по февраль 2017 года. Из графика следует, что системой выполняется до 2M задач в день, и среднее значение, выполняемых задач значительно превосходит 1M/день (это очень важный показатель масштабируемости системы). График дает представление о разделении ресурса между различными потоками заданий.

На рисунке 61 приведено количество данных, обработанных системой в течение одного месяца, а рисунок 62 дает представление о том, какие вычислительные ресурсы были использованы (из графа видно, что очереди выполнения были определены для центров грид (T1, T2), суперкомпьютеров (Titan) и ресурса облачных вычислений ЦЕРН (CERN_P1_DYNAMIC_MCORE), а также то, что управление потоками осуществлялось для ресурсов разных типов : LOMEM (< 2

Гбайт памяти на ядро), MCORE (multi-core, задача использует несколько ядер одновременно), SHORT (время выполнения задачи ограничено 24 часами) и т.д.

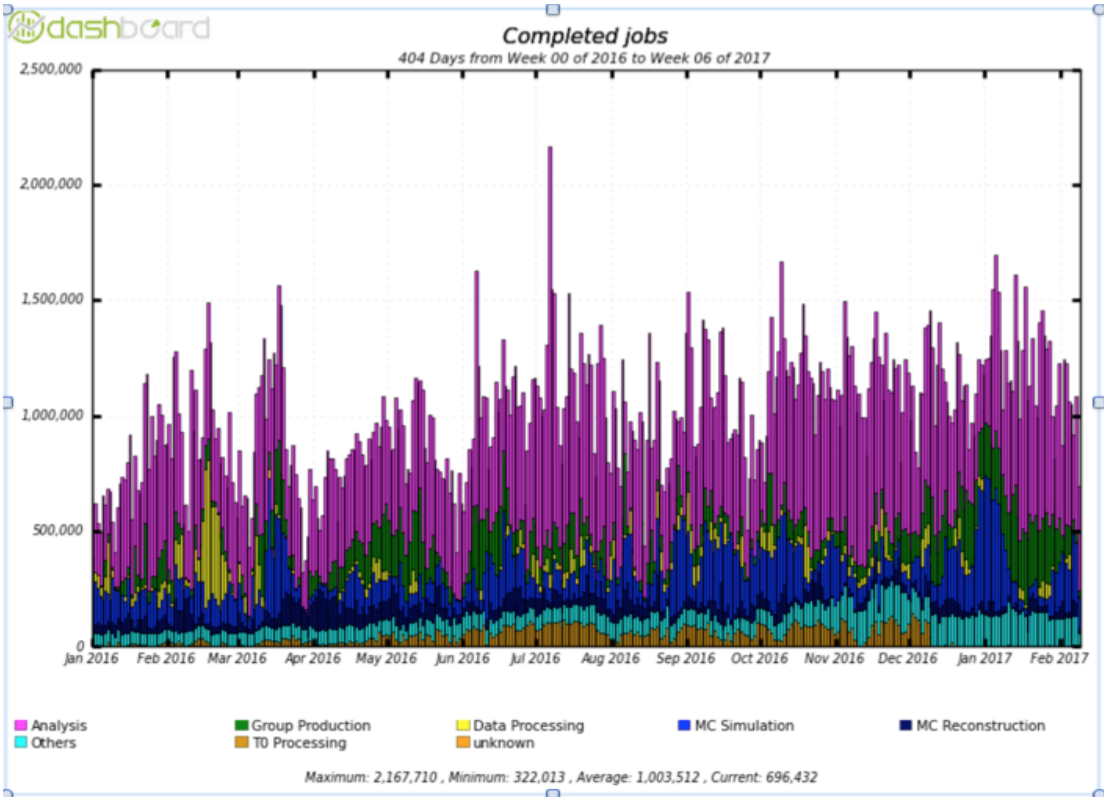


Рисунок 60 - Количество задач для различных потоков заданий, выполненных ежедневно

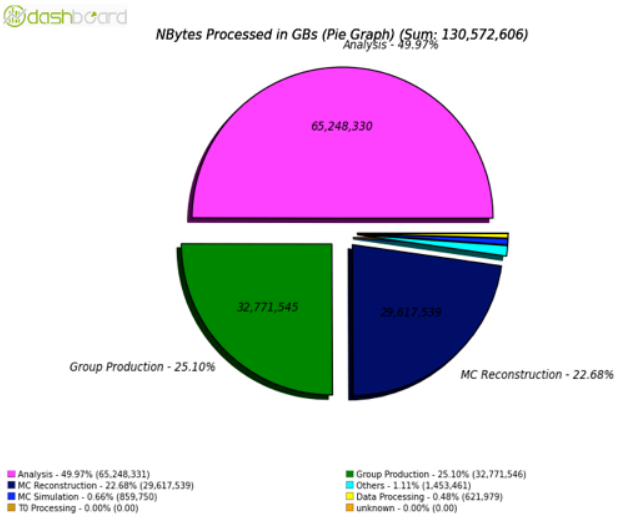


Рисунок 61- Количество данных обработанных в системе в течение месяца

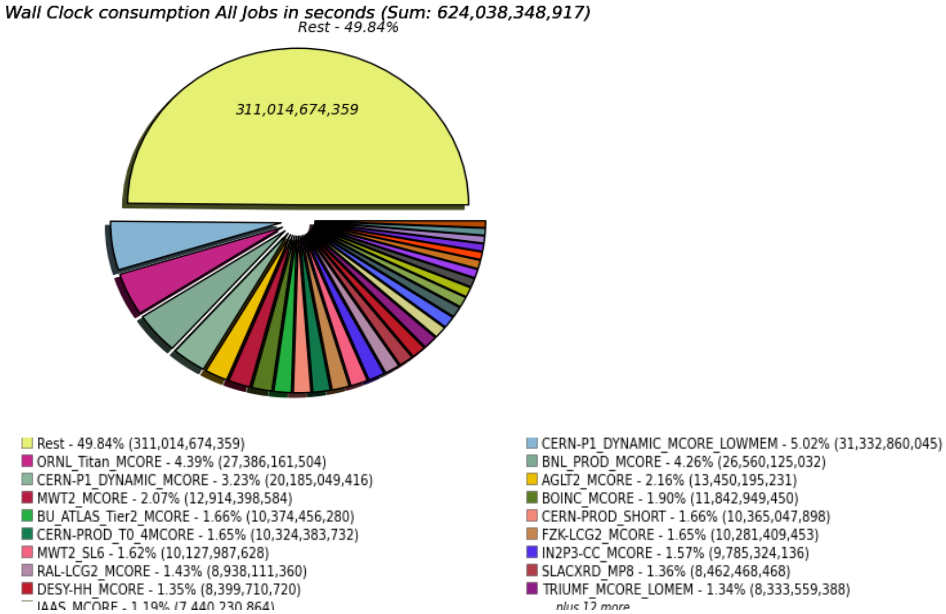


Рисунок 62 - Используемые вычислительные ресурсы

Созданная система является уникальной по своим параметрам и не имеет мировых аналогов. В эксперименте ATLAS система управляет до 2M вычислительных задач в день в гетерогенной компьютерной среде, состоящей из более чем 250 ВЦ, включая ресурсы облачных вычислений и суперкомпьютеры.

Система рассматривается как основной вариант программного обеспечения для распределенной обработки данных и управления загрузкой для экспериментов на коллайдере NICA (ОИЯИ, Дубна) и экспериментом по поиску темной материи (DESC) в проекте LSST, система используется в экспериментах COMPASS на ускорителе SPS (Super Proton Synchrotron) в ЦЕРН.

3.8 Создание подсистемы мониторирования для системы распределенной обработки данных эксперимента ATLAS. Архитектурные принципы, методы и технологии при реализации подсистем мониторирования для систем управления загрузкой

Система управления загрузкой в распределенной гетерогенной компьютерной среде представляет собой сложный и неоднородный комплекс аппаратных и программных средств. Система взаимодействует с другими системами, сервисами и базами данных распределенной инфраструктуры:а) системой управления данными, б) информационной системой, в) фабрикой пилотных заданий и многими другими (рисунок 41). Одним из основных назначений подсистемы мониторирования (система мониторинга) научного эксперимента является отслеживание ошибок при выполнении задач обработки и анализа данных. В эксперименте ATLAS аппаратные ошибки встречаются примерно в 11% от общего числа выполненных задач (рисунок 63). Из-за больших объемов метаинформации, связанной с каждой задачей, поиск ошибок превращается в достаточно ресурсоемкую процедуру. Ежедневное выполнение миллиона и более задач требует высокого уровня автоматизации, постоянного мониторирования, сбора и обработки больших объемов информации о параметрах работы системы, возникающих

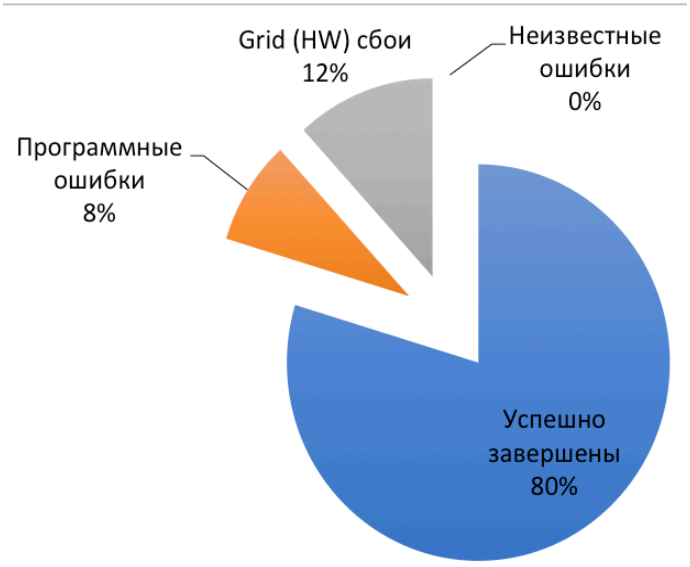


Рисунок 63 - Процент ошибок при выполнении задач в грид инфраструктуре

ошибках, обнаружения и по возможности предсказания аномалий в работе системы управления загрузкой.

Разработка системы мониторирования являлось сложной и комплексной задачей. Необходимо было ответить на следующие вопросы:

- какие задачи должна решать система мониторирования;
- какие ресурсы она должна контролировать;
- кто является пользователем системы;
- время жизни информации (и метаинформации) в системе и объемы хранимой информации;
- метод мониторинга: централизованный или децентрализованный. В первом случае сбор и интеграция информации происходит в одном узле (сервер системы мониторирования), в случае децентрализованного подхода, не происходит агрегации и интеграции информации в одном месте и допускается наличие автономных агентов самостоятельно собирающих информацию и взаимодействующих с сервером мониторинга используя push- или pull-модель доставки информации.

Система мониторинга должна не только давать представление о текущем состоянии и ходе выполнения потоков заданий, но и включать в себя функции аккаунтинга (от англ. accounting), для учета используемого ресурса. Метриками использования ресурса являются процессорное и астрономическое время, дисковое пространство, количество и классы, выполненных задач, информация о состоянии очередей и ВЦ. Система должна давать, как общее представление о состоянии обработки данных физического эксперимента, так и позволять проверить состояние и/или ход выполнения отдельной задачи.

Основные требования к системе мониторирования. Перечислим основные требования, предъявляемые к системе мониторирования :

- универсальность. Одна и та же система должна предоставлять информацию обо всех потоках заданий для обработки данных, выполняемых в рамках физического эксперимента;
- производительность. Производительность системы должна позволять мониторинг всех компонент системы управления загрузкой в реальном времени, допустимое время отклика должно составлять секунды;
- модульность. Уровни доступа к данным и визуализации данных должны быть разделены. При введении новых типов потоков обработки данных, новых вычислительных ресурсов (например, добровольных ресурсов суперкомпьютерных центров) система должна иметь возможность быть расширена без потери производительности и универсальности системы. Система должна позволять добавления новых функциональных возможностей и сбор информации от новых источников информации;
- масштабируемость. Время отклика системы мониторинга не должно деградировать при росте информации;
- доступность и стандартизация доступа к информации. Система должна иметъ развитые веб-сервисы, пользовательский интерфейс и интерфейс программирования приложений (API). В API должны быть реализованы основные сервисы для мониторинга и управления объектами системы. Соответствующий веб-сервис должны обеспечивать аутентификацию с помощью специальных цифровых ключей для разграничения доступа к системе и журналирования действий пользователей. Система должна иметь механизм отложенного выполнения запросов;
- целостность. Система должна предоставлять общую картину управления потоком заданий, а также детальную информацию о выполнении каждого запроса, задания, задачи, с возможностью исследования причины сбоя;
- анализ информации и автоматизация. Система должна иметь высокий уровень автоматизации и возможности анализировать имеющуюся

информацию, в частности ошибки в ее работе, что позволило бы сформулировать правила и применить, например, алгоритмы “машинного обучения” для обнаружения аномалий и/или сбоев в работе системы управления загрузкой.

- мобильность. Выбор используемых технологий таким образом, что система может быть использована различными “виртуальными организациями”;

Уровни и функции системы мониторинга. Система для распределенной обработки данных - является сложным объектом, пользователями системы являются несколько различных групп

- “виртуальная организация”:
  - “виртуальная организация” в целом (физический эксперимент), физические группы, работающие над отдельными темами научной программы эксперимента, физические группы в Университетах и Лабораториях, входящих в эксперимент, отдельные физики;
  - сотрудники ВО, отвечающие за каждодневную работу систем распределенного компьютинга (включая системы управления данными и потоками заданий);
  - сменные “операторы” и “службы поддержки” пользователей. В зависимости от размера и сложности физического эксперимента операторская служба может функционировать в режиме 24/7 в периоды работы ускорителя и основных кампаний по (пере)обработке или моделированию данных, и в режиме 8/5 в остальное время;
- компьютерные специалисты, системные администраторы и службы эксплуатации ВЦ, предоставляющих компьютерные ресурсы физическому эксперименту;

- финансирующие организации. Как правило, для данной группы наиболее интересным является информация об использовании вычислительного ресурса (т.е. аккаунтинг).

Можно выделить следующие уровня системы мониторинга :

- мониторинг работы вычислительных ресурсов :
    - стабильность работы, ошибки выполнения задач, типы ошибок (одним из типичных примеров может служить сбои в системе хранения данных);
    - производительность вычислительных центров. Количество выполняемых заданий и задач, их типы. Количество обработанных событий; Количество полученных и/или переданных для/после обработки данных;
- мониторинг вычислительной инфраструктуры системы управления потоками заданий, включая коммуникацию с внешними системами: системой управления данными, ИС и др.
    - Сбой в работе системы управления данными, приводит к остановке в определении новых потоков заданий и задержке в выполнении текущих заданий;
- мониторинг потоков заданий :
    - по группам пользователей:
        - всего эксперимента в целом;
        - отдельных физических групп;
        - отдельных ученых;
    - по классам потоков заданий :
        - обработка данных;
        - анализ данных;
        - Монте-Карло моделирование;

- …
- по выделенным квотам, долям, приоритетам для различных классов заданий;

- мониторинг хода исполнения цепочки заданий и/или запросов (например, запрос может быть одобрен / отвергнут или его исполнение может быть отложено лицом отвечающим за физическую программу эксперимента)
- мониторинг работы групп сайтов, выбранных системой управления потоками заданий для выполнения запроса/задания/цепочки заданий;

3.8.1 Реализация подсистемы мониторинга для системы megaPanDA эксперимента ATLAS на Большом адронном коллайдере и за его пределами

Основной задачей стала быстрое обнаружение ошибок и мониторирование хода выполнения задач для различных классов потоков данных под управлением системы megaPanDA. Одной из первых задач, которую необходимо было решить это популярность данных в зависимости от времени. За время жизни эксперимента ATLAS общий архив информации о выполненных задачах, заданиях и запросах содержал информацию о ходе выполнения более чем 10M заданий и около 3000M задач, включая информацию о месте выполнения, ПО, ошибках, количестве повторов задач и т.д. Необходимо было решить задачу о способе хранения данных и метаданных, обеспечить доступность к ним для приложений аналитического анализа информации и/или экспертных систем, одновременно обеспечив бесперебойную работу системы мониторирования. *Вся информация была разделена на 3 группы* :

- Текущая информация. В эту группу вошла информация о задачах и заданиях выполняемых в системе, ждущих выполнения и выполненных в течение последних трех месяцев. Для информации этой группы характерны запросы на модификацию содержимого, как изменения приоритета задания/задачи, превентивная остановка и/или перезапуск задачи.

- Среднесрочная информация. Информация о работе в период от 3х то 6 месяцев. Задачи этой группы часто составляют часть выполняемой цепочки заданий и доступ к ним в режиме RO (от англ. read only) происходит до 100 раз в день.
- Архив системы. Информация о потоках заданий (включая всю информацию об отдельных задачах) выполненных 6 месяцев назад или раньше.

Выбор шага в 3 месяца был сделан на основе количества запросов к информации, хранящейся в БД. Так все что касалось выполнения заданий старше этого периода использовалось только для проведения аналитических исследований по производительности работы системы управления загрузкой, классификации типов сбоев или для составления отчетов об использовании вычислительных ресурсов. Текущая информация, наоборот пользовалась наибольшей популярностью у всех пользователей. Все пользователи системы по своим “поведенческим” характеристикам были разделены на 4 группы :

- “системные администраторы ”. Компьютерные специалисты и системные администраторы ВЦ, предоставляющих вычислительные мощности;
- ”операторы”. Сменные операторы и служба поддержки первого уровня, следящие за выполнением потоков заданий;
- “физики”, участники международного сотрудничества ATLAS проводящие анализ данных, и запускающие задания для их анализа;
- “менеджеры”, участники международного сотрудничества ATLAS, формирующие потоки для различных классов заданий :
  - (пере)обработка данных;
  - Монте-Карло моделирование;
  - потоки заданий для физических групп;
  - обработка данных для триггера высокого уровня;

- проверка и валидация базового ПО эксперимента;

– “координаторы”. Участники международного сотрудничества ATLAS, координирующие физическую программу эксперимента или проекты в SW&C;

- эта группа, также отвечает за предоставление регулярных отчетов для проверяющих и финансирующих организаций;

– “эксперты”. Разработчики ПО системы управления загрузкой, базы данных, исследующие производительность ПО и ошибки в его исполнении и/или в случае возникновения аномалий в работе всей системы.

Такое распределение дало ответ на один из фундаментальных вопросов при создании системы мониторинга: “Кто является пользователем системы?”. А также потребовало создания такого набора функций, который позволил иметь как страницы, дающие представление о работе системы в целом и глобальном использовании вычислительного ресурса, так и детальной информации о выполнении задачи “физика”. Анализ журнальной информации о доступе к системе показывает, что наиболее быстро нужную информацию получают представители грипп “эксперты” и “операторы”, скорость получения информации представителями других группа существенно зависит от опыта работы с системой, поэтому был выбран подход интуитивного поиска, а также “моя любимая страница”, когда система “запоминала” предпочтения пользователя и на первом этапе работы предлагала выбор из наиболее посещаемых страниц.

Основными принципами в реализации были производительность, надежность, простота реализации. Была выбрана централизованная модель. Рассмотрим некоторые из выбранных технических решений.

Выбор технологии для хранения и представления информации. Разделение информации по временной шкале давало возможность реализовать хранение данных с выбором различных технологий. Хранилище могло быть как гомогенным (СУБД)

так и гетерогенным (комбинация СУБД_NoSQL). Сравнение возможных решений для технологий NoSQL подробно обсуждались на рабочем совещании в НИЦ “Курчатовский институт” “Big Data processing and analysis challenges in mega-science experiments” [75] с участием экспертов из ЦЕРН, НИЦ КИ, ДЕЗИ и ОИЯИ. Совещанию предшествовала интенсивная проверка различных технологий NoSQL и выбора наилучшего решения для технологии базы данных megaPanDA, и одновременно были проведены исследования по внутреннему разделению таблиц СУБД ORACLE для хранения информации. Были рассмотрены вопросы масштабируемости и производительности и было показано, что при контроле за размером “текущей компоненты” информации в 400 Гбайт, и общем размере информации менее 100 Тбайт (в настоящий момент размер базы данных составляет 20 ТБ) использование СУБД ORACLE дает лучшие характеристики при доступе к информации, при этом классы индексов и организация таблиц может быть различной для трех групп, например, ‘архив’ хранится в “сжатом” виде и имеет внутреннее разделение по годам. Важным фактором выбора технологии SQL vs NoSQL служил фактор “мобильности”, в предположении, что вся система управления загрузкой и ее мониторинг будет в будущем использоваться за пределами ATLAS, а также ФВЭ и ЯФ, и использование СУБД ORACLE позволит другим экспериментам использовать аналогичную технологию, например, MySQL (следует признать, что это решение оказалось правильным и эксперименты AMS и LSST используют именно его). Выбор технологии ORACLE (а не MySQL) также диктовался рещением группы разработчиков системы управления данными эксперимента и отделением ИТ ЦЕРН в выборе этой технологии и получения лицензии на ее использование безвозмездно для всех университетов и лабораторий, входящих в ATLAS (подробно исследование вопроса о сравнении технологий NoSQL отражено в работе, написанной в соавторстве с сотрудниками Лаборатории “технологии Больших данных” НИЦ КИ [76]). Выбор остальных технологий был менее драматичен и были использованы технологии, широко применяемые для

представления и визуализации информации в ИТ компаниях, с объемами информации и типом доступа, сравнимыми с требованиями к системе мониторинга (Instagram, Mozilla, Youtube, Google). Технология фреймворка django для веб-приложений и шаблон проектирования MVC (Model View Controller) [77]. Алгоритмы обработки информации были разделены между уровнями frontend и базой данных (используя большой набор функций и возможностей ORACLE), генерация графиков выполняется на стороне клиента с использованием библиотек ds.j3 (data driven documents) [78], технология ajax используется для обмена данными между броузером и веб-сервером. Для аутентификации пользователей был использован пакет SSO (Single Sign-On Management), предоставляемый ЦЕРН. Схематично архитектура и основные компоненты инфраструктуры системы мониторинга представлена на рисунке 64.

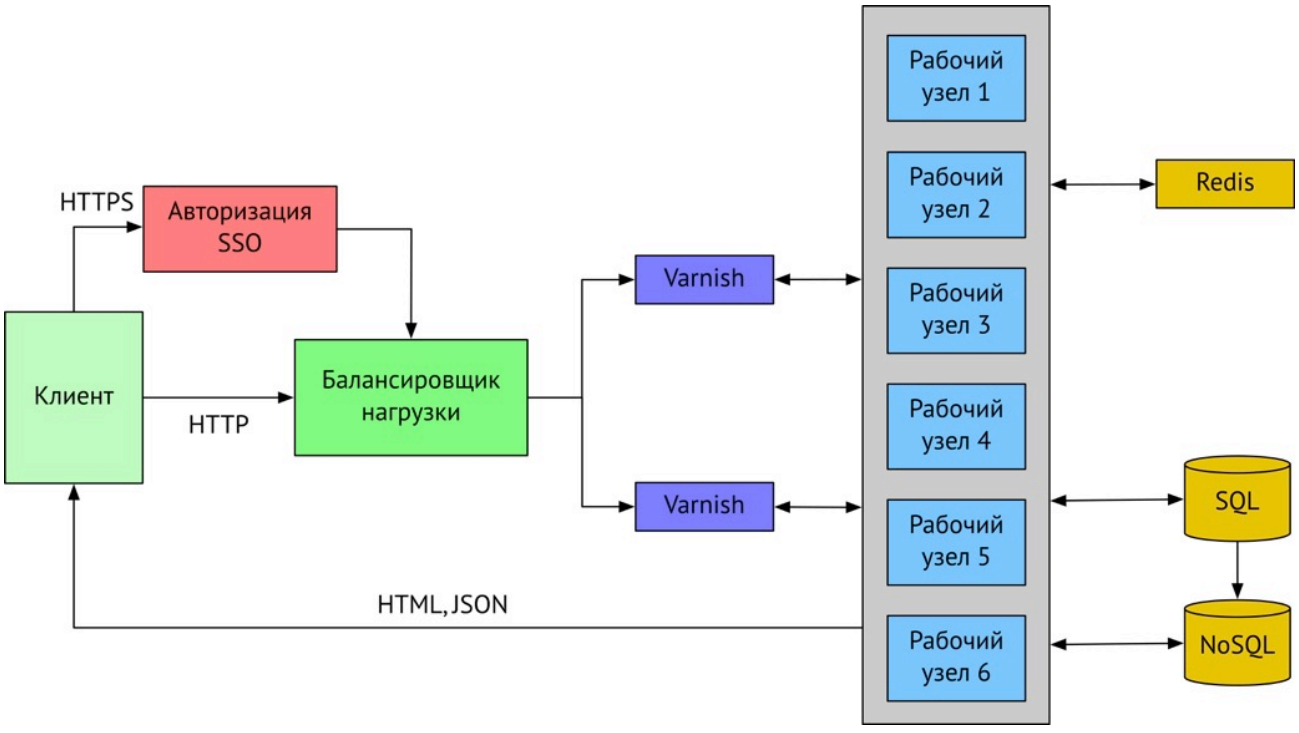


Рисунок 64 - Архитектура и основные компоненты инфраструктуры системы мониторинга

Созданная система отвечает всем требованиям и имеет высокую производительность, так генерация информации о выполнении задач системой

управления загрузкой (с информацией об ошибках) за последние 12 часов и для O(1M) задач занимает 10 сек [79, 80]. На рисунке 65 представлена статистика обращений к системе мониторирования за последние 6 месяцев 2016 года.

