ОБЪЕДИНЕННЫЙ ИНСТИТУТ ЯДЕРНЫХ ИССЛЕДОВАНИЙ

ЛАБОРАТОРИЯ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

*На правах рукописи*

*(подпись соискателя ученой степени)*

**Олейник Данила Анатольевич**

# Методика и программное обеспечение для подключения суперкомпьютеров к распределенной системе обработки данных эксперимента ATLAS

Специальность 05.13.11 – Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

АВТОРЕФЕРАТ

диссертации на соискание ученой степени

кандидата технических наук

Дубна – 2021

Работа выполнена в Лаборатории информационных технологий Объединенного института ядерных исследований.

**Научный руководитель:** Кореньков Владимир Васильевич, доктор технических наук

**Официальные оппоненты:**

доктор технических наук Дегтярев Александр Борисович

кандидат физико-математических наук Антонов Александр Сергеевич

С электронной версией диссертации можно ознакомиться на официальном сайте Объединенного института ядерных исследований в информационно-телекоммуникационной сети «Интернет» по адресу: https://dissertations.jinr.ru. С печатной версией диссертации можно ознакомиться в Научно-технической библиотеке ОИЯИ.

Ученый секретарь диссертационного совета ОИЯИ.05.01.2019.П, д.ф.-м.н. Земляная Елена Валериевна

**Общая характеристика работы**

Работа основана на результатах исследований и разработках, проведенных соискателем в рамках проекта BigPanDA. В диссертации представлена методика подключения высокопроизводительных вычислительных центров (суперкомпьютеров) к распределенной высокопропускной системе обработки данных эксперимента ATLAS. Показана реализация данной методики как в виде прототипа, так и полнофункционального программного комплекса. На основе проведенных исследований сформулированы рекомендации по организации высокопропускной обработки данных на инфраструктурах, ориентированных на высокопроизводительные вычисления, а также продемонстрирована возможность повышения эффективности использования суперкомпьютеров.

***Актуальность темы исследования***

Распределенные вычислительные системы являются базовой инфраструктурой для обработки данных современных крупных научно-исследовательских проектов. Толчком в развитии и применении таких систем стали эксперименты на Большом адронном коллайдере (БАК). ATLAS - самый крупный из экспериментов на LHC, представляет собой не только уникальную физическую установку, но и коллаборацию ученых и специалистов из многих стран мира. Эксперимент производит петабайты данных, для обработки и анализа которых была выбрана высокопропускная парадигма. Высокопропускной вычислительной системой называется объединение вычислительных ресурсов (узлов) на длительное время для одновременного выполнения большого количества независимых вычислительных задач. Производительность такой системы измеряется не количеством одновременно выполняемых элементарных вычислительных операций, а количеством задач, выполненных за единицу времени. Для обеспечения обработки