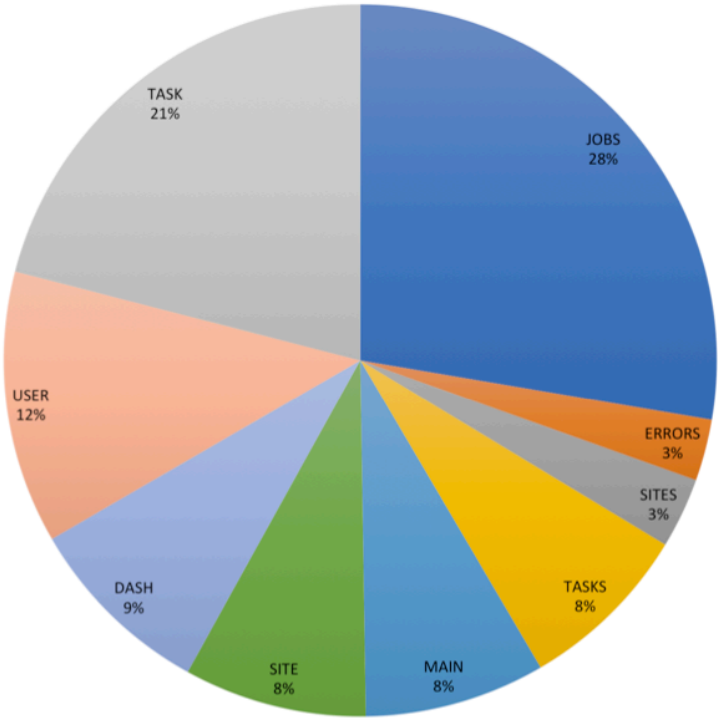


Рисунок 65 - Статистика обращений к системе мониторирования за последние 6 месяцев 2016 года

Количество ежедневных обращений составляет в среднем 13 тысяч, от 1500 индивидуальных пользователей, из приведенной статистики видно, что наиболее популярной является информация о заданиях (группах заданий: task, tasks) и задачах (jobs). Многие иллюстрации данной диссертации выполнены с использованием графических возможностей системы мониторинга. В данном разделе была подробно рассмотрена реализация подсистемы мониторинга эксперимента ATLAS. Как отмечалось ранее, мобильность является одним из требований к такой системе. Выбор технологий позволил применить данную систему для эксперимента LSST, для установки аппаратной части были использованы ресурсы Amazon EC2, а подсистема мониторинга для системы управления загрузкой, развернутой в НИЦ “Курчатовский институт” (приложения бионформатики) находится в ЦОД НИЦ КИ.

Далее мы кратко рассмотрим как информация и предсказание о времени выполнения заданий в WMS может быть интегрирована с системой мониторинга.

3.8.2 Подсистемы мониторинга системы управления заданий megaPanDA и оценка времени выполнения заданий в гетерогенной компьютерной среде

В предыдущем разделе была обоснована концепция разделения информации на 3 уровня: “текущая”, “среднесрочности” и “архив”. В базе данных хранится информация о сотнях миллионов выполненных заданий и задач. Анализ этой информации дает понимание о:

- различиях в производительности для классов потоков заданий;
- различиях в производительности для центров грид;
- различиях в производительности при различных методах доступа к данным;
- поведении пользователей. Эта информация может быть использована для оптимизации работы системы управления загрузкой, обучения самих пользователей, создание экспертных систем,...

Эти и другие аргументы привели к развитию аналитических платформ, используемых в ФВЭ и ЯФ наряду с классическими системами мониторинга. Кратко рассмотрим какая информация может быть использована для предсказания времени выполнения заданий в системе (TTC, от англ. Task Time to Complete) и насколько это важно для мониторирования работы системы для обработки данных в целом.

Помимо потребления конкретных ресурсов (таких как время ЦПУ, оперативная память и дисковый ресурс) задания могут потребовать дополнительных возможностей для своего выполнения. Например, передача данных между заданиями, неявно описанная в потоке заданий, потребует соответствующих возможностей и соответственно изменения порядка выполнения. Условие, «задание А выполняется одновременно с заданием Б» или «задание А выполняется после задания Б» потребует определенных конкретных изменений в исполнении, зависящих от времени. В таблице 4 перечислены некоторые из дополнительных

возможностей, которые надо принимать во внимание и от которых зависит время исполнения задания (и цепочки заданий).

Таблица 4 – Параметры выполнения заданий

Требование к выполняемому заданию	Пример
Данные	Результат выполнения задания А, является входными данными для задания Б (цепочка Монте-Карло заданий или задания физических групп, ожидающие данные в формате AOD)
Время начала исполнения	1. Как можно скорее 2. В 14:00 после окончания калибровки
Время окончания и/или максимальное время на выполнение	1. Задание должно быть завершено до 17:00 2. Задание должно быть выполнено за 2 часа
Порядок выполнения	Задания А и Б могут выполняться одновременно Задание С должно выполняться после окончания заданий D и E

Требования, перечисленные в таблице 4, могут быть различной степени детализации, чем требования, предъявляемые к вычислительным ресурсам, но всегда должны учитываться (в противном случае поток заданий не может быть выполнен). Например, зависимость от данных может подразумевать упорядочение и передачу данных. Ограничение времени выполнения задания или предельное время на выполнение могут ограничить выбор вычислительных ресурсов. Соответствующие сопоставления с инфраструктурными возможностями могут быть нетривиальными, но в то же время предоставляют возможности для адаптации и оптимизации работы

системы управления загрузкой. Важным шагом в для расширения контроля и анализа работы системы управления загрузкой явилось предсказание времени выполнения задания.

Модель предсказания времени выполнения заданий и ее интеграция с подсистемой мониторинга системы управления потоками заданий. На первом этапе была создана *холодная* и *теплая* модель оценки времени выполнения заданий (названия наследовали историю создания термодинамической модели и успеха ее применения (раздел 1.4.2)) В *холодной* модели время выполнения задания оценивается на основе истории выполнения заданий данного класса, а в "*теплой*" была введена коррекция после выполнения первых 10 задач в задании, т.н. задач-скаутов (классификация потоков заданий была рассмотрена в разделе 3.1). При построении модели и введения точности в предсказании в выборке данных для разделения заданий по меньшим группам использовалась мета-информация, хранимая в базах данных DEFT и JEDI. Для создания обучающей выборки использовалось 27 атрибутов для заданий, приведем список основных из них:

- *project* - информация о данных (реальные данные или моделируемые), информация об энергии пучков коллайдера, информация о режиме работы коллайдера;
- *format* - информация о формате данных;
- *step* - шаг обработки данных;
- *group* - эксперимент, физическая группа или пользователь, запросившие выполнение задания.
- *SW* - версия ПО, которая используется для обработки данных;
- *nucleus* - сайт-ядро во "всемирном облаке", на котором будет выполняться задание;
- *queue* - очередь, в которую было поставлено задание.
- *priority* - приоритет задания;

- *corecount* - количество процессорных ядер, фактически используемых задачами, составляющими задание. Доступно только для “тёплой” модели.
- *ramcount* - количество оперативной памяти, фактически используемых задачами, составляющими задание. Доступно только для “тёплой” модели.
- *walltime* - астрономическое время выполнения задания.
- *walltime_s* - астрономическое время выполнения задач-скаутов задания. Доступно только для “тёплой” модели.

На первом этапе было проведен сбор и хранение информации необходимой для построения моделей, интеграция информации между несколькими базами данных. Ниже мы рассмотрим более детально создание аналитической платформы и применение методов Машинного обучения для предсказания работы сложной системы обработки и управления данными.

Как было рассмотрено ранее система распределенной обработки данных эксперимента ATLAS (ProdSys2-megaPanDA) не имеет мировых аналогов по количеству запускаемых задач. В 2016 году системой было обработано 1.4 эксабайта данных. ProdSys2 постоянно изменяется в ответ на растущие требования со стороны эксперимента ATLAS и пользователей системы. В нынешнем её состоянии, ПО системы и часть ее сервисов представляет собой большую, высоконагруженную распределённую вычислительную среду, работающую в условиях, регулярно приводящих к необходимости выполнять высокоприоритетные задачи за кратчайшее время на ограниченных вычислительных ресурсах. «Ручное управление» такой сложной системой (например, «ручное» изменение приоритетов отдельных заданий) может привести к ухудшению работы системы, а также к неоптимальному использованию вычислительного ресурса.

Предварительный анализ журнальных данных о поведении всех элементов системы и метаинформации о ходе выполнения заданий и задач, позволяет

рассчитывать на то, что моделирование системы с помощью алгоритмов машинного обучения может дать практически значимые результаты.

На первом этапе было разработано семейство моделей, предсказывающих длительность выполнения вычислительных заданий на основе классификации, рассмотренной выше. Семейство моделей состоит из 2 моделей, использующих разное количество данных для предсказания длительности выполнения вычислительной задачи на разных этапах её жизненного цикла. “Холодная” модель использует лишь данные, доступные на этапе описания вычислительного задания, без учёта состояния вычислительной системы. Предсказания данной модели наименее точны (для 99% заданий ошибка не превышает 7 дней), но позволяют пользователю получить практически полезную консервативную оценку длительности задания. “Тёплая” модель, использующая также статистические данные, собранные первыми 10 задачами вычислительного задания, позволяет получать значительно более точные предсказания (для 99% заданий ошибка не превышает 1.5 суток). (следующим этапом должно стать создание “горячая” модели, использующей все доступные данные о гетерогенной распределенной киберинфраструктуре, включая информацию о состоянии WAN. “Горячая модель” будет конечным этапом для данного исследования). Все модели основаны на алгоритме Gradient-boosted decision trees (ансамбль решающих деревьев, объединённых методом градиентного бустинга).

Так как данная модель позволяет использовать как численные, так и классификационные предикторы. В качестве обучающей выборки использовались метаданные за июнь 2015 г., а в качестве тестовой - метаданные первой недели июля 2015 г. Модель обучалась на кластере “analytics” в ЦЕРН в среде разработки Spark, с использованием библиотеки MLLib. Результаты по реализации модели, выполненные под руководством автора диссертации, были доложены на международных конференциях 2016 года [81,82]. Реализация модели позволила расширить функции подсистемы мониторирования и предоставить новые

возможности, на рисунке 66 приведена расширенная информация о выполнении одного из зданий (задание 11016615 в марте 2017 года).

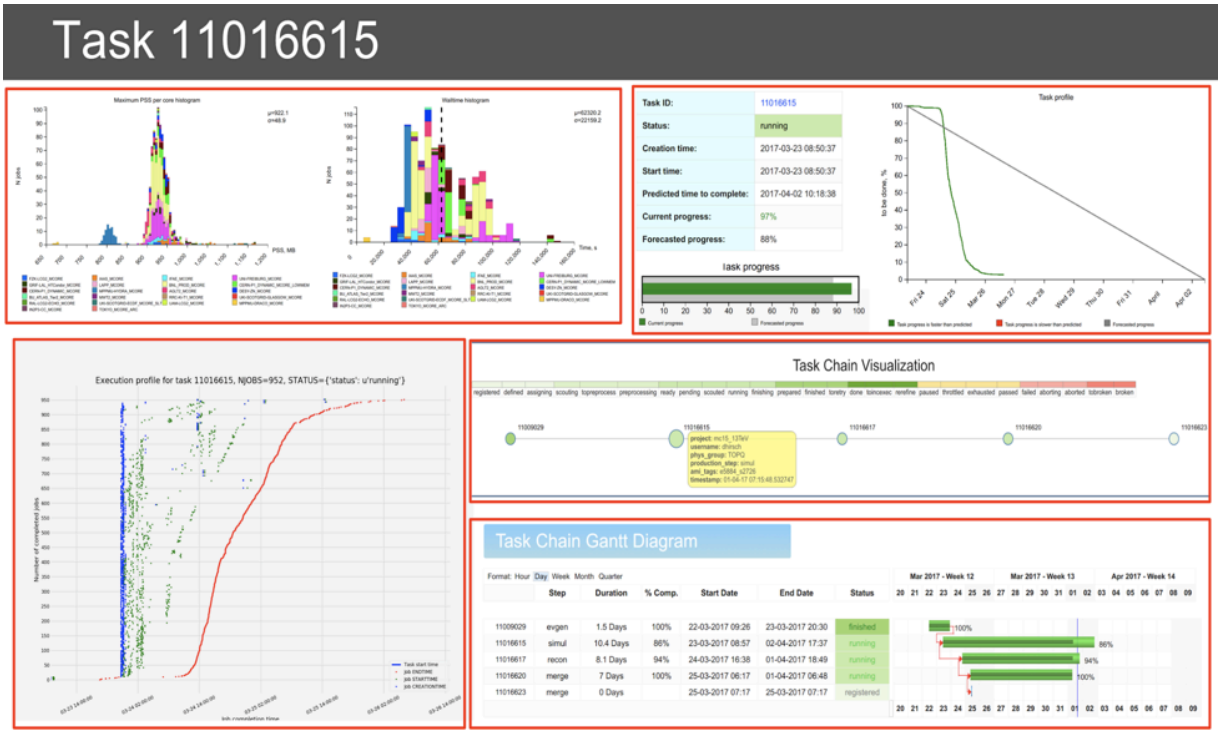


Рисунок 66 - Графики выполнения задания и цепочки заданий. Время предсказания времени выполнения задания

На графиках представлена информация о профиле выполнения задач (нижний левый график), предсказанное время выполнения задания (верхний правый график), а также ход выполнения всей цепочки, связанных логически заданий (второй и третий графики справа). На рисунке 67 показано реальное и предсказанное время выполнения заданий, из графика видно хорошее соответствие модели с реальным временем выполнения заданий.

В результате была разработана комплексная многоуровневая подсистема мониторинга системы управления потоками заданий (WMS) и работы центров грид всех уровней, в том числе суперкомпьютеров и ресурсов облачных вычислений, используемых для выполнения заданий. Подсистема мониторинга обеспечивает постоянный сбор и систематизацию информации об объектах WMS (задания/задачи)

и параметрах работы системы (очереди, квоты, доли), аппаратных и программных ресурсах и сервисах.

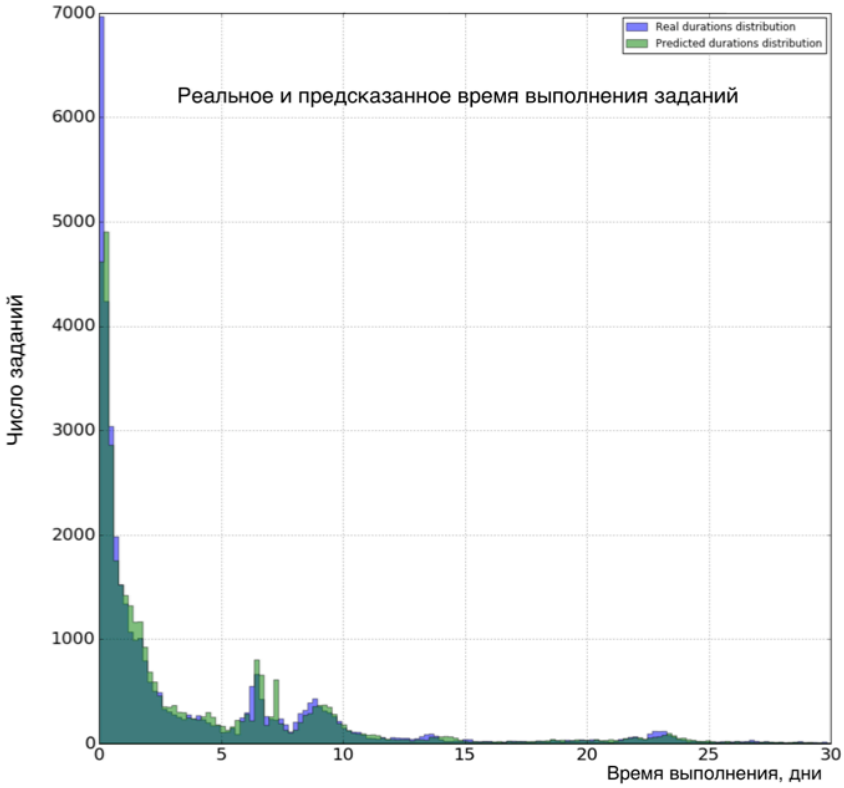


Рисунок 67 - Предсказанное и реальное время выполнения заданий

Подсистема обеспечивает обратную связь при наличии ошибок в выполнении заданий или аномалий в работе WMS, а также выполняют функции анализа и прогнозирования поведения заданий. Разработаны архитектурные принципы, методы и средства для реализации мониторинга системы управления загрузкой распределенной вычислительной среде.

Выводы к третьей главе. На основе проведенного анализа классов потоков заданий, разработана архитектура системы распределенной обработки данных. Разработана методика распределения вычислительного ресурса между различными классами заданий. Создана система управления потоками заданий всего физического

эксперимента, отдельных групп, работающих по определенной тематике, и индивидуальных пользователей. Система имеет квоты и приоритеты, а также позволяет разделять с высокой гранулярностью вычислительный ресурс между различными потоками заданий. Реализация системы и демонстрация ее работы для эксперимента ATLAS показали ее высокую степень масштабируемости, надежности и управляемости. Система не имеет мировых аналогов по этим показателям. Система имеет развитую подсистему мониторинга и аккаунтинга, позволяющая собирать и предоставлять информацию не только о ходе обработки и анализа данных, но и о работе сайтов в рамках гетерогенной инфраструктуры.

Параметры разработанной системы до 300К задач, выполняемых одновременно на всех имеющихся типах ресурсов : высокопропускные центры (грид), ресурсы облачных вычислений, суперкомпьютерные центры (более 250 центров по всему миру), более 2М задач в выполненных в день, и более 30М задач ежемесячно в течение 2015/16 годов. Более 1.3 эксабайта данных, обработанных в системе в 2014 и 2016 годах.

Результаты, представленные в третьей главе подтверждают следующие защищаемые положения:

- новые принципы построения и архитектура глобальной системы для обработки данных в гетерогенной компьютерной среде, которые позволяют эффективно использовать вычислительные ресурсы и снимают противоречие по доступу к ресурсу между экспериментами, группами пользователей и отдельными учеными.
- разработанный комплекс методик, методов и, созданная на их основе, система управления потоками заданий, повышает эффективность обработки данных экспериментов и обеспечивает обработку данных в эксабайтном диапазоне, в масштабе более 2М задач в день в более чем 200 вычислительных центрах для более чем 1000 пользователей;

- подсистема мониторирования и оценки эффективности работы глобальной системы для обработки данных обеспечивает высокий уровень автоматизации при анализе работы системы и сбоев в работе распределенной вычислительной инфраструктуры и ее аппаратно-программных компонент.

Глава 4. Дальнейшее развитие компьютерной модели. Интеграция суперкомпьютеров и ресурсов облачных вычислений с распределенными вычислительными ресурсами грид

В предыдущих главах были рассмотрены роль суперкомпьютеров для приложений физики частиц и создание динамической системы управления потоками заданий, обоснована идея интегрированной связки HPC-HTC, а также причины перехода от иерархической модели MONARC к “смешанной компьютерной модели”, а затем к динамическому управлению ресурсами и созданию “всемирного облака” для выполнения потоков заданий. Эти работы сделали возможным дальнейшее развитие компьютерной модели для приложений физики частиц и перехода к использованию гетерогенных вычислительных ресурсов. Данная глава посвящена вопросам интеграции суперкомпьютеров и ресурсов облачных вычислений с распределенными вычислительными ресурсами грид. В главе рассмотрены дополнительные требования к распределенной системе обработки данных при использовании гетерогенных вычислительных ресурсов. Проведен анализ технологий, позволяющих перейти к созданию федерации географически распределенных дисковых ресурсов и дальнейшему развитию компьютерной модели для экспериментов в области физики частиц.

Фундаментальным вопросом для развития компьютерной модели в области физики частиц является следующий вопрос - как данные будут обрабатываться, анализироваться и моделироваться через 7-10 лет? При ответе на вопрос необходимо учитывать ограничения бюджета, практически во всех странах, на увеличение вычислительных мощностей для экспериментов на LHC, и существующие бюджеты для новых комплексов (NICA, FAIR) и проектов (LSST, DUNE). До последнего времени компьютерная модель строилась в предположении, что эксперименты ФВЭ и ЯФ являются “собственниками” вычислительного ресурса. Работы многих групп в

разных странах в последние годы были направлены на то, чтобы показать, как вычислительная инфраструктура, не принадлежащая экспериментам и/или ассоциированным с ними ВЦ, может быть эффективно использована и интегрирована с системой распределенных вычислений грид. Вариантами ответа на поставленный вопрос могут быть:

- эксперименты ФВЭ и ЯФ будут продолжать покупать необходимое аппаратное обеспечение и расширять свою компьютерную инфраструктуру;
    - очевидное преимущество - это преимущество “собственника” ресурса, ресурс может быть использован и доступен в любой момент;
        - это преимущество надо учитывать только в случае, если есть достаточный ресурс в момент максимальной загрузки (кампании анализа и переобработки данных), в остальное время вычислительный ресурс не будет использован в полном объеме;
- эксперименты ФВЭ и ЯФ будут покупать мощности у тех, кто их предоставляет на коммерческой основе;
    - преимущество такого подхода состоит в том, что капитальные затраты несет третья сторона;
    - недостатком является отсутствие гарантий, что ресурс будет доступен в требуемом объеме или доступен для использования, когда это потребуется; А также необходимость “доверия” к третьей стороне и предоставления ей доступа к данным международной коллаборации.

Компромиссным является вариант, когда базовые ресурсы принадлежат экспериментам, а в момент максимальной нагрузки “используется” поставщики вычислительных услуг и сервисов.

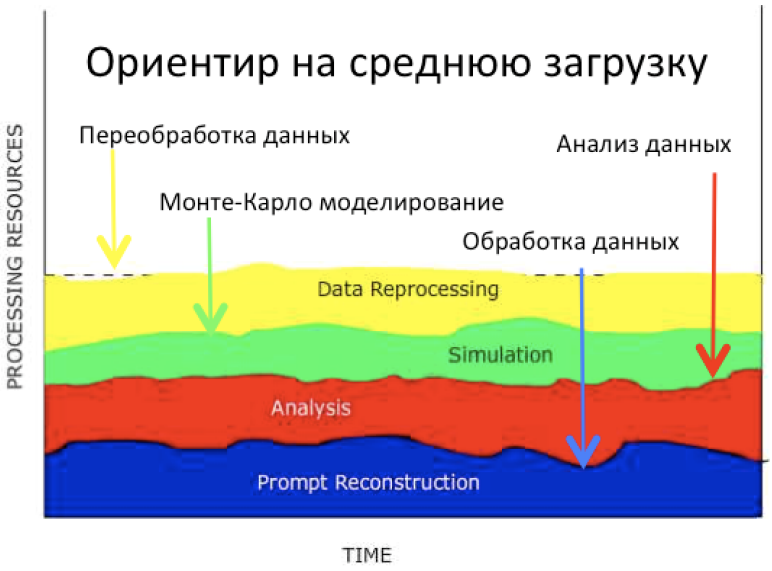


Рисунок 68 - График использование вычислительного ресурса при сценарии “средняя загрузка”

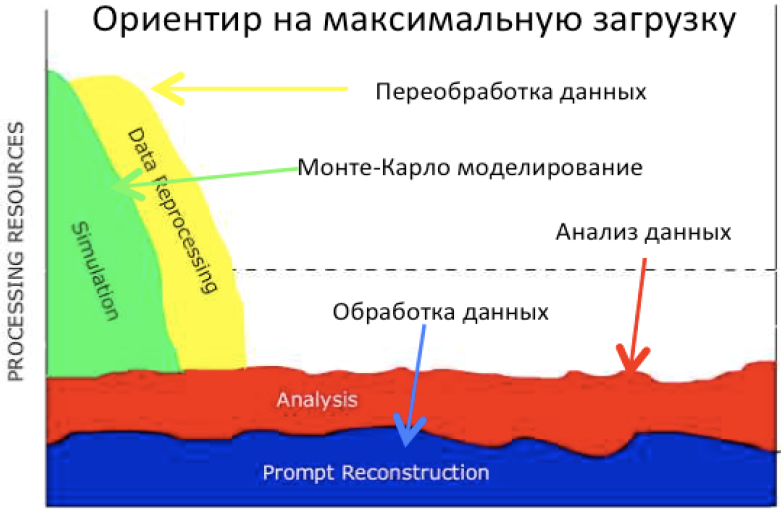


Рисунок 69. График использование вычислительного ресурса при сценарии “пиковая загрузка”

Как рассматривалась ранее, ландшафт современных вычислительных ресурсов и потребности в них драматически отличаются от ситуации 10 летней давности, когда приложения ФВЭ и ЯФ были одним из основных “потребителей” вычислительных мощностей в глобальном мире (за исключением приложений связанных с военной тематикой и исследованием климата). Долгие годы закупки компьютеров и расчет необходимой мощности ВЦ осуществлялись с ориентиром на среднею загрузку и одновременное выполнение всех потоков заданий (рисунок 68). В настоящее время существует больший пул вычислительных ресурсов, за пределами ФВЭ и ЯФ. В первую очередь это коммерческие ресурсы и суперкомпьютерные центры. Так вычислительный ресурс гигантов ИТ индустрии: Яндекс, Google, Amazon, Microsoft в сотни раз превышает мощности консорциума WLCG, ресурс третьего по мощности суперкомпьютера Titan превышает весь ресурс

WLCG). Это позволяет и требует пересмотра “усредненного” подхода к использованию вычислительных мощностей и смены модели с ориентацией использования максимального вычислительного ресурса на момент пиковой нагрузки и соответствующее планирование потоков заданий (рисунок 69). При таком сценарии классы потоков заданий могут быть переориентированы соответственно, т.к. переобработка данных и Монте-Карло моделирование, как правило имеют название “кампания”, и должны быть закончены как можно быстрее, в тоже время обработка данных и физический анализ данных имеют хорошо выраженную постоянную составляющую.

Ориентация на пиковые нагрузки предъявляет повышенные требования к системам распределенной обработки данных. Они должны быть готовы увеличить количество задач в сотни раз, быстро “захватывать” доступные ресурсы и уметь быстро их оставлять. Одновременно требования предъявляются и к базовому ПО (программы реконструкции событий, восстановления треков частиц и т.д.) физических экспериментов (помимо обозначенных в главе 2), при использовании коммерческого ресурса любые “бесконечные” петли в коде, огромные журнальные файлы (и соответственно дисковое пространство и время передачи необходимые для их хранения и передачи), неэффективное использование процессорного времени становятся настоящим расточительством, возможно необходимо было обратить на это внимание и раньше, но при новом сценарии такая неэффективность приведет к заметным финансовым затратам.

4.1 Интеграция ресурсов облачных вычислений и грид

Идея “облачных вычислительных” ресурсов не нова, в 1961 году пионер ИТ Джон МакКарти (более известный, как человек введший в обиход выражение “искусственный интеллект” - artificial intelligence”) предсказал, что “вычисления могут быть организованы, как общедоступный сервис” (программа-утилита), и он

продолжал размышлять и описывать как это может быть реализовано. Идея о том, что вычисления проводятся не на локальных ресурсах, а на централизованных вычислительных мощностях, предоставляемых и поддерживаемых некоторой третьей стороной получила новое развитие и реализацию в последние десятилетия.

Развитие коммерческих и академических ресурсов облачных вычислений и применение их для приложений физики частиц началось сравнительно недавно, для экспериментов в области ФВЭ, ЯФ и астрофизики, это совпало по времени с необходимостью обработки и анализа десятков петабайт данных, с пониманием ограничений грид и стоимости поддержания существующих центров и создания новых. В эти годы Amazon, Google, Yandex, Microsoft создали действующие центры, состоящие из сотен тысяч компьютеров. Существуют разные определения ресурсов “облачных вычислений”, остановимся на определении данном одним из “отцов-грид” Яном Фостером: “Широкомасштабная распределенная компьютерная парадигма, обусловленная экономией затрат, в которой пул абстрактных, виртуальных, динамически масштабируемых управляемых вычислительных мощностей, хранилищ, платформ и сервисов предоставляется по требованию внешним клиентам через интернет. “[83]. Определение данное Фостером и его статья, не противопоставляет грид и *“cloud”* компьютинг, а наоборот подчеркивают общность между различными подходами в архитектуре, технологии и взглядах на организацию распределенных вычислений. Фостер предсказал, что необходимо будет найти пути и решения, чтобы определить, как будет эволюционировать общая инфраструктура, но он думал, что поиск таких путей и решений займет гораздо больше времени и следующие пять лет приведут к созданию реальных прототипы и прообразом будущих инфраструктур.

Для научного сообщества стал откровением доклад профессора Мартина Севиора из университета Мельбурна на симпозиуме ISGC в 2009 году (International Symposium for Grid and Clouds) об использовании ресурсов компании Amazon для проведения моделирования методом Монте-Карло для эксперимента Belle [84]. В

докладе была приведена статистика о “стоимости” генерации 0.85M событий с использованием коммерческих ресурсов, “цена” одного события составила $0.53, что было меньше чем использование центра T2 в Мельбурне (с учетом накладных расходов). Интерес к докладу привел к дальнейшему развитию работ в данном направлении, потому что кампания по моделированию событий эксперимента Belle была кратковременной (неделя) и использовала сравнительно небольшие вычислительные мощности (20 HighCPU-XL машин архитектуры: 8 ядер, 17 ГБ RAM). Для проведения полного сравнения эффективности использования коммерческого ресурса облачных вычислений и грид необходимо было получить ответы на вопросы:

- насколько стабильно будут работать коммерческий ресурс облачных вычислений в течение продолжительного времени (месяцы), включая передачу и хранение данных;
- насколько стабильно будет работать коммерческий ресурс облачных вычислений если размер ресурса сравним с размером центра уровня T2;
- насколько легко и прозрачно ресурсы могут быть интегрированы с системой управления заданиями;
- и сколько это будет стоить.

С самого начала данное исследование рассматривалось с практической точки зрения с применением в будущем для больших экспериментов класса мегасайенс. Решение о проведении исследования совпало с желанием компании Google дать доступ к GCE (Google Compute Engine) [85]. Группой разработчиков под руководством автора диссертации, возглавлявшего тогда проект распределенных вычислений и компьютинга эксперимента ATLAS, и представителями компании Google была достигнута договоренность о предоставлении 5M ЦПУ часов (это примерно соответствовало 4000 ядер в течение двух месяцев). Вычислительный ресурс был интегрирован с системой управления потоками заданий и описан в ИС (тип рабочей очереди, доступность ресурса, формализованное описание сайта).

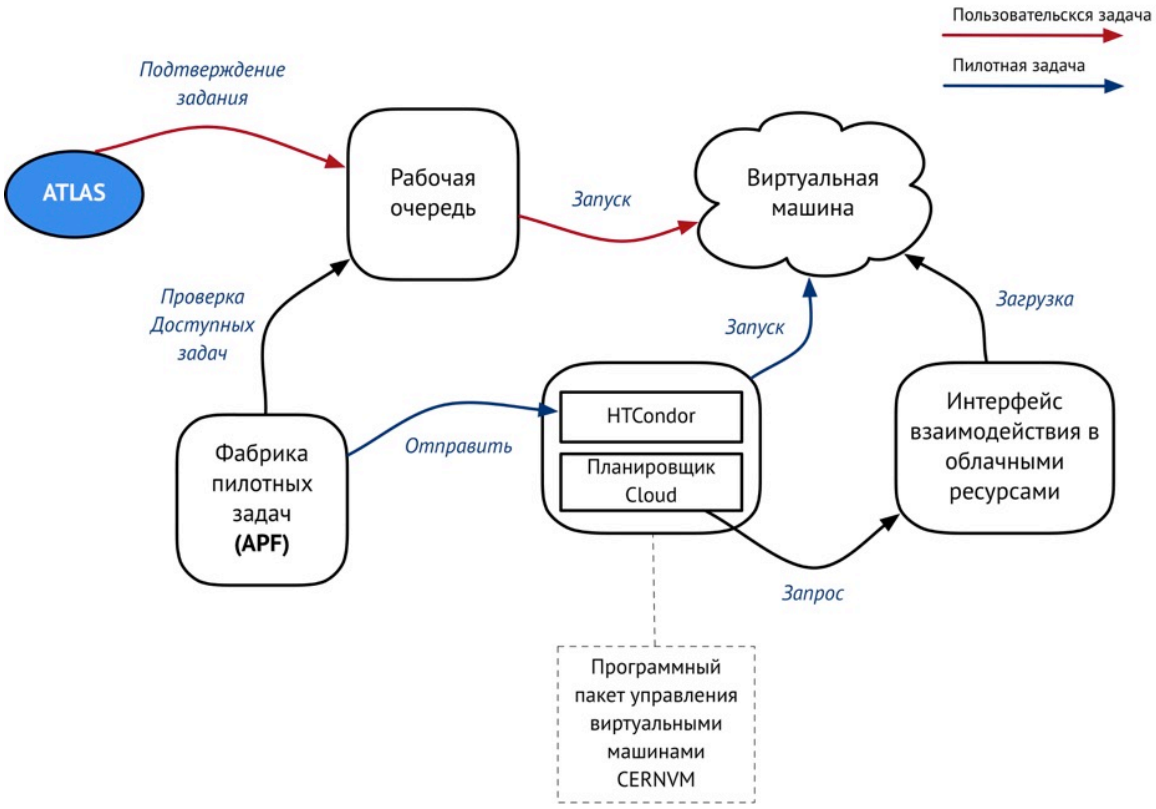


Рисунок 70 - Схема запуска задания в “облачном вычислительном ресурсе”

Система состояла из нескольких интегрированных вместе компонент. В состав системы входили: HTCondor [86] (использовался как для отправки и развертывания виртуальной машины (VM) на облачной платформе, так и как локальная (внутри VM) система пакетной обработки, промежуточное программное обеспечение рабочего узла (для авторизации и передачи файлов), сетевая файловая система CVMFS [87] (для хранения и доступа к программному обеспечению физического эксперимента) и фабрика пилотных заданий APF (AutoPyFactory). Также был использован пакет виртуализации, разработанный в ЦЕРН (CERNVM [88]). Схематично процесс запуска задания показан на рисунке 70 (значение и функции фабрики пилотных заданий и очередей подробно рассмотрены в главе 3).

Результаты исследования могут быть суммированы следующим образом :

- планирование, создание и настройка системы, включая описание очереди в ИС заняло 2 недели;

- вычислительный ресурс использовался в течение 8 недель (10 недель, включая настройку);
- около ~458K задач было выполнено, и произведено/обработано 214M событий. Максимальное достигнутое использование составило 15K задач/день. Число сбоев составило 6%, что меньше, чем в среднем по грид;
- работа, предоставленного коммерческого ресурса была стабильна и надежна. По окончании работы все результаты были переданы в один из центров уровня T1 эксперимента ATLAS.

График на рисунке 71 показывает ход выполнения задач ATLAS на ресурсах GCE по дням. Зеленым отмечены успешные задачи, розовым - задачи имевшие ошибки. Данная работа была пионерской. Впервые была показана возможность масштабного использование коммерческого ресурса “облачных вычислений” для приложений ФВЭ и ЯФ. Такая интеграция стала возможна только при наличии системы управления загрузкой, которая была разработана и реализована согласно разработанным автором методам и подходам, а также новой компьютерной модели.

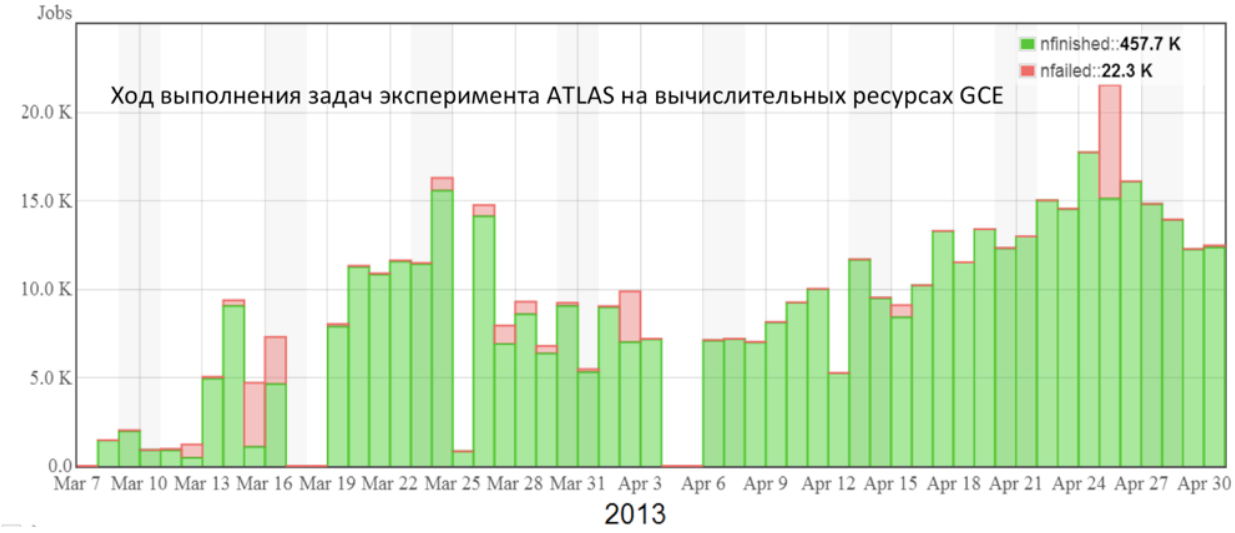


Рисунок 71 - Количество задач ATLAS, выполненных на ресурсах Google Compute Engine

Дальнейшее развитие “облачных вычислений” привело не только к их широкому использованию, но и расширению классов заданий для них. За последние

два года (2015/16) были интегрированы академические ресурсы облачных вычислений в Канаде и Австралии, а также ресурсы компании Amazon.

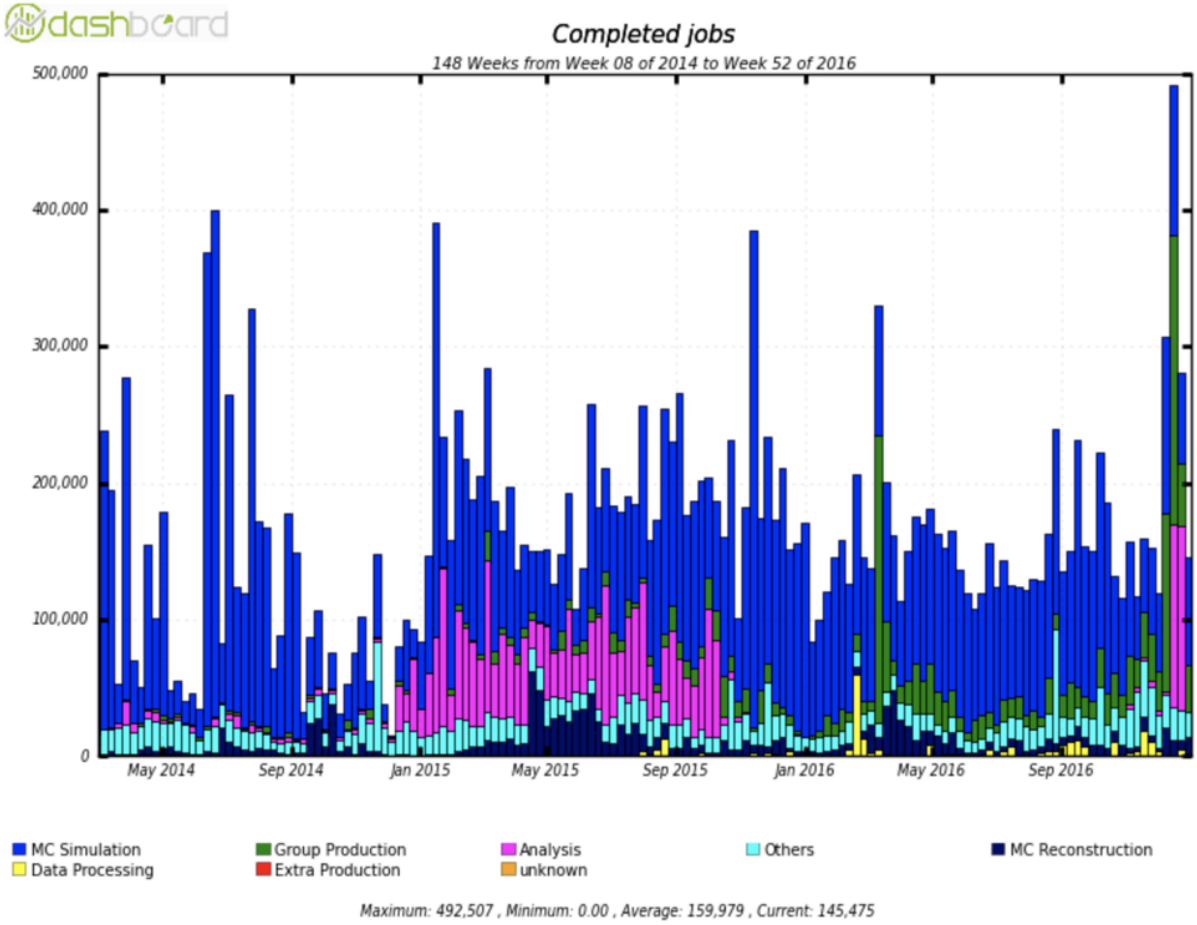


Рисунок 72- Еженедельное количество задач, выполненных только на ресурсах облачных вычислений в 2014-2016 годах

На рисунке 72 показано еженедельное количество задач, выполненных только на ресурсах облачных вычислений за 2.5 года (2014-2016 годы). Из графика видно, что до 400К задач выполняется еженедельно. Среди потоков заданий доминируют задачи Монте-Карло моделирования (MC simulation), в тоже время существуют периоды, когда задачи анализа (Analysis) активно используют “облачный ресурс”.

Следует отметить, что коммерческие ресурсы все еще дороже на 20-30%, чем ресурсы WLCG [89], но было важно начать исследования по созданию гетерогенной компьютерной среды и выхода за пределы WLCG, показав возможности системы

управления загрузкой для нового класса ресурсов и их интеграции с распределенной системой вычислений грид.

4.2 Интеграция суперкомпьютеров и грид

В главе 2 была подробно рассмотрена роль приложений ФВЭ и ЯФ для суперкомпьютерных центров и возможная роль суперкомпьютеров для научной программы физики частиц. Мы также обсудили, чем был мотивирован выбор распределенной компьютерной модели для экспериментов в области физики частиц. В настоящее время практически все эксперименты на новых и строящихся машинах используют и планируют использовать распределенную модель компьютинга. Потребности в дополнительных вычислительных мощностях, создание системы управления заданиями ProdSys2-megaPanDA позволили разработать методы интеграции суперкомпьютеров с грид и использовать мощности СК для научных приложений физики частиц. Одним из основных аргументов явился факт сравнения мощности WLCG и LCF Titan (LCF, от англ. Leadership Class Facilities),

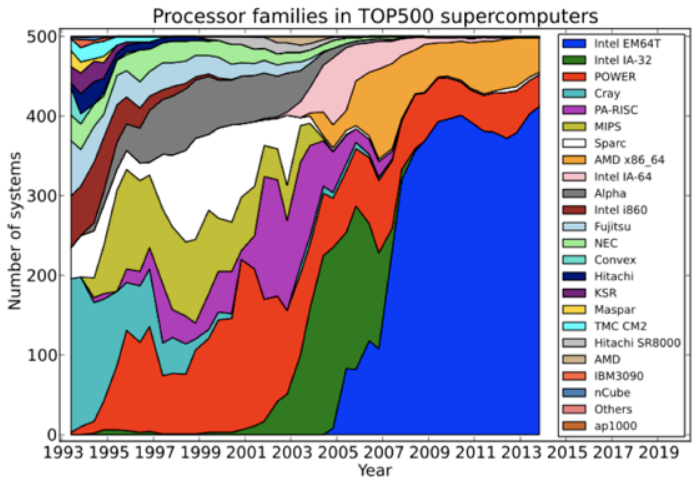


Рисунок 73 - Эволюция развития аппаратной базы суперкомпьютеров

- WLCG : 220000x86 вычислительных ядер;
- Titan : 300000x86 вычислительных ядер и 18000 графических процессоров;

а также взаимный интерес сторон к совместной работе, потому что научная программа международных сотрудничеств ATLAS и ALICE, а также возможные результаты исследований и потенциальных открытий выглядели очень привлекательно для суперкомпьютерных центров, а разработанная система

управления загрузкой предлагала новые возможности для управления потоками заданий в суперкомпьютерных центрах (более подробно этот вопрос рассмотрен в главе 2). Эволюция аппаратной базы суперкомпьютеров, приведенная на рисунке 73. Распределение вычислительной мощности между ТОП500 СК, представленное на рисунке 74 (по материалам [90] и суперкомпьютерной конференции 2016 года [91]) позволяет сделать два важных вывода:

- Ко второму десятилетию XXI века количество аппаратных решений значительно сократилось и фактически свелось к трем архитектурам. Многие СК из первой десятки аппаратно совместимы с вычислительными мощностями WLCG;
- Двенадцать первых машин в списке ТОП500 обладают половиной всей мощности всех компьютеров списка;

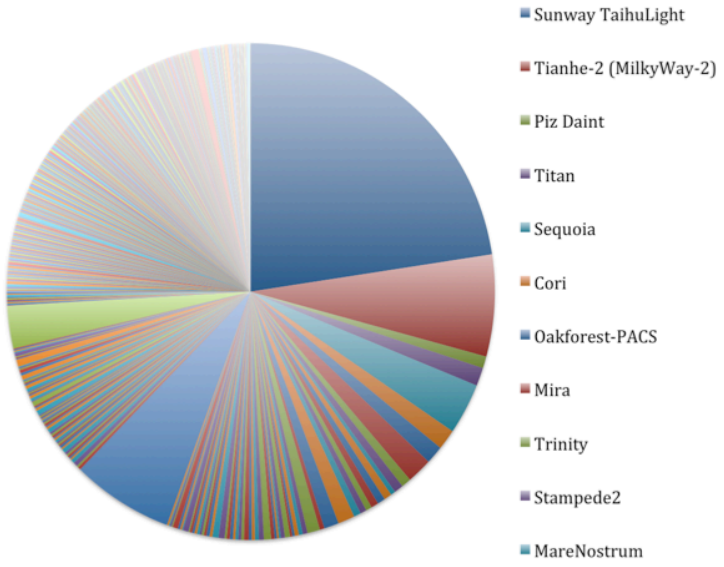


Рисунок 74- Распределение вычислительной мощности между ТОП500 суперкомпьютеров

Таким образом, наиболее прагматичным подходом было попробовать разработать не только универсальное решение, но и использовать СК первого десятка для его демонстрации. Необходимо было ответить на следующие фундаментальные вопросы :

- как получить «время» на суперкомпьютерах?
- как интегрировать суперкомпьютеры с инфраструктурой WLCG и системой распределенных вычислений, принятой экспериментами ФВЭ и ЯФ ?
- как выполнять программный код экспериментов ФВЭ и ЯФ на СК , и как делать это эффективно ?

Выделение ресурса на суперкомпьютерах. Выделение ресурса на СК - это та область, где существует очень высокая конкуренция между научными и техническими предложениями высокого класса, для различных областей знаний: биоинформатика, квантовая хромодинамика, исследование климата, моделирование в астрофизике и астрономии, и т.д. Распределение ресурса происходит после рассмотрения проектов, предложенных сравнительно небольшими (по меркам экспериментов ФВЭ и ЯФ) группами ученых. Многие группы имеют длительную историю и экспертизу в использовании СК. Этот подход принципиально отличается от политики WLCG, где распределение ресурса происходит между виртуальными организациями (экспериментами). Так коллаборация ATLAS не может участвовать в конкурсе на вычислительные мощности СК Titan или СК НИЦ КИ. Проект должен быть представлен группой ученых и поддержан соответствующим экспертным советом. Как и в случае с GCE интерес состоял не в разовом действии, а в интеграция СК ресурса на длительной основе. И оказалось, что интерес может быть обоюдным (мы вернемся к обсуждению классической системы выделение СК ресурса чуть позже).

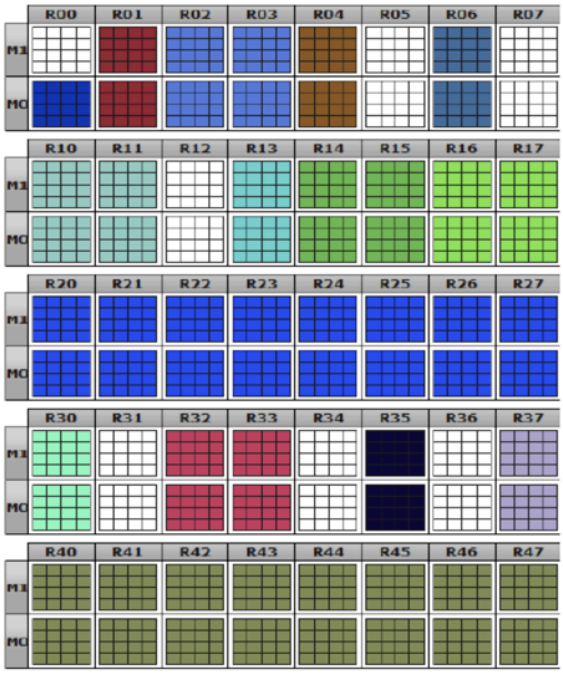


Рисунок 75 - Карта использование процессоров суперкомпьютера класса LCF

Классическая карта использования LCF показана на рисунке 75, возможно, что это был не лучший день работы конкретной машины, т.к. она была заполнена задачами только на 85% (заполненные узлы отмечены цветом, свободные узлы отмечены белым). Из карты видно, что наибольший свободный раздел может предоставить “только” 1024 узла, видимо задачи, ожидающие своего выполнения, требовали в этот момент большего

количества узлов, поэтому на короткий срок (а нам будет необходимо определить, что означает “короткий срок”) возможно использование этих узлов. Средне годичная занятость LCF Titan составляет 90%, т.е. “свободный” ресурс составляет около 300M ЦПУ часов в год. В отличии от HTC, “занятость” СК не всегда предполагает 100% загрузку (безусловно это зависит от типов задач, выполняемых на СК). В тоже время типичные приложения физики частиц, такие как Geant4 (G4 [92]) или ROOT [57]) являются прекрасными кандидатами. Эти приложения могут использовать временно свободные узлы СК. Предложение о выполнении задач ФВЭ и ЯФ (на первом этапе для экспериментов ATLAS и ALICE) на СК в фоновом режиме и тем самым повышение общей эффективности использования СК заинтересовало сразу несколько центров в России, США и Чехии. Для отладки данной методики и интеграции СК с WLCG были выделены квоты в 1M ЦПУ часов в год, в предположении, что основное время будет получено за счет работы в фоновом режиме, а квота будет гарантировать время необходимое для отладки и проверки предложенной методики. Так был найден ответ на первый фундаментальный вопрос, и группе под руководством автора диссертации удалось вступить в «клуб» пользователей СК.

Интеграция суперкомпьютеров с инфраструктурой WLCG и системой распределенных вычислений. Рассмотрим некоторые особенности суперкомпьютеров и как это может влиять на процесс интеграции HPC и HTC ресурсов.

- каждый суперкомпьютер уникален;
  - уникальная архитектура и оборудование;
  - специализированная операционная система, рабочие узлы с ограниченной оперативной памятью на рабочий узел;
  - архитектура может отличаться от принятой в WLCG Intel x86, в таком случае требуется кросс-компиляция кода;

- мы не рассматриваем здесь еще более сложный случай с использованием графических процессоров;

- уникальная система запуска задач в СК;
  - с ограничением числа задач в очереди, запускаемых одним пользователем;
- уникальная система безопасности;
  - Например: однократная интерактивная аутентификация с помощью пароля;
  - Отсутствие связи рабочих узлов с интернет (программа пилотных заданий не может работать на рабочем узле, т.к. не будет иметь связи с сервером). Единственный из узлов, имеющий связь с “внешним” миром - это входной узел (login node);
- высокая конкуренция в распределении времени между проектами с ориентацией на проекты демонстрации “превосходства” в различных областях науки.

Следует отметить, что для СК, которые могут быть рассмотрены как HPC-кластер и могут характеризоваться наличием:

- N x86 ядер;
- рабочие узлы имеют TCP/IP связь с внешним миром;

вопрос интеграции с HTC ресурсами не столь сложен, СК такого типа должны быть правильно описаны в ИС, но с точки зрения системы управления загрузкой могут рассматриваться как сайт инфраструктуры грид без дискового ресурса, и результаты вычислений должны быть переданы в место постоянного хранения. Использование таких машин и их интеграция имеют скорее значение для операторских служб, и не предъявляют дополнительных требований к системе управления потоками заданий.

Теоретическое обоснование и возможные подходы как эффективно интегрировать ресурсы суперкомпьютеров (HPC) и распределенные высоко-

пропускные ресурсы (HTC) грид были рассмотрены во второй главе. Проблема интеграции имеет более общий характер, чем просто применение суперкомпьютеров для обработки данных LHC (или экспериментов в области физики частиц и результатов наблюдений в астрономии).

С точки зрения крупных суперкомпьютерных центров, основным вопросом является вопрос как наилучшим образом интегрировать рабочие нагрузки с большими требованиями к вычислительному ресурсу, например, традиционные рабочие нагрузки для СК центров (климат, биоинформатика, расчет на решетках КХД), с большими объемами рабочей нагрузки, возникающими, например, при работе с экспериментальными данными и данными астрономических наблюдений? Для такой интеграции в первую очередь необходима система управления загрузкой (потоками заданий). Модульная структура системы управления потоками заданий оказалась применима для интеграции HPC и HTC ресурсов. На первом этапе необходимо было исследовать насколько предположение о “кратковременном” наличии свободных узлов справедливо и понять могут ли узлы эффективно использоваться для приложений ФВЭ и ЯФ. На рисунке 76 представлен двумерный график, показывающий корреляцию между количеством свободных узлов и длительностью временного интервала, в течение которого они были свободны (измерения проводились для OLCF Titan, в течение месяца и было сделано более 62.5К измерений).

Из графика видно, что в среднем свободен 691 рабочий узел в течение 126 минут (красная и оранжевая линии на графике, соответственно), и до 15К узлов в течение 30-100 минут. Это дает брокеру задач дополнительные возможности в их распределении по вычислительным ресурсам, а варьируя число генерируемых или моделируемых событий, “создавать” задачи различной длительности. Среднее время выполнения задач приведено на рисунке 77 (задание №9235668, ноябрь 2016).

Из приведенного распределения времени выполнения 99.4К задач видно, что среднее время выполнения составляет около 64 минут, таким образом, варьирование

числа обрабатываемых/генерируемых событий, а значит, и времени выполнения задания в зависимости от количества и длительности свободных узлов СК позволит более эффективно использовать “свободные узлы” СК и для выполнения задач.

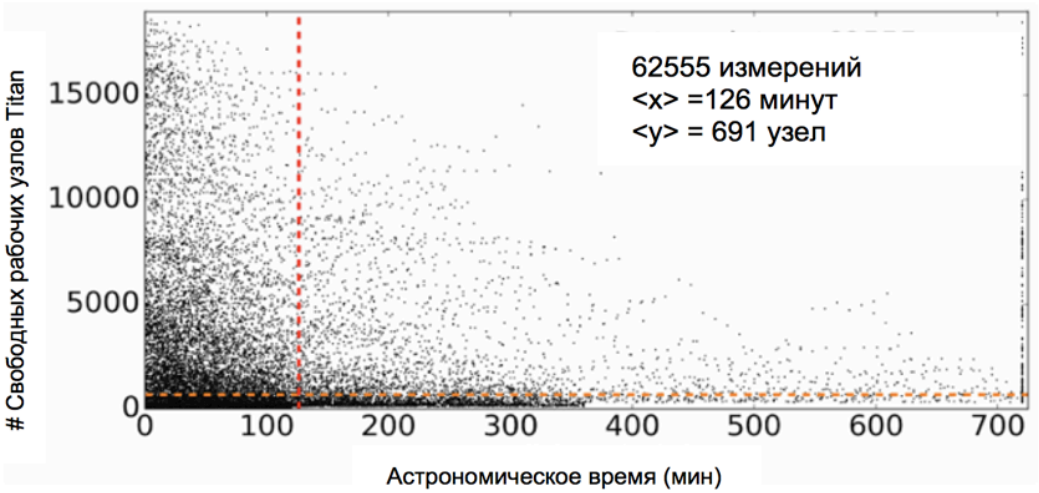


Рисунок 76 - Корреляция количества свободных узлов LCF Titan и времени, когда узлы были свободны

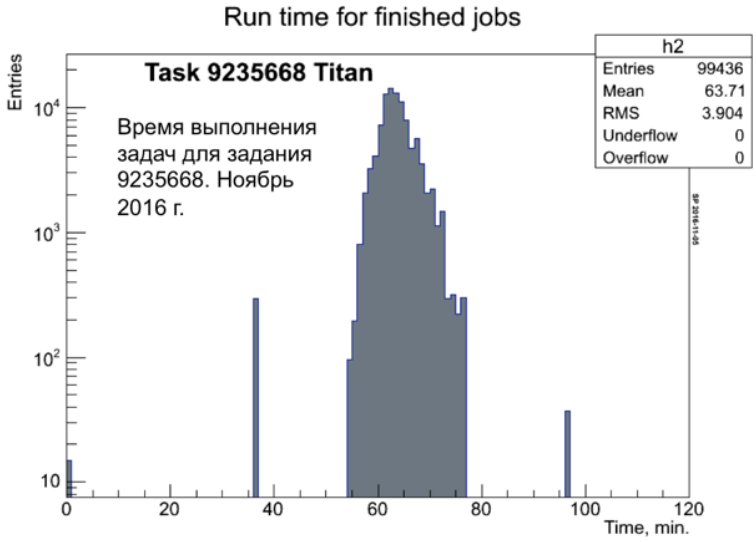


Рисунок 77 - Среднее время выполнения задач моделирования эксперимента ATLAS на СК Titan

Интеграция HPC ресурсов и их будущее эффективное использование, требуют изменения в методике управления заданиями. “Размер” задачи (время ее

выполнения) должен устанавливаться с учетом “размера” (мощности) свободного ресурса. Такого требования не существовало в гомогенной системе грид, когда размер всех задач в рамках одного задания был одинаков. При архитектуре WMS, описанной ранее, и разделении уровней подготовки задания (DEFT) и выполнения задания (JEDI), требование о динамическом распределении количества задач становится реализуемым на уровне JEDI, более того, разработанная методика позволяет использовать различные ресурсы (HTC, HPC, “облачные ресурсы”) для одного задания, сформировав “всемирное облако”.

Интеграция суперкомпьютера Titan с грид инфраструктурой, используя подход ProdSys2-megaPanDA. Основной идеей подхода было использование уже существующих компонент системы управления загрузкой и следование логики ее работы. Основные изменения были связаны с работой и логикой работы задач-пилота. Классическое использование таких задач предполагает их выполнение на каждом рабочем узле и передачу информации центральному серверу PanDA (рисунок 40, раздел 3.2) Единственный узел СК, имеющий связь с внешним миром - это узел DTN (от англ. Data Transfer Node), реализованная архитектура показана на рисунок 78. Пилотная задача (PanDA Broker) :

- обменивается информацией с центральным сервером;
- получает информацию от сервиса СК о наличии свободных узлов и периоде их доступности;
- через систему пакетной обработки СК запускает задачи на выполнение;
- по окончании задачи инициирует передачу результатов на центр грид.

Данная архитектура не является специфичной для ФВЭ и ЯФ, или для конкретного типа СК. Для взаимодействия с системой пакетной обработки СК используется интерфейс на база пакета SAGA-Python (Simple API for Grid Applications) , выполняемый программный код должен находится на распределенной файловой системе СК, центральный сервер может находиться в любом месте (так

для различных применений сервер был установлен в ЦЕРН (эксперимент ATLAS), в EC2 (проект LSST), в ОИЯИ (эксперимент COMPASS), в НИЦ КИ (приложения биоинформатики).

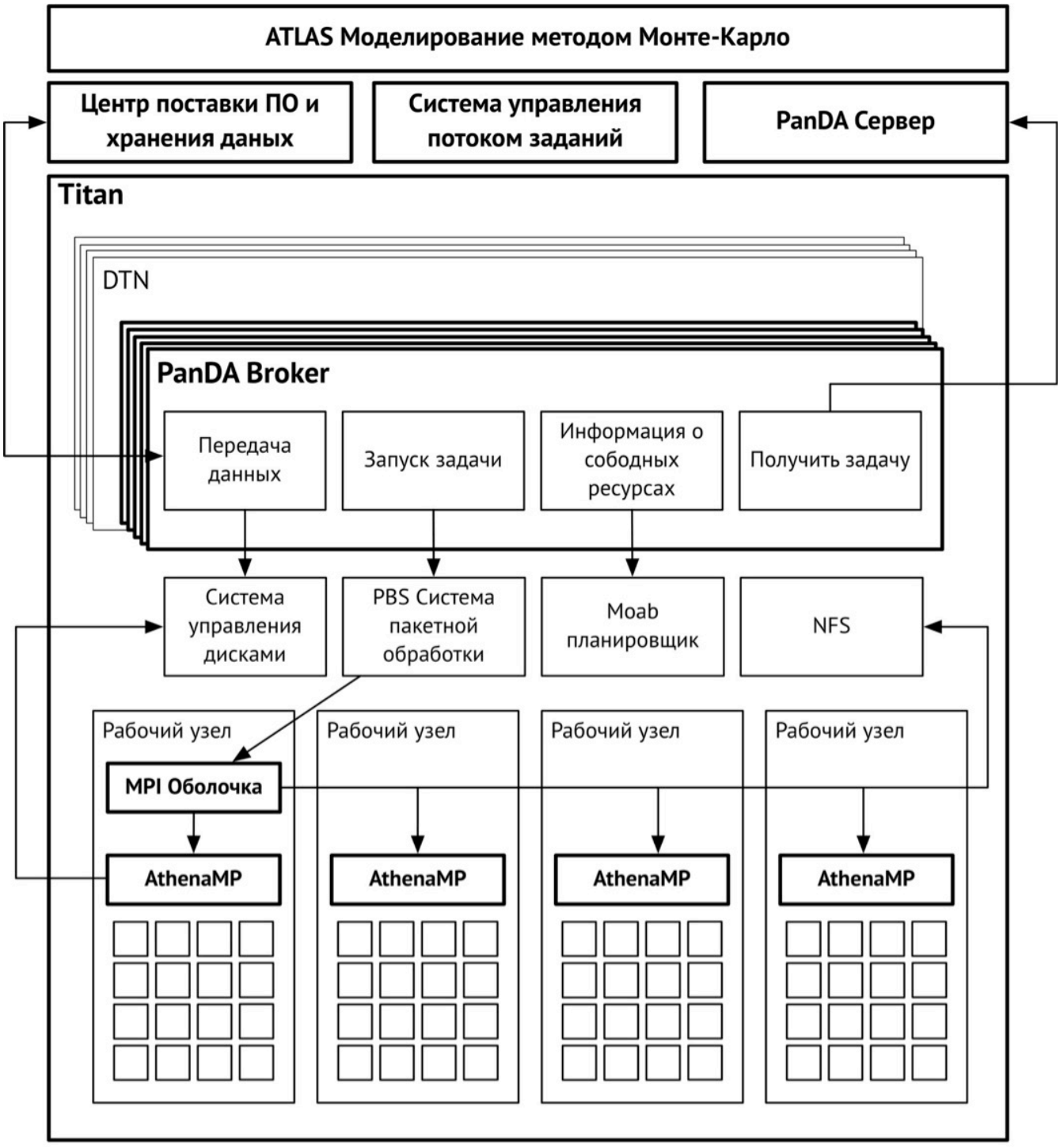


Рисунок 78- Архитектура системы управления заданиями для суперкомпьютера

Сама методика интеграции и использования HPC и HTC ресурсов была успешно применена для многих СК, в том числе Titan, Anselm, НИЦ КИ и СК Университета Иллинойс Урбана Шампейн. Количество свободного ЦПУ ресурса и его использование для 15 месяцев 2016/17 годов показано на рисунке 79. Из графика видно, что использование свободного ресурса наращивалось постепенно, и февралю/марту 2017 года превысило 31%.

Общий вклад суперкомпьютерных ресурсов в реализацию программы экспериментов в области ФВЭ и ЯФ рассмотрен в конце данной главы.

Третьим фундаментальным вопросом интеграции суперкомпьютеров является: **Как выполнять программный код экспериментов ФВЭ и ЯФ на СК, и как делать это эффективно**. Разработка архитектуры базового программного обеспечения и оптимизация работы кода физических программ выходит за пределы

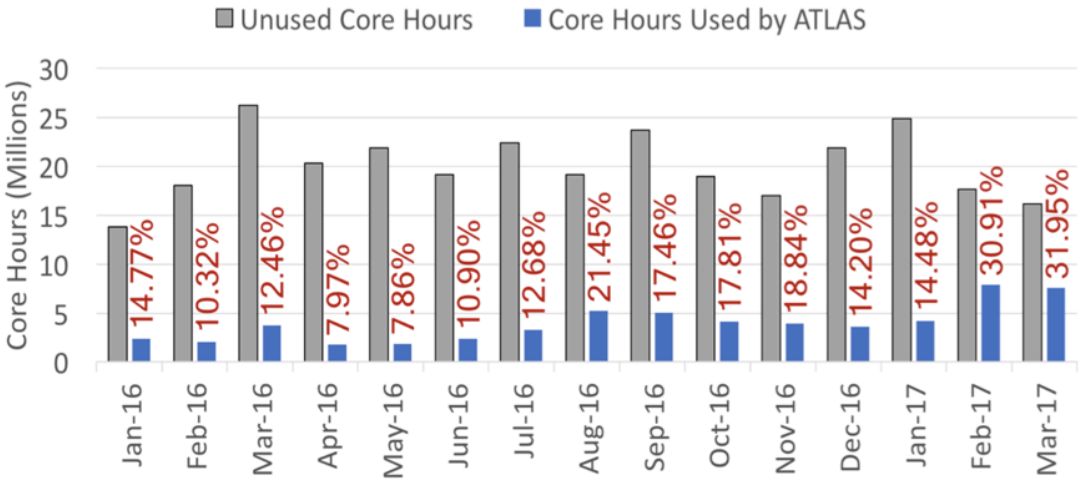


Рисунок 79- Количество свободного ЦПУ ресурса LCF Titan, и его использование в фоновом режиме

данного исследования и представляет собой отдельный крупный проект в ФВЭ и ЯФ. Мы кратко остановимся только на том, что будет способствовать дальнейшему использованию СК для приложений ФВЭ и ЯФ. Ответ на вопрос требует значительного изменения в структуре кода, его большей модульности, например, четкого разделения между кодом моделирования и реконструкции событий, что

позволит уменьшить зависимость программ моделирования от внешних библиотек. Код многих экспериментов органично рос с годами работами и сейчас достигает 4М линий (в большей части на языках python и C++). Используемые фреймворки (Gaudi, Athena. AliROOT [96-98]) были разработаны 10-15 лет назад. Вышеизложенное требует совместной работы физических экспериментов и специалистов в области системотехники и других компьютерных наук. И проблема хорошо понимаемая в физическом сообществе, НИР последних лет направлен на разработки новых фреймворков : AthenaMP, AlFa.

Использование графических процессоров (ГПУ) может дать гораздо больший эффект, если они будут использованы на СК для приложений физики частиц. Одними из первых кандидатов могут стать задачи “машинного обучения” для исследования работы сложных систем обработки и анализа данных, а также программы восстановления треков частиц. Первые работы в рамках эксперимента ALICE с восстановлением треков камеры TPC в триггере высокого уровня являются успешными, также есть тенденция к разработке общего фреймворка для будущих экспериментов и для будущих машин (AlFa) [99].

4.2.1 Развитие компьютерной модели. Интеграция суперкомпьютера НИЦ “Курчатовский институт” с системой вычислений грид

В данном разделе мы рассмотрим интеграцию суперкомпьютера НИЦ “Курчатовский институт” с грид. Эта работа привела к использованию СК не только для приложений ФВЭ и ЯФ, но и применение системы управления потоками заданий для приложения в биоинформатике.

Детектор переходного излучения эксперимента (TRT – Transition Radiation Tracker) ATLAS является составной частью внутренней системы эксперимента, измеряющей импульсы заряженных частиц. Роль TRT состоит в улучшении пространственного разрешения при восстановлении высокоэнергичных треков и позволяет провести распознавание электронов и пи-мезонов. Восстановление

сигнала от каждого пропорционального счетчика в TRT в условиях большой загрузки детектора увеличивает затрачиваемое процессорное время и позволяет провести исследования для одних из самых сложных задач, а именно исследовать работу детектора и промоделировать его работу на этапе суперLHC (работа ускорителя с увеличенной светимостью и при большей энергии пучков).

Для выполнения задач реконструкции использовалась компьютерная инфраструктура WLCG, но на момент проведения (пере)обработки данных все выделенные эксперименту компьютерные ресурсы полностью были загружены. В связи с этим, важным этапом стала интеграция новых компьютерных мощностей, таких как суперкомпьютеры, в единую компьютерную инфраструктуру.

Задача реконструкции p-p событий при высокой множественности является одной из наиболее сложных задач, возникающих в ходе физических исследований в эксперименте ATLAS, и решение задачи требует значительного вычислительного ресурса. Необходимо было решить две задачи, интегрировать Суперкомпьютерный Центр (СКЦ) и центр T1 грид, в терминах созданной системы управления загрузкой (megaPanDA), проверить полученные физические результат (в силу сложности программного кода ATLAS и аппаратных различий между T1 и СКЦ, этот шаг был необходим). Проверка была осуществлена в несколько этапов. На первом этапе происходило подтверждение наличия всех необходимых версий программных пакетов на ресурсах T1 и СКЦ. Для этой цели были запущены базовые задания по реконструкции протон-протонных событий в трех центрах: ЦЕРН, T1 НИЦ КИ и суперкомпьютере НИЦ КИ.

После успешного завершения пилотных задач на ресурсах Курчатовского института в необходимой версии программной среды Athena (программная среда эксперимента ATLAS для физического кода), требования к реконструкции событий были изменены. Дополнительно к базовой реконструкции было добавлено требование о восстановлении полной информации о треках частиц в детекторе

переходного излучения. Данное введение было применено для полного соответствия задач реальным физическим задачам в группе ATLAS TRT.

Временные тесты были проведены с использованием моделированных данных. В качестве входных файлов были использованы данные, содержащие информацию о траекториях частиц в виде электронных сигналов, снимаемых с детекторов. Временные тесты, использующие 500 реконструированных событий в детекторе ATLAS, показали идентичные результаты для всех трех центров.

Проверка физических параметров продемонстрировали 100% согласие в выходных данных, полученных для двух разных вычислительных архитектур в ЦЕРН (T0), НИЦ КИ (T1) и НИЦ КИ (СК). Для данной проверки использовались комплексные TRT переменные, согласие в распределении которых может быть достигнуто только при условии совпадения распределений многих кинематических величин, таких как импульс частицы, псевдо быстрота и т.п. На рисунке 80 представлено одно из таких распределений: доля треков частиц, имеющих более 19 хитов (сработавших пропорциональных счетчиков) в TRT в зависимости от загрузки детектора.

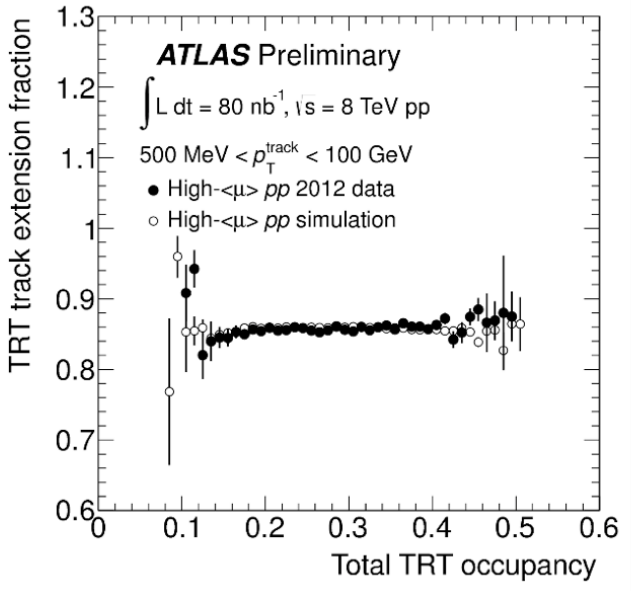


Рисунок 80- Доля треков частиц, имеющих более 19 хитов (сработавших пропорциональных счетчиков) в TRT в зависимости от загрузки детектора

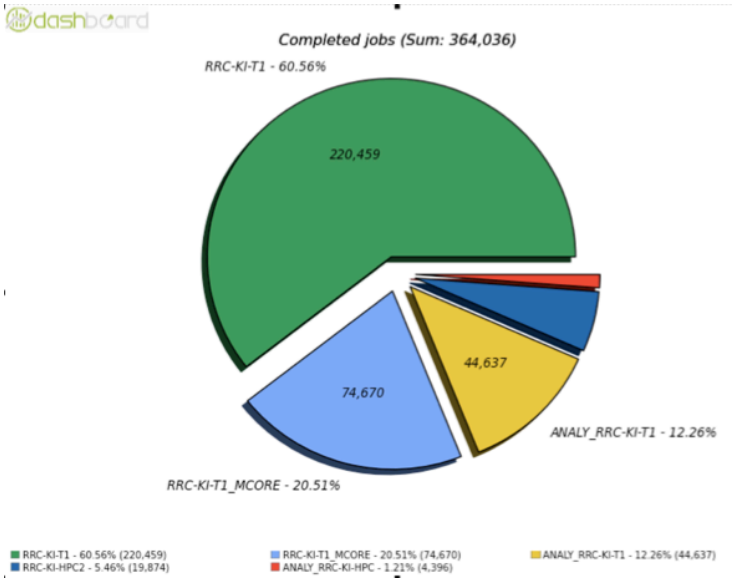


Рисунок 81- Статистика выполнения заданий ATLAS, с использованием вычислительных ресурсов НИЦ КИ

На втором этапе, после подтверждение идентичности физического результата, было необходимо интегрировать СКЦ и провести массовое моделирование и обработку данных. Для этого была создана специализированная *очередь выполнения заданий* (RRC-KI-HPC) с параметрами - 32 узла (256 вычислительных ядер, 2 ГБ оперативной и 1 ГБ swap-памяти на ядро) для выполнения задач моделирования и реконструкции. На рисунке 81 показана статистика выполнения заданий ATLAS, с использованием вычислительных ресурсов НИЦ КИ (суперкомпьютерные очереди имеют префикс HPC). Эта работа позволила начать масштабную интеграцию суперкомпьютеров с центрами обработки данных грид сначала для эксперимента ATLAS, а потом и для эксперимента ЯФ ALICE, и эксперимента на ускорителе SPS - COMPASS.

4.2.2 Реализация и использование системы управления загрузкой megaPanDA для приложений биоинформатики на суперкомпьтере НИЦ КИ

Демонстрация успешного решения при управления потоками заданий для приложений физики частиц, интеграция СК и их эффективное использование, а также повышение эффективности использования СК за счет выполнения заданий в “фоновом режиме” привлекли интерес научных сообществ за пределами LHC, ФВЭ и ЯФ, требующих значительных вычислительных ресурсов и высокоинтенсивных вычислений. Одним из таких примеров стало использование системы управления

загрузкой megaPanDA для решения задачи анализа данных геномного секвенирования шерстистого мамонта (анализ древней ДНК).

Древняя ДНК – генетический материал, извлеченный из древних биологических образцов. Первые исследования, с выделением и анализом древних фрагментов ДНК начались более 30 лет назад – первоначально работы проводились с небольшими участками митохондриального или ядерного генома.

К настоящему времени исследования в области древней ДНК все чаще используются для решения многих фундаментальных и прикладных вопросов. Возможность использования ДНК из археологического и палеонтологического материала позволяет решать многочисленные задачи, связанные с эволюцией экосистем в различных климатических условиях, с происхождением и эволюцией многих патогенных микроорганизмов.

В данной случае использовались геномные чтения шерстистого мамонта (Mammuthus primigenius), опубликованные ранее [99]. В настоящее время для анализа древней ДНК используются специальные программные пакеты-конвейеры, включающие ряд программных компонент, с помощью которых осуществляется быстрая обработка данных NGS. Одним из наиболее популярных программных конвейеров является пакет PALEOMIX[100]. Ранее опубликованные результаты, были получены на 80 ядерном сервере, имеющим 512 ГБ оперативной памяти. Получение результата заняло около двух месяцев. Анализ данных требовал 900M парных чтений. Срок в два месяца был обусловлен несколькими причинами:

- большое количество неавтоматизированных операций на этапе подготовки и выполнения программы секвенирования;
    - ручной запуск, и перезапуск задачи (в случае сбоя);
    - загрузка входных данных;
    - мониторирование времени исполнения задачи;
- особенности программного пакета PALEOMIX и метода его использования.

- отсутствия параллельного выполнения между несколькими узлами;
- отсутствие разбиения процесса выполнения на этапы;

– требование выделенного ресурса с аппаратными характеристиками (оперативная память, количество ядер).

Было предложено использовать созданную систему управления загрузкой, выделить задания, которые могут быть выполнены параллельно и автоматизировать процесс запуска и выполнения, т.е. применить разбиение входных данных (около 350 ГБ) на группы и провести “сборку” результатов в конце работы программного пакета PALEOMIX. Таким образом было введено параллельное выполнение заданий на уровне групп данных. И цепочка заданий стала аналогичной цепочке моделирования методом Монте-Карло для приложений ФВЭ и ЯФ. На первом этапе производится разбиение всего пространства данных на отдельные файлы, согласно логике определенной специалистами. Этот процесс оформлен в виде отдельного задания, выполняемой на одном узле. На втором этапе для каждого полученного файла запускается конвейер PALEOMIX как набор задач в рамках одного задания (аналоги *task* и *job,* рассмотренных ранее). Эти задачи могут выполняться в параллельном режиме на распределенной инфраструктуре. На третьем этапе при помощи конвейера PALEOMIX объединяются все результаты предыдущего этапа, что реализуется в одном задании.

Для демонстрации был использован СК НИЦ КИ и система управления заданиями megaPanDA. В результате весь процесс анализа данных занял четыре дня [98]. Тем самым было продемонстрировано, что программные средства и методы обработки больших объемов экспериментальных данных, которые были разработаны в области физики высоких энергий для экспериментов на ускорителе LHC, могут быть успешно применены в других областях науки.

4.3 Роль суперкомпьютеров для научной программы экспериментов в области физики частиц

В главе 2 и предыдущих разделах данной главы мы рассмотрели методику и подходы при интеграции суперкомпьютеров с вычислительными мощностями грид. Такая интеграция позволила провести исследования, которые были отложены на неопределенный срок из-за отсутствия вычислительного ресурса WLCG. Причем исследования эффективности работы детектора переходного излучения не являются единичным примером. Группа под руководством профессора Р.В. Коноплича работала над исследованием редкого процесса: $pp \rightarrow x0 \rightarrow ZZ \rightarrow 4l$. Для этого необходимо было выполнить полное моделирование процессов физики элементарных частиц с участием бозона Хиггса, адаптированное в соответствии с требованиями коллаборации ATLAS, моделирование должно было учитывать специфику протекающих событий на генераторном уровне, адронизацию конечного состояния и особенности установки ATLAS. По причинам, выходящим за интерес обсуждения в данной работе, это исследование не было признано приоритетным координатором физической программы эксперимента и группе не была выделена квота для использования грид ресурсов, тогда было предложено использовать суперкомпьютерный ресурс, для демонстрации интеграции СК и грид на реальном физическом приложении, и показать прозрачность использования СК ресурса для системы управления загрузкой, и одновременно демонстрации возможности как дополнительный ресурс может быть использован для решения реальной задачи физики частиц. Было промоделировано более 15М событий (рисунок 82), результат исследований был признан настолько значительным, что привел к публикации научной работы в журнале Physics Letters B, и нескольким докладам на международных конференциях [101].

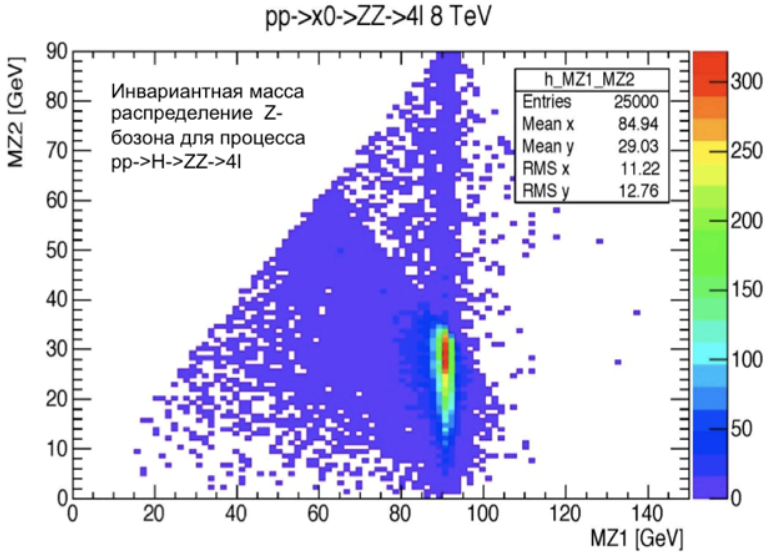


Рисунок 82 - Результат моделирования 15M событий для распада pp→x0→ZZ→4l

Разработанные методика, методы и архитектура позволили создать глобальную распределенную система для обработки данных. Реализация такой системы стала ключевым этапом для дальнейшего развития компьютерной модели и сделала возможным создание гетерогенной киберинфраструктуры, позволив использовать ресурсы суперкомпьютеров и ресурсы “облачных вычислений” наряду с существующей инфраструктурой грид, нивелировав архитектурные различия вычислительных мощностей. На рисунке 83 схематично представлена компьютерная модель, принятая после интеграции ресурсов грид с ресурсами суперкомпьютеров, облачными ресурсами и ресурсами университетских кластеров. Созданная система распределенной обработки данных имеет беспрецедентную эффективность (более 2M задач, выполняемых ежедневно). На рисунке 84 представлен график, показывающий количество задач, выполненных на суперкомпьютерных ресурсах в 2014/2016 годах. В среднем выполняется более 115 тысяч задач в день, а из потоков заданий наиболее популярными являются задачи моделирования и анализа. Вклад HPC в компьютерный бюджет эксперимента ATLAS исчисляется миллионам ЦПУ часов в месяц.

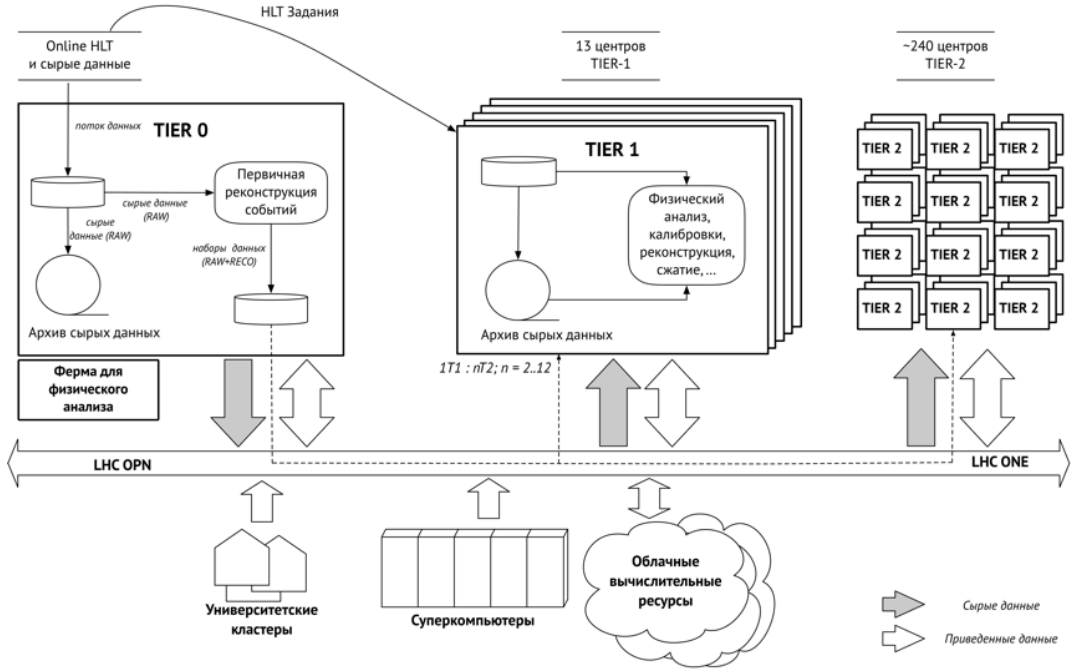


Рисунок 83 – новая компьютерная модель реализованная для второго и последующих этапов работы LHC

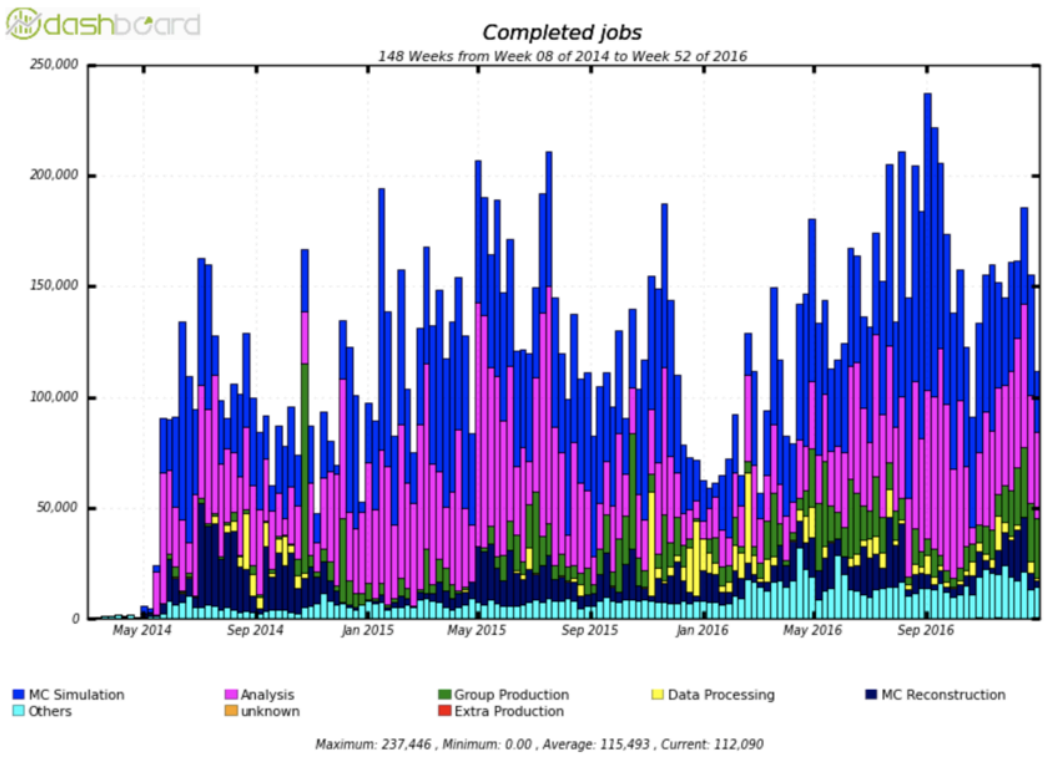


Рисунок 84 - Количество задач, выполненных на ресурсах HPC в 2014/2016 годах

4.4 Архитектурные принципы, методы и технологии при создании географически распределенного федеративного дискового пространства в рамках гетерогенной киберинфраструктуры

В данном разделе мы рассмотрим базовые принципы и тенденции при управлении и хранении данных физического эксперимента и как это влияет или может повлиять на развитие компьютерной модели. Каковы основные положения и требования со стороны физических экспериментов к системам хранения данных :

- цена данных может быть рассчитана, исходя из стоимости строительства и ввода в эксплуатацию научной установки и ускорителя, стоимости эксплуатации установки и ускорителя, нормированными на время работы ускорителя.
    - цена данных может составить до тысячи долларов за секунду работы ускорителя;
    - в случае астрофизических экспериментов цена может быть гораздо выше (из-за затрат по запуску эксперимента в космос);
- эксперименты не могут доверить свои данные и ответственность за них третьей стороне (на это есть как научные, так и социологические причины, не говоря о цене такого решения);
- места хранения данных и места их обработки до последнего времени были жестко связаны между собой;
    - расширение компьютерных ресурсов за счет суперкомпьютеров и ресурсов “облачных вычислений” сделали эту зависимость менее явной, но для гарантированного ресурса - данное утверждение справедливо практически для всех центров уровня T0, T1 и T2;

  - такое решение ограничивает выбор третьей стороны, которая могла бы предоставлять вычислительный ресурс на коммерческой основе;
- данные распределены между многими (O(100)) центрами по всему миру;
- сценарий работы желательный для физических экспериментов:
  - данные хранятся все время, но эксперименты оплачивают их обработку только когда они используются;
  - данные должны динамично доставляться в точку, где они будут обрабатываться;
- у нас есть данные различных классов, возможно применение различных технологических и архитектурных решений для данных, используемых в физическом анализа и при архивировании (презервации) данных.
- информация о локализации данных требует существенного упрощения, и “укрупнения” (поиск данных по 100 центрам и управление всеми копиями, не оптимален уже сегодня и ведет к задержкам при обработке и анализе данных).

Но основной проблемой остаются сценарий, когда петабайты данных передаются между сотнями ВЦ, поэтому предложенная идея “всемирного облака” при управлении потоками заданий и динамическому созданию “облака ресурсов” заставляет искать аналогичное решение для организации хранения и управления данными.

В предыдущих главах была подробно рассмотрена эволюция компьютерной модели для экспериментов на Большом адронном коллайдере. Объемы данных в течение первого рабочего пуска LHC (Run1 2009-2013) составили сотни петабайт для двух ведущих экспериментов в области физики высоких энергий (ATLAS) и ядерной физики (ALICE). С увеличением энергии и светимости LHC во время второго рабочего пуска (Run2 2015-2018) и этапа superLHC количество данных возрастет в 10 и 100 раз соответственно (рисунок 32). Это требует нового подхода к организации

вычислительных мощностей и дисковых ресурсов. В предыдущих главах было рассмотрено как создание нового поколения системы управления потоками заданий способствовало созданию гетерогенной вычислительной инфраструктуры и привело к использованию коммерческих и академических ресурсов облачных вычислений (ATOS, Helix Nebula, GCE, Amazon EC2, NECTAR) и суперкомпьютеров (Titan, Mira, HPC2 НИЦ КИ, Anselm, и др), что позволило использовать дополнительные ресурсы для моделирования и анализа данных. Следует отметить, что во всех перечисленных случаях это был дополнительный вычислительный ресурс с минимальным использованием дискового пространства, предоставляемого коммерческими фирмами (в силу его стоимости) или суперкомпьютерными центрами. Рассмотрим возможное архитектурное решение при создании географически распределенной федеративной системы хранения информации с единой точкой доступа и собственной системой внутреннего управления.

Основными требованиями к федеративной системе хранения информации являются :

- для пользователя система должна выглядеть как единое целое, в то время как представляет собой географически распределенные дисковые ресурсы;
- система должна обладать свойствами отказоустойчивости через дублирование ключевых компонент;
- система должна обладать масштабируемостью с возможностью изменения топологии без остановки работы всей системы;
- информационная безопасность системы должна обеспечиваться взаимной авторизацией и аутентификацией доступа к данным и метаданным;
- оптимальность доступа к данным и оптимизация передачи данных, требует оптимальности передачи данных с предоставлением клиенту

доступа к данным напрямую на ближайшем (оптимальном) конечном сервере;

- универсальность, подразумевающая применимость для широкого спектра научных проектов различного масштаба, включая, но не ограничиваясь экспериментами на LHC.

При большом разнообразии существующих технологий систем хранения данных в грид-среде: CASTOR, dCache, DPM, EOS, Xrootd [52,102-106] не многие из них отвечают перечисленным выше требованиям. **CASTOR и DPM** были разработаны в ЦЕРН. Обе системы были популярны в начале проекта WLCG, у обеих систем был один архитектор. DPM наследовал многие решения CASTOR, но в настоящее время системы находятся в “замороженном” состоянии и их дальнейшее развитие и добавления новых функций практически не проводятся. К основным недостаткам систем можно отнести: CASTOR - отсутствие разделения на архивируемые и активные данные, в системе нет гарантии, что данные автоматически не мигрировали на ленту. Система DPM получила широкое распространение, как система хранения данных для T2 грид сайтов, система имеет внутренний каталог файлов (LFC - Local File Catalog), масштабируемость и скорость доступа к информации каталога не очевидны, и не изучались за пределами необходимыми для центра уровня T2 (ни один из 13 сайтов уровня T1 не использует систему DPM. Сайты T1, ранее использующие систему хранения CASTOR, переходят на другие системы хранения данных). Таким образом, обе системы не могут быть рассмотрены в силу ограничений их функциональных возможностей.

Система хранения данных EOS. Идея разработки EOS связана с требованием разделить классы данных и виды активности (например: архивирование и анализ данных, рассматриваются как два различных вида активности). Пилотный проект EOS был впервые представлен на рабочем совещании WLCG в июле 2011 года Др.Дирком Дуельманом (ЦЕРН) и совместно с автором диссертации на международной конференции Grid2012, пилотный проект назывался - LST (Large

Scale Test for pool project) [107]. К этому моменту стало понятно, что скорость набора данных на LHC и предоставление доступа к данным тысячам пользователей, требуют изменения подхода к хранению данных, а входные данные для задач анализа и “сырые” данные не обязательно должны храниться с использованием одинаковой технологии. Так для задач анализа главным критерием является производительность, а при хранении “сырых” данным главным критерием является надежность. Надо отметить быстрое развитие пилотного проекта, и введение в эксплуатацию системы хранения EOS. К 2013 году EOS стал основной системой для хранения неархивированных данных в ЦЕРН. Основными особенностями и характеристиками системы являются :

- эффективный доступ к метаданным (до ~ 100 КГц при чтении имён файлов) в сочетании с ограниченным размером требуемой памяти (например, 100 миллионов имён файлов требуют 128 ГБ памяти для их хранения);
- поддержка четырех систем аутентификации: SSS, Unix, KRB и GSI;
- система квот для пользователей, групп пользователей и виртуальных организаций на занимаемое пространство;
- контрольные суммы по файлам и контрольные суммы по блокам;
- возможность дополнительного увеличения копий файлов и режим гарантированной доступности файла на основе программной RAID-1 репликации;
- специализированные интерфейсы для пользователя и системного администратора;
- введение понятия “популярность” файлов (учет количества обращений к файлу);

– поддержка со стороны разработчиков базового кода и их заинтересованность в развитии системы.

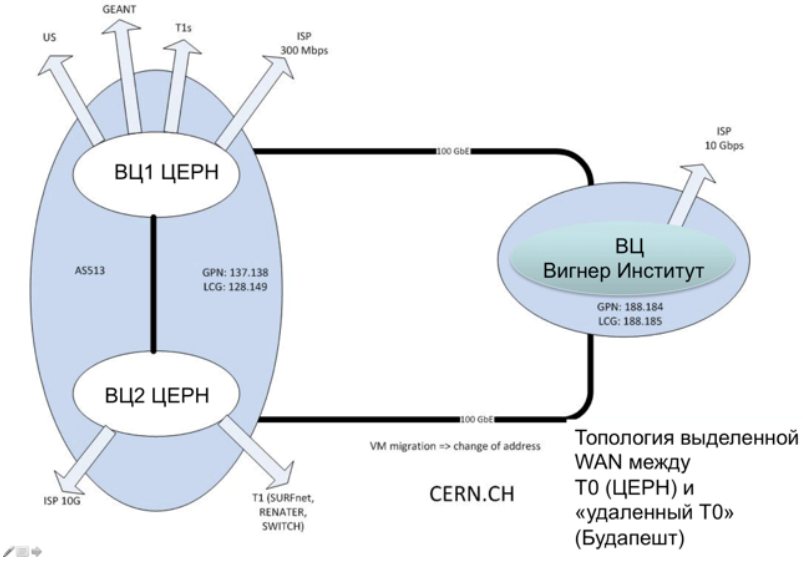


Рисунок 85 - Топология выделенной WAN между центрами ЦЕРН и Институт Вигнера при реализации федерации EOS между двумя центрами

EOS не поддерживает систему архивирования, что не является критичным для случая федеративного дискового пространства, т.к. изначально предполагается, что федеративное хранилище будет использоваться для хранения данных необходимых для физического анализа. Перед началом второго этапа работы LHC ЦЕРН создал географически распределенную систему хранения данных на основе EOS, которая располагается в двух центрах обработки данных, находящихся в Швейцарии (Женева) и Венгрии (Будапешт). Эта связка центров получила название Т0 - “удаленный Т0”. На рисунке 85 показана топология выделенной WAN между двумя центрами. Центры связывают две выделенные линии в 100 Гбайт. Административный домен для обоих центров одинаков (cern.ch), а локальные команды системных администраторов, поддерживающих аппаратные и сетевые ресурсы, являются совершенно различными. Как показывает опыт ЦЕРН, такое разделение не влияет на предоставляемый уровень сервиса (а кластеры EOS, о которых идёт речь, используются в режиме промышленной эксплуатации всеми четырьмя экспериментами на Большом адронном коллайдере).

Система хранения данных dCache. Проект dCache имеет более чем 15 летнюю историю. Во главе проекта и его основу составляет группа разработчиков

центра DESY (Гамбург, Германия) во главе с др. Патриком Фурманом. Первоначальная мотивация проекта была связана с созданием системы способной управлять набором дисковых пулов с балансировкой нагрузки. dCache самая популярная система среди центров консорциума WLCG, подавляющее число центров уровня T1 (11 из 13) и большинство центров уровня T2 используют эту систему. dCache обеспечивает хранение более 50% данных LHC экспериментов. Например страны северной Европы (грид консорциум Nordic Data Grid Facility (NDGF)) предложили и реализовали решение о создании распределенного центра T1, унифицировав используемые аппаратно-программные средства и создав единую дисковую федерацию для хранения данных: кластер dCache в NDGF состоит из централизованных элементов для управления и, предоставляемых университетами, пулов для непосредственного хранения данных (T1 распределен между университетами Осло и Копенгагена, при этом система архивирования T1 находится только в Осло). Особенности системы схожи с особенностями системы EOS (т.к. EOS разрабатывался позже, то его функциональность во многом повторяла функциональность системы dCache). Отличительными особенностями dCache являются система балансировки и распределения нагрузки между пулами (серверами хранения файлов) и наличие слоя архивирования данных, с возможностью автоматической миграции заранее предопределенных “семейств” файлов на ленточный носитель (например, данные формата RAW). Пространство имен (метаданные о файлах) хранится в реляционной базе данных, доступ к которой возможен по протоколам: NFS и FTP. Как и EOS, dCache отличает высокая степень поддержки и сервисной помощи со стороны основной команды разработчиков.

Система хранения данных Xrootd. Начальная мотивация проекта Xrootd состояла в разработке системы хранения, где будут храниться данные в формате NTUP, и они будет использоваться пакетом анализа данных ROOT, являющегося де факто стандартом для приложений анализа и визуализации информации в экспериментах ФВЭ и ЯФ. В отличии от EOS и dCache, одной из основных

особенностей Xrootd является полная открытость кода и желание разработчика привлечь внешних участников в проект, обеспечив большую функциональность за счет простоты добавления новых модулей через стандартизованный интерфейс, плагины (от англ. plug-in) и определения API. В систему могут включаться сложные функции: аутентификация / авторизация, интеграция с другими системами, глобальное распределение данных и т.д. К основным свойствам Xrootd можно отнести балансировку нагрузки, проверку подлинности: аутентификацию и авторизацию, высокую гибкость и отказоустойчивость, простоту интеграции различных файловых систем. Несколько лет назад (2012/15 годы) стали популярны федерации на основе Xrootd с возможностью децентрализованного поиска данных и наличия динамических каталогов. Проверка этого решения при анализе данных для двух возможных вариантов:

- Xrootd федерация;
- Xrootd центр с удаленным доступом к данным;

показала преимущество второго варианта, федерация на основе Xrootd не могла обеспечить устойчивую и безотказную работу в масштабах необходимых для экспериментов на LHC, это также было связано с тем, что в отличии от командных проектов: EOS и dCache, Xrootd - является индивидуальным проектом.

Развитие и состояние российской грид-инфраструктуры. Участие российских университетов и лабораторий в проекте грид, имеет более чем десятилетнюю историю. В рамках совместных работ с ЦЕРН и ЕС был создан проект RDIG (Russian Data Intensive Grid - Российский грид для интенсивных операций с данными) [108], успех проекта, его роль в развитии грид инфраструктуры в России и вкладе Российских центров в получение новых фундаментальных результатов в физике частиц во многом связан с работами В.А.Ильина и В.В.Коренькова [109]. Созданная инфраструктура успешно использовалась на всех этапах обработки данных для экспериментов на LHC. Дальнейшее развитие RDIG и введение в 2014 году в его состав двух новых центров уровня Т1 (в ОИЯИ и НИЦ

КИ), а также опыт создания и поддержания Российского Грид сегмента показывает, что вычислительные центры (Грид сайты) внутри одной страны неоднородны и могут быть разделены на две категории:

- крупные сайты, которые могут поддерживать сложную грид инфраструктуру (как аппаратную, так и программную ее часть), необходимую для нужд обработки и анализа данных, включая стек системного обеспечения, вспомогательные сервисы, программное обеспечение промежуточного уровня, сетевую инфраструктуру, сервисы хранения данных, сервисы экспериментов, и имеющие квалифицированный персонал : системные и сетевые администраторы, инженерные службы, эксперты в области обработки данных экспериментов, службы поддержки;
- сайты, созданные в небольших университетах и исследовательских институтах, которые не меняются во времени, не успевают выполнять необходимые требования по поддержанию и изменению аппаратно-программных средств, как того требуют эксперименты, не обновляют системное программное обеспечение и программное обеспечение промежуточного уровня. Как результат многие такие сайты перестают играть сколько-нибудь значимую роль в общей структуре WLCG/RDIG и не могут предоставить ресурсы необходимые для анализа данных ученым своих Институтов/Университетов, которые заинтересованы в обработке и анализе данных.

Данное разделение типично для многих национальных сегментов в рамках WLCG (например, среди 15 грид сайтов Германии наряду с такими крупными центрами, как DESY и КИТ (Карлсруэ), есть совершенно небольшие центры в Дрездене или Тюбингене). Создание единой вычислительной среды является сложной задачей, на первом этапе возможна организация общего дискового пространства внутри RDIG, а также общего дискового пространства между центрами уровня НИЦ КИ и ОИЯИ в России и ЦЕРН, DESY, GSI в Европе (все перечисленные

выше центры имеют/строят установки класса мегасайенс). При этом решается две разноплановые задачи :

- федерация в конфигурациях НИЦ КИ - DESY - ОИЯИ и НИЦ КИ-ЦЕРН - ОИЯИ будут интересны не только для экспериментов на LHC, но и для экспериментов на будущих ускорителях/установках. Общее дисковое пространство между центрами такого уровня позволит ученым выполнять задания в гетерогенной компьютерной среде с доступом к федерированным данным (при таком подходе также решается задача о передаче данных на хранение из ЦЕРН на хранение в Российские центры для экспериментов на LHC, и в будущем передача данных экспериментов на коллайдере NICA в Европейские центры обработки);
- федерирование ресурсов T1-nT2, для RDIG позволят более эффективно использовать Российский ресурс, и обеспечить прозрачный доступ к данным для их анализа Российским участникам экспериментов на LHC.

Для обоих сценариев решается фундаментальная проблема, задания на обработку и анализ данных не должны более проверять наличие данных в каталогах, данные должны находиться внутри федерации. При этом доступ к данным потоков заданий может быть организован не только из центров грид, но и из ресурсов облачных вычислений или суперкомпьютеров. Следует отметить, что постановка задач отличается от задачи, решенной ЦЕРН при создании “связки” T0-”удаленный” T0, который был создан на «эксклюзивной̆ однородной̆ связке центров» (ЦЕРН/Вигнер) и специально созданной компьютерной сети, в данном случае речь идет об использовании гетерогенной среды и трех различных центров (университетский̆ кластер, центр высокопроизводительных вычислений, центр высокоскоростных вычислений).

Общие выводы по выбору технологии для системы хранения при создания федеративного дискового пространства. Технологии, наиболее полно удовлетворяющие всем требованиям, и позволяющие реализовать обе задачи - это

системы хранения данных EOS и dCache. Системы обладают необходимой функциональной гибкостью и масштабируемостью, обе поддерживают различные протоколы доступа к данным, команды разработчиков обеих систем гарантируют их поддержку в течение длительного (десятилетие) периода времени, обе системы не являются ориентированными на приложения ФВЭ и ЯФ и/или определенный формат данных, что позволяет говорить о возможности использования систем для других научных приложений. Выбор был сделан в пользу системы EOS, и, пожалуй, одним из решающих аргументов явилось желание обеих команд (ЦЕРН/Россия (ОИЯИ, НИЦ КИ) разработать технологию, которая позволила в будущем федерировать дисковые ресурсы на основе разных технологий. Такой надстройкой может стать метауровень на основе динамической федерации HTTP (DynaFed,) [110]. DynaFed - разработка ЦЕРН, и она позиционируется как надстройка над существующими хранилищами на базе протокола HTTP/DAV и пока не используется экспериментами для повседневной работы, хотя сама идея заслуживает пристального внимания.

Методика создания федеративной архитектуры для географически распределенных дисковых ресурсов. Для проверки методики федерирования дисковых ресурсов и построения прототипа в рамках консорциума RDIG для решения двух различных задач были выбраны следующие центры :

Задача 1 : Будущие установки, в частности доступ к данным коллайдера NICA со стороны российских центров и международных участников проекта.

- Коллайдер NICA: ОИЯИ (T0), ЦЕРН, НИЦ КИ;

Задача 2 : Хранение данных LHC экспериментов в российских центрах, и доступ к данным, хранящимся в ЦЕРН.

- Эксперимент ATLAS:
  - НИЦ КИ (T1), ПИЯФ (T2),ОИЯИ(T2), МИФИ (T3), НИИЯФ (T3), ЦЕРН;
- Эксперимент ALICE:

- НИЦ КИ (Т1), СПбГУ (Т2), ПИЯФ (Т2), ОИЯИ(Т2), НИИЯФ(Т3), ЦЕРН;

Такой набор центров (рисунок 86) включает центры различной стабильности и с различным уровнем сервисного обслуживания и позволяет наилучшим образом промоделировать обе задачи, определенные ранее. Для мониторирования состояния WAN использовалась инфраструктура perfsonar (см главу 1) успешно примененная ранее при переходе к «смешанной компьютерной модели» для экспериментов на LHC. Пакет perfsonar также позволял проверить характеристики сетевого соединения между тестируемыми центрами. Эти характеристики позволяют понять зависимость тестов производительности от сетевой топологии и пропускной способности сетевых каналов, связывающих компоненты системы и исключить флуктуации, связанные с работой сетевого оборудования и WAN. Для тонкой настройки прототипа и проверки его аппаратной части использовалась методика синтетических тестов, разработанная и реализованная в соавторстве с А.К.Кирьяновым и А.К.Зароченцевым (Методика тестирования прототипа подробно рассмотрена в статье [111]). При моделировании работы прототипа федерации для реальных приложений ФВЭ и ЯФ, были выбраны реальные программы восстановления треков в детекторе переходного излучения ATLAS (автор Д.В.Краснопевцева) и программа фильтрации и отбора событий эксперимента ALICE (автор П.Христова). Первая из программ требует значительного вычислительного ресурса и интенсивного обмена информацией между оперативной памятью и диском после каждого шага реконструкции, вторая программа требует значительной производительности ввода/вывода при отсутствии требований к оперативной памяти и ЦПУ.

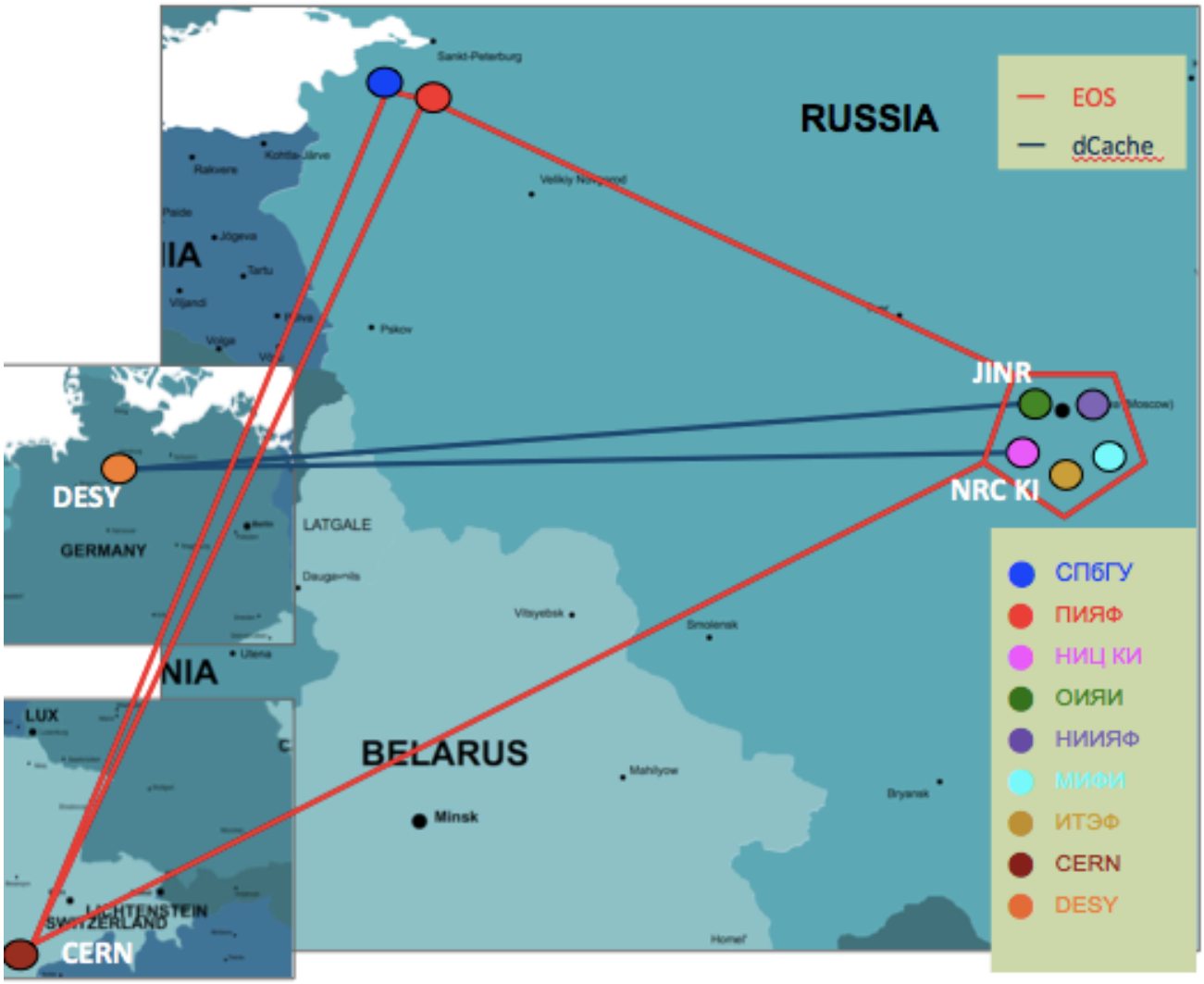


Рисунок 86 - Прототип федерации в рамках российского сегмента RDG

Прототип федерации центров RDIG на основе технологии EOS. При создании системы хранения необходимо было определить топологию хранилищ и общую систему авторизации. Учитывая, что ресурсы, предоставляемые участниками для данного прототипа, примерно эквивалентны, было решено использовать простейшую схему: установить в каждой организации по одному управляющему серверу (mgm) и одному серверу хранения данных (fst). Поскольку EOS не поддерживает одновременную работу нескольких одноранговых управляющих серверов в рамках одного сегмента, то в одной из организаций сервер mgm работает в режиме master (первичный), а в других организациях в режиме slave (вторичный),

обеспечивая автоматическую синхронизацию метаданных. Такое решение позволяет иметь единую точку входа через первичный управляющий сервер и повышает отказоустойчивость системы, т.к. с можно использовать один из вторичных серверов в режиме первичного, в случае его отказа.

В общем случае обращение клиента идет сначала к единой точке входа - управляющему или федеративному серверу верхнего уровня – на котором проходит авторизация, после чего появляется доступ к метаданным. Сервер верхнего уровня находит и передает клиенту указатель на физическое расположение файла (PFN) или выделяемого пространства, после чего запрос клиента перенаправляется по данному указателю на соответствующий сервер хранения, обеспечивая тем самым оптимальность передачи данных. Система EOS включает в себя поддержку четырёх стандартных систем авторизации: SSS, unix, krb и GSI. Системы SSS и UNIX не являются достаточно защищенными для их использования на удаленных серверах, и используются обычно для упрощения локальной настройки хранилища. Из оставшихся двух систем авторизации была выбрана GSI как наиболее надежная и широко применяемая в рамках глобального консорциума WLCG. Авторизация GSI подразумевает наличие у каждого участника (клиента или сервера) своего индивидуального цифрового сертификата в формате X.509, подписанного одним из центров сертификации (Certification Authority, CA). Окончательная конфигурация прототипа федерации и выбор ролей для центров участников был проведен на основе измерений пропускной способности WAN между всеми центрами (рисунок 87).

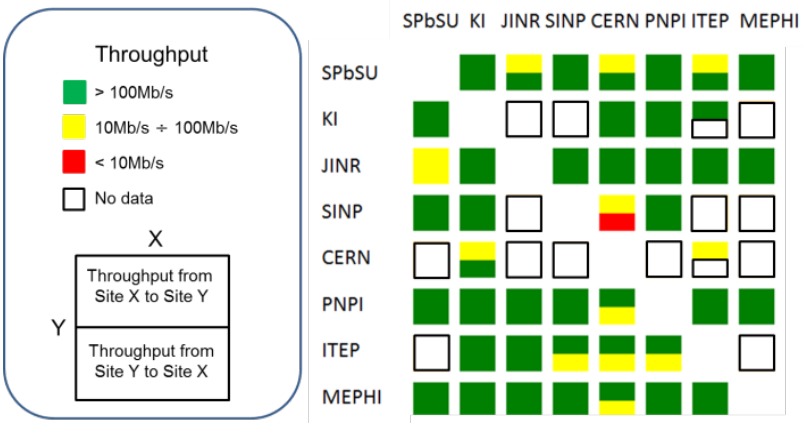


Рисунок 87 - Матрица эффективности при кросс-передаче данных между сайтами федерации

Из приведенной матрицы видно, что скорость передачи данных не симметрична. Такое поведение связано с тем, что сетевые линии, участвующих ресурсных центров не являются выделенными, и одновременно используются для передачи информации для других проектов. Это дает возможность протестировать федерацию в реальных условиях с насыщенными сетевыми каналами. При выборе ролей учитывалась стабильность работы центров в экспериментах на LHC (исходя из среднего времени доступности центра в течение календарного года, так для ЦЕРН, НИЦ КИ и ОИЯИ этот показатель превышает 95%, для ИТЭФ, МИФИ и НИИЯФ он составляет менее 75%. Последняя группа центров имеет доступ к федерации на чтение данных, но эти центры не могут являться головным сервером, что соответствует ситуации, когда центры используются только для физического анализа).

Проверка работы федерации дисковых ресурсов, созданной по технологии EOS, для приложений ФВЭ и ЯФ. Программа анализа данных эксперимента ATLAS представляет собой реконструкцию событий детектора при протон-протонных столкновениях. В качестве исходных наборов данных использовались, так называемые «сырые» события (формат RAW). Эта программа аналогична программе, которая использовалась для проверки интеграции СК НИЦ КИ с системой грид (раздел 4.2.1). Рисунок 88 демонстрирует результаты работы программы обработки данных детектора TRT ATLAS при распределении входных данных в нескольких

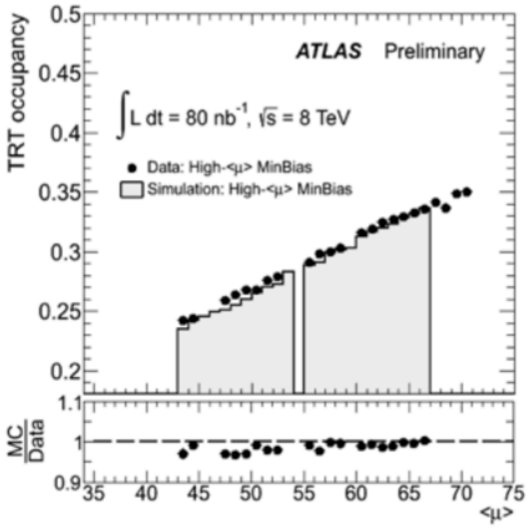


Рисунок 88 - Загрузка детектора переходного излучения как функция среднего количества столкновений в событии для всего детектора. Реальные данные (закрашенные черные точки) и моделирование

географически распределенных центрах внутри федерации. Особенностью этой задачи является требование на восстановление всей информации о треках элементарных частиц в детекторе переходного излучения. Восстановление сигнала от каждого пропорционального счетчика в TRT в условиях большой загрузки детектора увеличивает процессорное время на обработку событий. Реконструкция событий осуществляется в несколько этапов, каждый из которых требует интенсивного чтения и записи информации, хранимой в федеративной системе. Кроме того, программа обработки анализирует кинематические распределения для TRT. Этот позволяет провести проверку на совместимость результатов теста для классического и федеративного способов хранения данных. По завершению выполнения задачи ее журнальный файл содержит информацию о затраченных вычислительных ресурсах на трёх основных этапах выполнения задачи: инициализация среды для реконструкции, непосредственно последовательная обработка событий и окончание работы и закрытие выходных файлов. Дополнительно в ходе выполнения задачи в журнальный файл записывается информация о времени (астрономическом и процессорном), затраченном на обработку отдельного события. (Общее количество задач анализа и реконструкции TRT выполненных в рамках тестирования прототипа составило 43 тысячи).

Программа отбора и фильтрации данных эксперимента ALICE последовательно читает события из файлов, анализирует их и в качестве выходных данных предоставляет статистику работы программы и информацию о наиболее интересных событиях, а также файл с выбранными событиями. Эта задача позволяет оценить производительность доступа к системе хранения для приложений с интенсивным чтением/записью информации. При этом входной набор данных был расположен на федеративном и локальном хранилищах.

Для моделирования поведения работы прототипа с использованием приложений экспериментов ATLAS и ALICE были развёрнуты стандартные пользовательские интерфейсы, содержащие весь стек программного обеспечения:

сертификаты и пакеты CA для авторизации пользователя на системе хранения, клиент для системы хранения EOS, а также программное обеспечение самих тестов: Bonnie++ [112] для синтетического теста и систему доступа к программному обеспечению экспериментов LHC для тестов этих экспериментов (CVMFS), что полностью соответствует классической обработке данных, принятой в обоих экспериментах.

Во время теста реконструкции событий ATLAS входные файлы необязательно должны быть доступны в локально смонтированной файловой системе и могут быть прочитаны удаленно через протокол xrootd. Поэтому в результатах этого теста имеется еще и параметр: тип доступа, которым является либо протокол FUSE [113], либо протокол xrootd. Зависимость времени выполнения программ от различных комбинаций клиент-сервер и протоколов доступа к данным показана на рисунке 89 (левый график соответствует программе эксперимента ALICE, правый график - ATLAS) .

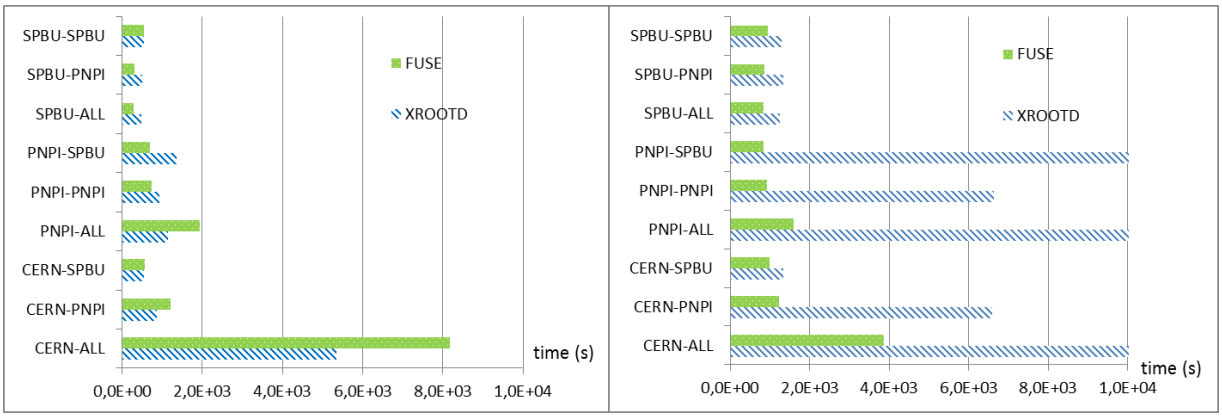


Рисунок 89 - Зависимость времени выполнения программ ATLAS и ALICE для различных комбинаций клиент-сервер и протоколов доступа к данным

Распределение входных наборов данных было сделано для трех возможных сценариев :

1. Данные распределены случайным образом между центрами (согласно ролям, рассмотренным выше);

2. Все данные находятся в НИЦ КИ или ОИЯИ (соответствует сценарию для экспериментов на будущем коллайдере NICA, или LHC);
3. Полная копия набора данных находится в НИЦ КИ или ОИЯИ (а вторая копия распределена случайным образом между центрами, согласно их ролям);

На рисунке 90 представлены результаты работы программы фильтрации событий для эксперимента ALICE при трех сценариях распределения данных (голубой цвет: сценарий 1, черный цвет : сценарий 2, зеленый цвет : сценарий 3), по оси ординат - имя сайта, на котором выполнялась программа, по оси ординат - скорость чтения в МБ/сек. На левом графике программа фильтрации событий выполнялась на всех центрах одновременно, на правом поочередно на каждом из них.

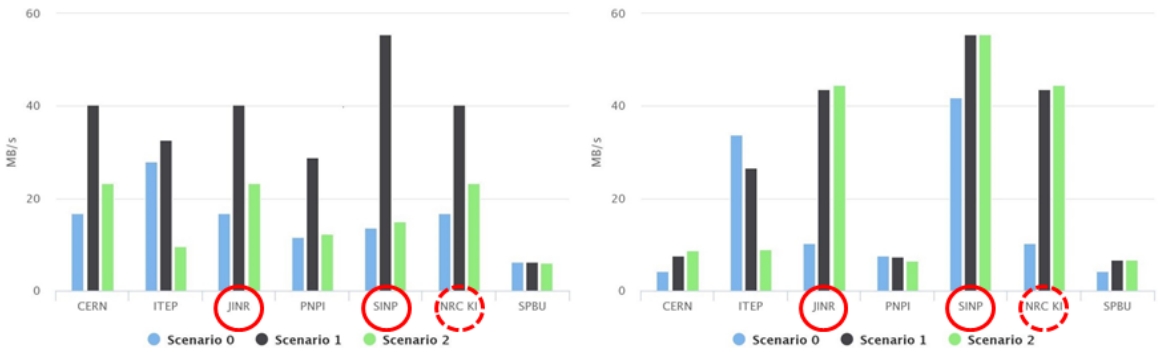


Рисунок 90 - Зависимость эффективности доступа к данным при различных сценариях распределения данных между центрами федерации

Из графика видно, что доступ к наборам данных (и анализ данных), находящимся в федерации может быть успешно осуществлен со всех сайтов. Неудивительно, что решающую роль играет качество WAN, например, клиент, выполняемый на сайтах ОИЯИ и НИИЯФ, эффективно работает с удаленными данными и/или с набором данных, находящимся в НИЦ КИ.

Выводы к созданию прототипа географически распределенной федерации. Создан первый прототип географически распределенного хранилища данных, состоящего из центров ЦЕРН и RDIG. Было продемонстрировано, что такая система

хранения может быть эффективно использована для обработки и анализа данных научными приложениями LHC. Моделирование работы федерации было проведено для реальных ВЦ, входящих в RDIG, и реальных научных приложений экспериментов ATLAS и ALICE, для различных сценариев распределения наборов данных. Такой подход позволяет эффективно использовать небольшие ВЦ (например в ИТЭФ, НИИЯФ МГУ, МИФИ), одновременно предоставив ученым прозрачный доступ к научным данным. Переход топологии RDIG и дальнейшее федерирование дискового ресурса позволят снизить операционные расходы и более эффективно использовать вычислительные ресурсы. Эти исследования будут важны для принятия новой компьютерной модели и создания объединенной киберинфраструктуры для эпохи superLHC, а также для экспериментов на новых комплексах, таких как XFEL и NICA. Результаты работ были представлены на многих международных конференциях и опубликованы в 2015/16 годах, по результатам обсуждения было принято решение о создании исследовательского проекта для экспериментов на LHC под руководством автора диссертации и Др.Д.Дуельмана (ЦЕРН ИТ).

Заключение

В диссертации обобщен многолетний опыт автора по разработке систем для распределенной обработки данных экспериментов ФВЭ и ЯФ, а также опыт по созданию и развитию компьютерной модели экспериментов в области физики частиц.

Исследования в области физики высоких энергий и ядерной физики невозможны без использования вычислительных систем, а также программного обеспечения для управления, обработки, моделирования и анализа данных. Это определяется рядом факторов: а) большими объемами информации, получаемыми с установок на современных ускорителях; б) сложностью алгоритмов обработки данных; в) статистической природой анализа данных; г) необходимостью (пере)обрабатывать данные после уточнения условий работы детекторов и ускорителя и/или проведения калибровки каналов считывания; д) необходимостью моделирования условий работы современных установок и физических процессов одновременно с набором и обработкой «реальных» данных.

Введение в эксплуатацию Большого адронного коллайдера, создание и запуск установок такого масштаба как ATLAS, CMS, ALICE, новые и будущие проекты класса мегасайенс (NICA, FAIR, XFEL, LSST), характеризующиеся сверхбольшими объемами информации, потребовали новых подходов, методов и решений в области информационных технологий.

Создание распределенной компьютерной модели для экспериментов в области ФВЭ и ЯФ стало одним из наиболее важных этапов развития компьютинга и изменило подход к работе с данными. Компьютерная модель обработки данных физического эксперимента прошла в своем развитии много этапов :

1. Иерархическая модель MONARC. Иерархическая модель распределенных вычислений, характеризующаяся иерархией центров: T0, T1,

T2, T3. Предопределением функций центров, статической организацией групп центров и “связок”: T1:nT2, статическим характером распределения данных между центрами с заранее определенным количеством копий.

2. Эволюция модели MONARC. Введение понятия популярности (востребованности) классов и наборов данных, переход к «смешанной компьютерной модели». Исследование популярности данных, разработка методов для определения стабильности работы центров уровня T1 и T2, оценка роли и возможностей глобальной вычислительной сети при обработке данных, а также создание методики и методов динамического распределения данных на основе их популярности, позволили отказаться от ограничений, введенных иерархической моделью, и позволили перейти к «смешанной компьютерной модели»;

3. «Смешанная модель». Создание «смешанной компьютерной модели» и оптимизация использования ресурсов всех центров, независимо от их классификации в рамках WLCG, в частности, переход к динамической организации групп центров для выполнения заданий, разработка методики динамического распределения ресурса, разработка принципиально нового поколения системы управления потоками заданий, привели к изменению парадигмы обработки и анализа данных, предоставив ученым по всему миру (в больших и малых центрах) гарантированный доступ к данным и вычислительным ресурсам, а также возможность активно участвовать в получении физических результатов;

4. Использование гетерогенных компьютерных ресурсов. Интеграция суперкомпьютеров, ресурсов «облачных вычислений», университетских кластеров и высокопропускных вычислений (грид) в единую инфраструктуру с прозрачным доступом ко всем ресурсам через систему управления загрузкой. Использование гетерогенной компьютерной инфраструктуры с динамическим разделением ресурса между различными потоками заданий и группами

пользователей. Создание прототипа дисковой федерации, и работа с данными в эксабайтном диапазоне.

На основе методики, методов и архитектуры, разработанных в диссертации, создана глобальная распределенная система для обработки данных на основе динамического управления потоками заданий и динамическим распределением данных с учетом пропускной способности WAN. Реализация такой системы стала ключевым этапом для дальнейшего развития компьютерной модели и сделала возможным создание гетерогенной киберинфраструктуры, что позволило использовать ресурсы суперкомпьютеров и ресурсы “облачных вычислений” наряду с существующей инфраструктурой грид, нивелировав архитектурные различия вычислительных мощностей. Таким образом, разнородные вычислительные ресурсы стали доступны пользователям в виде единой киберинфраструктуры.

Созданная система для глобальной распределенной обработки данных позволяет динамически организовывать группы ресурсов (“всемирное облако”) для выполнения научных заданий и динамически разделять вычислительный ресурс между различными классами заданий: Монте-Карло моделирование, обработка данных, анализ данных, потоки заданий отдельных научных групп. Система имеет уникальные характеристики, выполняя более 2М задач в день в 250 ВЦ. Использование системы для приложений в других научных областях, таких как астрофизика и биоинформатика, подтверждает ее универсальность.

Были исследованы, разработаны и реализованы базовые принципы подсистемы мониторирования, включая оценку времени выполнения заданий, и учета (аккаунтинга) работы отдельных центров. Были исследованы и реализованы методы определения популярности (востребованности) классов данных и отдельных наборов данных физического эксперимента.

Дальнейшее развитие компьютерной модели будет связано с использованием дополнительного вычислительного ресурса (включая суперкомпьютеры и

коммерческие вычислительные ресурсы) в момент пиковых нагрузок и переходом от отдельных центров к федеративной организации ресурсов.

Создание географически распределенной федерации в рамках российского сегмента грид (RDIG) стало важным шагом в развитии методологии создания распределенной киберинфраструктуры. Прототип федерации и демонстрация ее возможностей для реальных приложений ФВЭ и ЯФ явились важным шагом исследований компьютерной модели и ее дальнейшего и развития.

Основные результаты диссертационной работы.

1. На основе методики, методов и архитектуры, разработанных в диссертации, создана глобальная распределенная система для обработки данных на основе динамического управления потоками заданий и динамическим распределением данных с учетом пропускной способности WAN. Реализация такой системы стала ключевым этапом для дальнейшего развития компьютерной модели и сделала возможным создание гетерогенной киберинфраструктуры, позволив использовать ресурсы суперкомпьютеров и ресурсы “облачных вычислений” наряду с существующей инфраструктурой грид, нивелировав архитектурные различия вычислительных мощностей. Таким образом, разнородные вычислительные ресурсы доступны пользователям в виде единой киберинфраструктуры. Созданная система имеет уникальные характеристики и позволяет :
    a. динамически организовывать группы ресурсов (“всемирное облако”) для выполнения научных заданий и динамически разделять вычислительный ресурс между различными классами заданий: Монте-Карло моделирование, обработка данных, анализ данных, потоки заданий отдельных научных групп.
    b. выполнять до 2М задач в день в 250 ВЦ (1.4 Эбайта данных было обработано только в 2016 году).

   c. использовать систему для приложений других научных областей, таких как, астрофизика и биоинформатика, что подтверждает универсальность созданного ПО.
2. Исследованы, разработаны и реализованы базовые принципы подсистемы мониторирования, включая предсказание времени выполнения заданий, и аккаунтинга (учета работы отдельных центров и заданий).
3. Исследованы и реализованы методы определения популярности (востребованности) классов данных и отдельных наборов данных физического эксперимента.
4. Разработана методика управления научными приложения ФВЭ и ЯФ для суперкомпьютеров с использованием информации о временно свободных ресурсах, повышающая эффективность использования СК (реализация методики для СК Титан позволила повысить эффективность использования с 89% до 94%).

Основные результаты, выводы, рекомендации и архитектурные решения, изложенные в диссертации, использовались при реализации следующих национальных и международных проектов:

- эксперименты ATLAS и ALICE на LHC в ЦЕРН (распределенная обработка, моделирование и анализ данных);
- эксперимент COMPASS на ускорителе SPS в ЦЕРН;
- проект развития глобальной грид-инфраструктуры для LHC (WLCG);
- эксперименты AMS и AMS-02 на МКС;
- проект “Federated data storage” WLCG;
- проект по развитию и применению методов “машинного обучения” для обнаружения аномалий в работе сложных распределенных систем и исследования их работы WLCG;

- проект «Создание Лаборатории Технологий Больших Данных для проектов в области мегасайенс» в рамках реализации постановления 220 Правительства РФ.
- проект Российского научного фонда: «MetaMiner for BigData: Создание гетерогенной системы хранения метаинформации для научных экспериментов эксабайтного масштаба и применение методов "машинного обучения" для выявления нарушений при функционировании распределенных систем обработки и анализа Больших данных».
- проект Российского фонда фундаментальных исследований: “Исследование гетерогенных киберинфраструктур, разработка и создание прототипа компьютерной федерации на основе высокоскоростных вычислений, облачных вычислений и суперкомпьютеров для хранения, обработки и анализа Больших Данных”.

Основные результаты данной работы являются пионерскими и используются в действующих научных экспериментах. Уже сейчас результаты диссертации используются в двух крупнейших экспериментах в области ФВЭ и ЯФ: ATLAS и ALICE на LHC, эксперименте COMPASS на SPS, а также в приложениях биоинформатики на суперкомпьютерах НИЦ КИ.

Результаты диссертационной работы могут быть использованы при создании компьютерной модели для этапа высокой светимости LHC (этап superLHC), а также для новых комплексов, таких как FAIR, NICA, ErIC, и проектов класса мегасайенс: XFEL, LSST.

Другими перспективными направлениями данной работы являются: созданная система для глобальной распределенной обработки данных и федерирование географически распределенных дисковых ресурсов. Первое направление может быть использовано как метауровень для планирования потоков заданий на современных суперкомпьютерах, что, как показано в диссертации, повышает эффективность использования СК мощностей, второе направление интересно при рассмотрении

эволюции российского сегмента грид (RDIG) и предоставляет широкие возможности по оптимизации его инфраструктуры и повышению надежности его работы.

По материалам диссертации подготовлены и читаются лекционные курсы в НИЯУ МИФИ, ТПУ, МФТИ. Подготовлена магистерская программа в ТПУ и Университете “Дубна”.

Благодарности

Автор выражает глубокую благодарность своим учителям и наставникам профессорам А.В. Арефьеву, Ю.А. Камышкову, С.Ч.Ч. Тингу и Ю.В. Галактионову, своим коллегам по экспериментам ATLAS, AMS, L3 за плодотворную совместную работу. Коллективу Лаборатории “Технологии больших данных для экспериментов класса мегасайенс” НИЦ “Курчатовский институт” и заместителю директора НИЦ КИ В.Е. Велихову за плодотворную совместную работу, своей маме Климентовой Нине Васильевне.

Профессору В.В. Коренькову, научному консультанту, коллеге и другу, за его поддержку исследований по теме диссертации и плодотворную совместную работу в течение более 10 лет по темам, связанным с разработкой и созданием архитектур систем для обработки данных и программного обеспечения для современных физических экспериментов.

Сотрудникам НИЦ КИ и ОИЯИ П.В.Зрелову, М.А.Григорьевой, В.А. Ильину и А.И.Малахову за их интерес к работе, конструктивные замечания, рекомендации и обсуждение результатов диссертации.

Профессору В.В.Иванову за плодотворное обсуждение первых результатов диссертации и планов по разработке системы обработки и анализа данных эксабайтного диапазона на конференциях NEC, что положило начало сотрудничеству автора с ЛИТ ОИЯИ.

Профессору И.М.Иванченко за обсуждение истории развития систем сбора и обработки данных экспериментов в области ФВЭ и ЯФ и разработок ПО в ОИЯИ и ЦЕРН.

Сотрудникам НИЦ КИ О.С. Ковалевой, М.А. Титову, Е.А. Тужилкиной и Е.Ю. Щукиной за помощь в работе над диссертацией и подготовке к изданию диссертации и автореферата.

Перечень принятых сокращений и наименований

БАК	Большой адронный коллайдер, ускорительный комплекс ЦЕРН
ВЦ	Вычислительный центр
ДНК	Дезоксирибонуклеи́новая кислота́, одна из трех основных макромолекул
ЕС	Европейский Союз
ИВК	Информационно-вычислительный комплекс
ИС	Информационная система
ИТ/IT	Информационные технологии
КХД	Квантовая хромодинамика — калибровочная теория квантовых полей, описывающая сильное взаимодействие элементарных частиц
Лаборатория “Больших Данных”	“Лаборатория Технологий Больших Данных для проектов в области мега-сайенс» НБИКС НИЦ КИ
ЛИТ	Лаборатория информационных технологий ОИЯИ.
МКС	Международная космическая станция.
НБИКС	Комплекс нано-, био-, информационных и когнитивных технологий НИЦ КИ
НИР	Научно-исследовательская работа
НИЦ КИ	Национальный исследовательский центр «Курчатовский институт»
ОИЯИ	Объединенный институт ядерных исследований (Дубна)
ОО	Объектно-ориентированный подход при создании

	ПО
ПО	Программное обеспечение
СК	Суперкомпьютер
СКЦ	Суперкомпьютерный Центр
СУБД	Система управления базами данных
Т0	Tier0 (Тир0) головной центр в иерархии центров WLCG. Таким центром является ЦЕРН
Т1, Т2, Тх	Tier1/2/x : центр первого (второго,...) уровня в иерархии центров WLCG
Т1 НИЦ КИ	Гридовский сайт первого уровня, входящий в консорциум WLCG
Теватрон	ускоритель заряженных частиц в Лаборатории Ферми, Чикаго, США.
У70	Кольцевой ускоритель в Институте физики высоких энергий (Протвино)
ФВЭ	Физика высоких энергий
ФермиЛаб	Fermi Lab., национальная лаборатория имени Ферми в Чикаго, США
ФРКИ	Федеративная распределенная киберинфраструктура
ЦЕРН	Европейская организация по ядерным исследованиям
ЭБайт, экс(з)абайт	Эксабайт (экзабайт) (10^{18} байт)
ЯФ	Ядерная физика
AGIS	ATLAS Grid Information System, информационная система грид эксперимента
AGS	Alternative Gradient Synchrotron , ускоритель в

	Брукхейвенской Национально Лаборатории США
AJAX	Подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в «фоновом» обмене данными браузера с веб-сервером
AlFa	Новое поколение фреймворка для базового физического кода, совместная разработка эксперимента ALICE и специалистов ИТ центра FAIR в Дармштадте, Германия
ALICE	‘A Large Ion Collider Experiment’, тяжелоионный эксперимент на БАК
AliEN	Система управления загрузкой эксперимента ALICE в среде грид
AliRoot	Фреймворк для базового физического кода в эксперименте ALICE
AMS-01	‘Alpha-Magnetic Spectrometer’, астрофизический эксперимента на космическом челноке Shuttle (STS91) в 1998 году
AMS-02	‘Alpha-Magnetic Spectrometer’, астрофизический эксперимента на МКС
AOD	Analysis Object Data, формат приведенных данных в ФВЭ и ЯФ, используемых для физического анализа
APF	Autopilot factory, система запуска пилотных заданий
API	Application programming interface, интерфейс прикладного программирования

ARC	ATLAS Resource Control, программный модуль, разработанный в консорциуме NDGF, для доступа к ресурсам консорциума
Athena	Фреймворк для базового физического кода в эксперименте ATLAS
AthenaMP	Athena Multi-Process фреймворк, развитие фреймворка Athena, версия поддерживающая многоядерность
ATLAS	‘A Toroidal Lhc ApparatuS’, один из двух универсальных экспериментов на БАК
BaBar	Эксперимент ФВЭ на ускорителе SLAC
Blue Brain	Международный проект по моделированию работы мозга, и создания модели работы мозга человека. Проект использует СК по всему миру
BNL	Brookhaven National Laboratory, национальная лаборатория в США, также T1 центр для эксперимента ATLAS (~22-24% всех выделенных коллаборации ресурсов грид)
CASTOR	CERN Advanced Storage Manager, гибридная иерархическая система хранения данных
CERN	см. ЦЕРН
CDF	Эксперимент ФВЭ на ускорителе Tevatron
CE	Computing Element, один из элементов грид инфраструктуры, предназначенный для выполнения вычислительных задач пользователей
CRIC	Computing Resource Information Catalog, второе поколение информационной системы AGIS

CMS	‘Compact Muon Solenoid’, один из двух универсальных экспериментов на БАК
COMPASS	Эксперимент ФВЭ на суперпротонном ускорителе ЦЕРН
Condor-G	Менеджер ресурсов, доступных в среде Грид
D0	Эксперимент ФВЭ на ускорителе Tevatron
DAOD	Derived Analysis Object Data формат совместимый с форматом AOD, и получаемый путем выборки отдельных событий согласно критериям (например маска триггера высокого уровня), используются для физического анализа данных
DDM	Distributed Data Management, система управления данными при распределенной модели обработки данных
DESC	Dark Energy Science Collaboration, научное сообщество по поиску “темной материи” на основе исследований, проводимых на телескопе LSST
DST	Data Summary Tape, формат приведенных данных в ФВЭ и ЯФ, как правило в DST формате записываются результаты работы программы реконструкции событий (см также ESD)
DTN	Data Transfer Node, в суперкомпьютерной архитектуре это узел, имеющий доступ к интернет
DUNE	Deep Underground Neutrino Experiment, международный эксперимент, планируемый на базе ускорителя в Лаборатории Ферми США. Сам

	эксперимент изначально будет установлен в ЦЕРН (т.н. этап protoDUNE), а в дальнейшем в шахте штата Северная Дакота, США
EC2	Elastic Compute Cloud, коммерческий сервис компании Amazon для проведения облачных вычислений
EGEE	Европейский проект развертывания грид-систем для научных исследований –Enabling Grids for E-science in Europe
EGI	European Grid Initiative, один из трех консорциумов входящих в WLCG
EPFL	Ecole Polytechnic Federale de Lausanne : Политехнический Институт в Лозанне (Швейцария), один из ведущих технический Университетов в мире
ESD	Event Summary Data, формат приведенных данных в ФВЭ и ЯФ, как правило в ESD формате записываются результаты работы программы реконструкции событий. До работы коллайдера LHC - этот формат как правило назывался DST (data summary tape)
FAIR	Facility for Antiproton and Ion Research, будущий ускоритель в GSI ориентированный на тяжелоионные и антипротонные исследования
FTS	File Transfer Service, пакет программ для передачи файлов между центрами обработки БАК, FTS3 - третье поколение пакета FTS

G4	см Geant4
GAUDI	Проект ЦЕРН по разработке единого подхода к созданию фреймворков для экспериментов на LHC
GCE	Google Compute Engine, платформа облачных вычислений компании Google
Geant4	Четвертое поколение пакета программ моделирования (Geant) широко используемого в эксперимента ФВЭ и ЯФ для моделирования электроники, детекторов и физических процессов. В последние году этот пакет также используется для приложений биоинформатики и ядерной медицины
GSC	Ground Space Computers, станции AMS-02, установленные непосредственно внутри периметра центра MSFC NASA, штат Алабама, США
GSI	Центр имени Гельмгольца для исследований физики тяжелых ионов, расположен в г.Дармштадт, Германия
JIRA	ПО, используемое для учета этапов разработки, отладки и фиксации ошибок большими командами разработчиков ПО
HENP	High Energy and Nuclear Physics (см ФВЭиЯФ)
HEP	High Energy Physics (см ФВЭ)
HLT	High Level Trigger, система отбора событий “высшего” уровня, производит окончательный выбор событий для записи на носители (диск/лента), и последующей обработки и

	физического анализа
HPC	High Performance Computing (суперкомпьютеры)
HTC	High Throughput Computing (грид)
HTML	HyperText Markup Language, язык гипертекстовой разметки
HTTP	Hypertext Transfer Protocol, протокол передачи гипертекста
HMSF	Hybrid Metadata Storage Framework - Гибридное хранилище метаданных
HS06	HEP SpecInt, см MIPS, дословно количество миллионов инструкций кода ФВЭ, характеристика производительности вычислительного узла
IP	Internet Protocol, межсетевой протокол
HITS	Формат данных полученных в результате отцифровки событий, произведенных методом Монте-Карло (RDO)
JEDI	Jobs Execution and Definition Interface, динамическая система запуска задач
L3	Эксперимент в области ФВЭ на LEP
LAN	Local Area Network, локальная вычислительная сеть
LCF	Leadership Class Facilities, группа суперкомпьютеров в национальных лабораториях США : MIRA (Аргонская национальная Лаборатория), Titan (национальная лаборатория в Оак Ридж), NERSC (национальная лаборатория Беркли), а также СК в Ливерморской

	национальной лаборатории
LEP	Large Electron-Positron collider, ускоритель в ЦЕРН в 1989-2000 годах
LHC	см. БАК
LHCb	Специализированный эксперимент в области ФВЭ на БАК
LHCOPN	LHC Optical Private Network, глобальная вычислительная сеть, связывающая центры первого уровня WLCG (Tier-1) с центром нулевого уровня (Tier-0, ЦЕРН)
LHCONE	LHC Open Network Environment, дополнительная (к LHCOPN) вычислительная сеть для связи центров разных уровней консорциума WLCG
LSF	Load Sharing Facility, система пакетной обработки заданий
LSST	Large Synoptic Survey Telescope, международный проект по созданию телескопа в Южной Америке (Чили) для исследований в области астрономии, стоимость проекта более 1 миллиарда долларов США
megaPanDA	Система управления потоком заданий и загрузкой, созданная в НИЦ КИ, см. Также PanDA и WDMS
MIPS	Million Instructions per Second, характеристика производительности вычислительного узла в миллионах операций в секунду (данная характеристика в 2005/08 годах была в ФВЭ заменена на HS06)

MLlib	Масштабируемая библиотека для среды Apache Spark , содержщая API для основных языков программирования
MonALISA	Monitoring Agents using a Large Integrated Services Architecture. Пакет программ созданный для мониторирования (и в начале для моделирования поведения) распределенных вычислительных систем. Пакет был создан в рамках проекта MONARC
MONARC	Models of Networked Analysis at Regional Centres for LHC Experiments http://monarc.web.cern.ch/MONARC
NorduGrid	Nordic Grid, проект развертывания грид-систем для научных исследований в странах Норвегии, Дании, странах Балтии, Украине, Швейцарии, Словении.
MSFC	Marshall Space Flight Center, центр NASA имени Маршалла в штате Алабама, США
MySQL	Реляционная база данных, со свободным кодом
NASA	National Aeronautics and Space Administration, космическое агентство США
NGE	Natural Generic Engineering
NoSQL	Non relational, not only relational, технологии баз данных на основе графов, и/или имеющих существенные отличия от моделей, используемых в традиционных реляционных СУБД с доступом к данным средствами языка SQL

NWS	Network Weather Service, специальный сервис для сбора и хранения информации о состоянии WAN в эксперименте ATLAS
NTUP	Табличный формат данных в ФВЭ и ЯФ, как правило в формате ROOT (см ROOT), данные в формате NTUP используются в физическом анализе
Objectivity	Коммерческая БД, созданная одноименной компанией. Позволяла решить вопрос временного/постоянного хранения для коммерческих приложений. Опыт использования Objectivity для хранения метаинформации и ввода/вывода данных для приложений ФВЭ и ЯФ был неудачным
ORACLE	Коммерческая реляционная база данных.
OSG	Open Science Grid, проект развертывания грид-систем для научных исследований в США
PanDA	Production and Distributed Analysis Workload Management System
PBS	Portable Batch System, система пакетной обработки
Phedex	Система управления данными эксперимента CMS
POCC	Payload Operations and Control Center, центр AMS-02 в ЦЕРН по сбору данных и телеметрии, контроля работы детектора на борту МКС
PS	Proton Synchrotron, кольцевой ускоритель в ЦЕРН
RAW	“сырые” (неприведенные) наборы данных, получаемые с экспериментальной установки,

	данные содержат адрес канала считывания и величину, полученную непосредственно с электроники считывания
RDIG	Russian Data Intensive Grid, проект развертывания грид-систем для научных исследований в России
RDO	Raw Data Object, формат данных, “сырые” данные полученные в результате моделирования методом Монте-Карло
RHIC	Reliativistic Heavy Ion Collider, коллайдеров тяжелых ионов в Лаборатории BNL (США)
RO	Read only, метод доступа к информации на чтение, не предполагает изменение информации
RW	Read write, метод доступа к информации на чтение и запись предполагает изменение информации (иногда удаление информации также относится к этой группе)
Rucio	Второе поколение системы управления данными эксперимента ATLAS
SAGA	Simple API for Grid Applications, фреймворк для работы задач пилотов с различными локальными системами пакетной обработки, Разработка университета Ратгерс, США
SDC	Software for Distributed Computing, лаборатория департамента ИТ в ЦЕРН
SE	Storage Element, один из компонентов грид инфраструктуры, предназначенный для хранения данных

SLAC	Stanford Linear Accelerator, линейный ускоритель в одноименной лаборатории. Калифорния, США
SLURM	Simple Linux Utility for Resource Management, менеджер ресурсов для для кластеров вычислительных узлов под управлением операционной системы Linux
SOC	Science Operations Center, центр AMS-02 по обработке данных в ЦЕРН.
Spark (Apache Spark)	Программный каркас с открытым исходным кодом для распределенной обработки неструктурированных или слабоструктурируемых данных
SPS	Super Proton Synchrotron, кольцевой ускоритель в ЦЕРН
SQL	Structured Query Language, язык структурированных запросов
SRM	Storage Resource Manager, протокол доступа к постоянному дисковому хранилищу данных
SSO	Single Sign-On Management, система аутентификации пользователей
SW	Software, программное обеспечение
SW&C	Software and computing, один из проектов в экспериментах ФВЭ и ЯФ, под общим руководством одного/двух координаторов
TDR	Technical Design Report, документ описывающий основные компоненты, функции и характеристики экспериментальной установки, или одной из ее

	подсистем. Компьютинг, как правило, рассматривается как одна из подсистем
Tevatron	см. Теватрон
Tier-0	Гридовский центр 0 уровня в классификации WLCG, таким центром является только ЦЕРН. Tier-0 используется для экспресс обработки данных и их постоянного хранения
Tier-1	Гридовский центр 1 уровня в классификации WLCG ,13 центров по всему миру. Tier-1 используются для обработки и анализа данных, а также для долгосрочного хранения данных
Tier-2	Гридовский центр 2 уровня в классификации WLCG, ~140 центров по всему миру. Tier-2 используются для анализа данных и их временного хранения
Tier-3	Гридовский центр 3 уровня в классификации WLCG, ~200 центров, включая центры в Университетах. Вычислительный ресурс Tier-3 не является гарантированным, и может предоставляться на добровольной основе
TRT	Transition Radiation Tracker, детектор переходного излучения установки ATLAS
TWIKI	ПО (разработка ЦЕРН) на базе платформ Web для хранения документации, инструкций, информации о проекте. Позволяет осуществлять контролируемый доступ к Web-страницам, в том числе их редактирование группами, ведущими

	работы над проектом.
VO	Virtual organization, понятие грид: совокупность институтов, университетов, групп, объединённых для решения общей задачи в режиме скоординированного использования распределенных вычислительных ресурсов, выделенных для данной виртуальной организации
WAN	Wide Area Network, глобальная вычислительная сеть
WDMS	Workload and Data Management System: система управления загрузкой, вычислительными ресурсами (потоком задач) и данными
WLCG	Worldwide LHC Computing Grid, консорциум, объединяющий грид ресурсы, предназначенные для работ на Большом Адронном Коллайдере
WMS	Workload Management System: система управления потоком задач
WP	Work packages; раздел работ в рамках проекта

Список литературы

1. LHC – The Large Hadron Collider, http://lhc.web.cern.ch/lhc
2. The ATLAS Collaboration, G. Aad et al., “The ATLAS Experiment at the CERN Large Hadron Collider”, Journal of Instrumentation, Vol. 3, S08003, 2008.
3. The CMS Collaboration, S. Chatrchyan et al. “The CMS experiment at the CERN LHC”, Journal of Instrumentation, Vol. 3, S08004, 2008.
4. ALICE Collaboration, K. Aamond et al., “The ALICE experiment at the CERN LHC”, JINST 3 (2008) S08002
5. H.H. Gutbrod et al. (Eds.) “FAIR Baseline Technical Report”, ISBN 3-9811298-0-6 Nov. 2006
6. M. Altarelli et al. (Eds). “XFEL: The European X-ray Free-Electron Laser Technical Design Report”, DESY 2006-097 (DESY, 2007)
7. G.V. Trubnikov et al. “Project of the Nuclotron-based Ion Collider Facility (NICA) at JINR”, Proceedings of EPAC 08 (Genoa, 2008), pp. 2581–2583.
8. The ATLAS Collaboration, G. Aad, A. Klimentov et al “Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC”, Physics Letters B, 716, 2012, pp 1-29
9. J. Ratchford, U. Colombo, “Megascience,” UNESCO World Science Report, 1996.
10. J. Markoff, “A Deluge of Data Shapes a New Era in Computing” http://www.nytimes.com/2009/12/15/science/15books.html
11. http://www.fourthparadigm.org
12. J. Gray, “eScience—A Transformed Scientific Method”, Talk given to the NRC-CSTB, Mountain View, CA, USA, January 11, 2007. http://research.microsoft.com/en-us/um/people/gray/talks/NRC-CSTB_eScience.ppt

13. “О Стратегии научно-технологического развития Российской Федерации”, Указ Президента Российской Федерации от 01.12.2016 г. № 64

14. Э. Таненбаум, М. ван Стеен: Распределенные системы. Принципы и парадигмы // - СПб., Питер, 2003, с. 876.

15. В.В. Воеводин, Вл.В. Воеводин: Параллельные вычисления // - СПб., БХВ-Петербург, 2002, с. 608.

16. Н.Н. Говорун: Некоторые вопросы применения электронных вычислительных машин в физических исследованиях // Автореферат диссертации на соискание ученой степени доктора физико- математических наук, ОИЯИ, 10-4437, Дубна, 1969.

17. WLCG: Worldwide LHC Computing Grid, http://wlcg.web.cern.ch

18. I. Foster, K. Kesselman: GRID: a Blueprint to the New Computing Infrastructure // Morgan Kaufman Publishers, 1999, p. 690.

19. В.В.Кореньков. “Методология развития научного информационно – вычислительного комплекса в составе глобальной грид-инфраструктуры”. Диссертация на соискания ученой степени доктора технических наук, Дубна, 2012г.

20. LHCOPN : LHC Optical Private Network, http://wlcg.web.cern.ch

21. LHCONE : LHC Open Network Environment, http://lhcone.cern.ch

22. А. Климентов, В. Кореньков: Распределенные вычислительные системы и их роль в открытии новой частицы // Суперкомпьютеры, 2012, №3 (11), стр. 7-11.

23. А. Ваняшин, А. Климентов, В. Кореньков. “За большими данными следит PanDA”, Суперкомпьютеры, 2013, №3 (11), pp. 56-61

24. А.Климентов. “К вопросу о федеративной организации распределенной ЦЕРН”. Журнал "Суперкомпьютеры", 20, 2015. стр 26-28

25. http://toolkit.globus.org/toolkit/about.html

26. В. Бедняков, В. Кореньков: Перспективы Грид-технологий в промышленности и бизнесе // «Знание-сила», 2010, No10, с.97-103

27. V. Ilyin, V. Korenkov, A. Soldatov: RDIG (Russian Data Intensive Grid) e- Infrastructure // Proc. of XXI Int. Symposium of Nuclear Electronics&Computing ((NEC`2007, Varna, Bulgaria), ISBN 5-9530-0171-1, Dubna, 2008, p.233-238.

28. V. Ilyin, V. Korenkov, A. Kryukov, Yu. Ryabov, A. Soldatov: Russian Date intensive Grid (RDIG): current status and perspectives towards national Grid initiative // Proc. of Int. Conf. "Distributed computing and Grid-Technologies in Science and Education, GRID-2008", ISBN 978-5-9530-0198-4, Dubna, 2008, p.100-108.

29. В.Н. Коваленко, Д.А. Корягин: Распределённый компьютинг и грид // книга «Технологии грид», Т.1, - М., ИПМ им. М.В.Келдыша, 2006, с.7-28.

30. A.P. Afanasiev, S.V. Emelyanov, Y.R. Grinberg, V.E. Krivtsov, B.V. Peltsverger, O.V. Sukhoroslov, R.G. Taylor, V.V. Voloshinov: Distributed Computing and Its Applications. // Felicity Press, Bristol, USA, 2005, 298p.

31. А.П. Афанасьев, В.В. Волошинов, С.В. Рогов, О.В. Сухорослов: Развитие концепции распределенных вычислительных сред // Проблемы вычислений в распределенной среде: Сборник трудов ИСА РАН / Под ред. С.В. Емельянова, А.П. Афанасьева, - М., Эдиториал УРСС, 2004.

32. Вл.В. Воеводин, С.А. Жуматий: Вычислительное дело и кластерные системы // - М., Изд-во МГУ, 2007, 150с

33. Вл.В. Воеводин: Top500: числом или уменьем? // Открытые системы, 2005, No10, с.12-15.

34. В.В. Топорков: Модели распределенных вычислений // - М., ФИЗМАТЛИТ, 2004, с.320.

35. V. Korenkov: Grid activities at the Joint Institute for Nuclear Research // Proc. of the 4th Intern. Conf. «Distributed Computing and Grid-Technologies in Science and Education, GRID-2010», ISBN 978-5-9530-0269-1, Dubna, 2010, p.142-147

36. M. Aderholz et al.: Models of Networked Analysis at Regional Centers for LHC Experiments (MONARC) - Phase 2 Report // CERN/LCB, 2000-001 http://monarc.web.cern.ch/MONARC

37. A. Klimentov, M.Pohl "AMS-02 Computing and Ground Data Handling", Computing in High Energy Physics Conference Proceedings, Sep 2004, Interlaken, Switzerland.(2000).

38. S.Campana, A.DiGirolamo, J.Elmsheuser, S.Jezequel, A.Klimentov, J.Schovancova, C.Serfon, G.Stewart, D.van der Ster, I.Ueda and A.Vaniachine, "ATLAS Distributed Computing Operations : Experience and improvements after 2 full years of data-taking", May 2012, 19th International Conference on Computing in High Energy and Nuclear Physics (CHEP12). May 2012.

39. A. Klimentov et al., "Extending ATLAS Computing to Commercial Clouds and Supercomputers", PoS ISGC2014 (2014) 034

40. A. Zarochentsev, A. Kiryanov, A. Klimentov, D. Krasnopevtsev and P. Hristov, "Federated data storage and management infrastructure", Journal of Physics: Conference Series, Volume 762, Number 1

41. Load Sharing Facility. https://www-03.ibm.com/systems/spectrum-computing/products/lsf/index.html

42. Portable Batch System. http://www.pbspro.org/

43. HTCondor. Official site : https://research.cs.wisc.edu/htcondor/

44. S. Bagnaso, L. Betev, P. Buncic et al., "The ALICE Workload Management System: Status before the real data taking", Journal of Physics: Conference Series 219 (2010) 062004

45. S.K. Paterson and A. Tsaregorodtsev, “DIRAC optimized workload management”, Journal of Physics: Conference Series. Volume 119 part 6 (2008)

46. A. Klimentov et al., “Next Generation Workload Management System For Big Data on Heterogeneous Distributed Computing”, J. Phys.Conf. Ser. 608 (2015) no.1, 012040.

47. S. Cittolin et al., “A Remus Based Crate Controller For The Autonomous Processing Of Multichannel Data Streams”. CERN preprint 81-07

48. A. Klimentov et al., “The distributed DAQ system of hadron calorimeter prototype”. Preprint ITEP-18 (1989).

49. А.А.Климентов. Создание комплекса автоматизированных стендов для проведения тестовых испытаний при производстве, сборке и запуске адронного калориметра установки Л3 на ускорителе ЛЕП. Автореферат на соискание степени кандидата физико-математических наук. 01.04.01 / Ин-т теорет. и эксперимент. физики.- Москва, 1991.-РГБ ОД, 9 91-2/3642-7

50. A. Klimentov et al., “Computing Strategy of Alpha-Magnetic Spectrometer Experiment”, NIM (2003) 502

51. A. Klimentov et al., “AMS-02 Computing and Ground Data Handling”, Computing in High Energy Physics Conference Proceedings, Sep 2004, Interlaken, Switzerland.

52. J-P Baud et al., “CASTOR status and evolution”, Computing in High Energy and Nuclear Physics Conference (CHEP 2003), Ла Хойя, Калифорния, США.

53. Mihai Dobre, C. Stratan: Monarc simulation framework // Proceedings of the RoEduNet International Conference, Buletinul Stiintific al Universitatii “Politehnica” din Timisoara, Romania, Seria Automatica si Calculatoare Periodica Politechnica, Transactions on Automatic Control and Computer Science, Vol.49 (63), 2004, ISSN 1224-600X, p.35-42.

54. C. Grogoras, “Monitoring ALICE sites with MonALISA”, рабочее совещание эксперимента ALICE, 20 avgusta 2008, Сибу, Румыния.

55. Европейский проект развертывания грид-систем для научных исследований – EGEE (Enabling Grids for E-science in Europe) - http://www.eu-egee.org

56. Проект по разработке фундаментальных грид-технологий, Альянс Globus - http://www.globus.org/.

57. ROOT : Data analysis framework. https://root.cern.ch

58. D. Costanzo, A. Klimentov et al., “Metadata for ATLAS”, препринт АТЛАС ATL-GEN-PUB-2007-001, ATL-COM-GEN-2007-001.

59. M. Lassnig, “Using machine learning algorithms to forecast network and system load metrics for ATLAS Distributed Computing”. Доклад на конференции Computing in High Energy and Nuclear Physics, Сан-Франциско, США, октябрь 2016 год.

60. M. Titov, G. Zaruba, A. Klimentov, and K. De, “A probabilistic analysis of data popularity in ATLAS data caching,” Journal of Physics: Conference Series, vol. 396, no. 3, 2012.

61. perfSONAR, http://www.perfsonar.net/

62. A. Klimentov, M. Titov “ATLAS Data Transfer Request Package (DaTRI)”, J. Phys.: Conf. Series. Proc. 18th Int. Conf. on Computing in High Energy and Nuclear Physics (CHEP2010)

63. A. Anisenkov, A. Klimentov, R. Kuskov and T. Wenaus, “ATLAS Grid information system”, J.Phys. Conf. Ser. 331 (2011) 072002.

64. M. Pradillo et al., “Consolidating WLCG topology and configuration in the Computing Resource Information Catalogue”. Конференция Computing in High Energy and Nuclear Physics, Сан-Франциско, США, октябрь, 2016

65. D. Oleynik, A. Petrosyan, V. Garonne, S. Campana: On behalf of the ATLAS Collaboration: DDM DQ Deletion Service, Implementation of Central

Deletion Service for ATLAS Experiment // Proceedings of the 5th Intern. Conf. «Distributed Computing and Grid-Technologies in Science and Education, GRID-2012», ISBN 978-5-9530-0345-2, Dubna, 2012, p.189-194

66. Les Robertson. “LHC Data Analysis will start on the Grid. What’s next?”, пленарный доклад на конференции Computing in High Energy and Nuclear Physics (CHEP 2009), Прага, Чехия, март 2009.

67. Аристотель “Метафизика”, 2015, ISBN 978-5-699-83195-1

68. H. S. C. Martin, S. Jha, S. Howorka, and P. V. Coveney. “Determination of Free Energy Profiles for the Translocation of Polynucleotides through α-Hemolysin Nanopores using Non-Equilibrium Molecular Dynamics Simulations”, Journal of Chemical Theory and Computation, August 11, 2009, Volume 5, Issue 8, Pages 1955-2192

69. Top500, ноябрь 2016, https://www.top500.org/lists/2016/11/

70. G. Stewart. “Evolution of computing and software at LHC : from Run2 to HL-LHC”, Конференция Computing in High Energy and Nuclear Physics, апрель 2015, Окинава, Япония.

71. M.Borodin, K.De, J.Garcia Navarro, D.Golubkov, A.Klimentov, T.Maeno and A.Vaniachine, “Scaling up ATLAS production system for the LHC Run 2 and beyond : project ProdSys2”. J.Phys.Conf.Ser. 664 (2015) no.6, 062005.

72. P. J. Laycock, N. Ozturk, M Beckingham, R. Henderson, L Zhou, “Derived Physics Data Production in ATLAS: Experience with Run 1 and Looking Ahead”, Journal of Physics: Conference Series, Volume 513, Track 3

73. http://www.wired.com/magazine/2013/04/bigdata

74. Bernd Panzer-Steindel. “Introduction to CERN Computing”. Летняя лекция ЦЕРН 2015 года

75. Материалы рабочего совещания “BigData Processing and Analysis Challenges”, 29 / 31 января 2015 г, НИЦ “Курчатовский институт”, Москва. https://indico.cern.ch/event/364112/

76. M.A.Grigorieva, M.V.Golosova, M.Y.Gubin, A.A.Klimentov, V.V.Osipova and E.A.Ryabinkin, “Evaluating non-relational storage technology for HEP metadata and meta-data catalog”, Journal of Physics: Conference Series, Volume 762, Number 1.

77. Django software foundation. https://www.djangoproject.com/foundation/

78. ds.j3 https://d3js.or

79. K.De, A.Klimentov, J.Schovancova, T.Wenaus, “The new Generation of the ATLAS PanDA Monitoring System”, PoS ISGC2014 (2014) 035.

80. T.Korchuganova, S.Padolski, T.Wenaus. “ATLAS BigPanDA monitoring and its evolution”. Доклад на 7 международной конференции “Distributed Computing and Grid-technologies in Science and Education”, Дубна, Россия, 2016 г.

81. F. Barreiro, M. Borodin, M. Gubin, D. Golubkov, A. Klimentov, T. Maeno. “Machine Learning Technologies to Predict the ATLAS Production System Behaviour”, Доклад на 7 международной конференции “Distributed Computing and Grid-technologies in Science and Education”, Дубна, Россия, 2016 г.

82. M. Gubin, F. Barreiro, M. Borodin, D. Golubkov, A. Klimentov, T. Maeno, “Machine Learning Technologies to Predict the ATLAS Production System Behaviour” // Proc. of the 2nd International scientific conference “Science of the Future”, 20-23 September 2016, Kazan.

83. I.Foster, Y.Zhao, I.Raicu, S.Lu, “Cloud Computing and Grid Computing 360-Degree Compared”. https://arxiv.org/pdf/0901.0131.pdf

84. M.Sevior. “Belle Monte-Carlo production on the Amazon EC2 cloud”, международная конференция ISGC, Тайпей, Тайвань, апрель 2009 год.

85. Google Compute Engine Portal https://cloud.google.com/products/compute-engine

86. HTCondor Project http://research.cs.wisc.edu/htcondor

87. CVMFS Portal http://cernvm.cern.ch/portal/filesystem

88. Пакет программ для управления виртуальными машинами. CERNVM https://cernvm.cern.ch

89. Amazon EC2. http://aws.amazon.com/ec2/pricing

90. ТОП500 суперкомпьютеров https://en.wikipedia.org/wiki/TOP500

91. Международная конференция Supercomputers2016, http://sc16.supercomputing.org , Солт-Лейк Сити, США.

92. Geant4. http://geant4.cern.ch

93. SAGA-Python (Simple API for Grid Applications) , http://saga-project.github.io/saga-python/

94. P. Calafiura et al., “Running ATLAS workloads within massively parallel distributed applications using Athena Multi-Process framework (AthenaMP)”. Computing in High Energy and Nuclear Physics, апрель 2015 год, Окинаша, Япония.

95. Проект Gaudi. http://gaudi.web.cern.ch/gaudi/

96. AliROOT. ALICE Offline project. https://alice-offline.web.cern.ch

97. M.Al-Turani et al., “ALFA: The new ALICE-FAIR software framework”. J. Phys.: Conf. Ser. 664 (2015) 072001.

98. В.А. Аулов, А.А. Климентов, Р.Ю. Машинистов, А.В. Недолужко, А.М. Новиков, А.А. Пойда, И.С. Тертычный, А.Б. Теслюк, Ф.С. Шарко Интеграция гетерогенных вычислительных инфраструктур для анализа данных геномного секвенирования. Математическая биология и биоинформатика, Том 11, выпуск 2, 2016 год, С. 205-213. doi: 10.17537/2016.11.205.

99. Skryabin K.G., Prokhortchouk E.B., Mazur A.M., Boulygina E.S., Tsygankova S.V., Nedoluzhko A.V., Rastorguev S.M., Matveev V.B., Chekanov N.N., Goranskaya D.A., Teslyuk A.B., Gruzdeva N.M., Velikhov V.E., Zaridze D.G., Kovalchuk M.V. “Combining two technologies for full genome sequencing of human”. *Acta Nat*. 2009. V. 1. № 3. P. 102–107.

100. Schubert M., Ermini L., Sarkissian C.D., Jonsson H., Ginolhac A., Schaefer R., Martin M.D., Fernandez R., Kircher M., McCue M., Willerslev E., Orlando L. Characterization of ancient and modern genomes by SNP detection and phylogenomic and metagenomic analysis using PALEOMIX. *Nat Protoc*. 2014. V. 9. P. 1056-1082.

101. ATLAS collaboration. Search for a Charged Higgs Boson Produced in the Vector-Boson Fusion Mode with Decay H(±)→W(±)Z using pp Collisions at √s=8 TeV with the ATLAS Experiment. Phys Rev Lett. 2015 Jun 12;114(23):231801. Epub 2015 Jun 9

102. L.Mascetti et al. Disk storage at CERN. J.Phys.Conf.Ser. 664 (2015) 042035 (2015-12-23)

103. dCache, https://www.dcache.org

104. DPM, http://lcgdm.web.cern.ch/dpm

105. EOS, https://eos.web.cern.ch

106. Xrootd, http://xrootd.org

107. А.Климентов “Distributed Computing Beyond The Grid”, пленарный доклад на международной конференции Grid2012, Дубна, Россия.

108. Российский консорциум РДИГ (Российский грид для интенсивных операций с данными – Russian Data Intensive Grid, RDIG) - http://www.egee-rdig.ru

109. V. Ilyin, V. Korenkov, A. Soldatov: RDIG (Russian Data Intensive Grid) e-Infrastructure: status and plans. Proc. of XXII Int. Symposium on Nuclear Electronics & Computing (NEC`2009, Varna, Bulgaria), ISBN 978-5-9530- 0242-4, Dubna, 2010, p.150-153.

110. DynaFed, https://svnweb.cern.ch/trac/lcgdm/wiki/Dynafeds

111. A.Zarochentsev, A.Kiryanov, A.Klimentov, D.Krasnopevtsev and P.Hristov. Federated data storage and management infrastructure. Journal of Physics: Conference Series, Volume 762, Number 1

112. Bonnie++, http://www.coker.com.au/bonnie++/

113. FUSE, http://fuse.sourceforge.net/

114. A.Klimentov et al Integrating Network Awareness in ATLAS Distributed Computing using ANSE project. Доклад на конференции Computing in High Energy and Nuclear Physics (CHEP2015), Окинава, Япония, апрель, 2015 год.

115. X.Espinal et al. Di-EOS. Running EOS across two computing centres. https://indico.cern.ch/event/214784/session/9/contribution/96/attachments/340864/475686/Poster-distributed-EOS.pdf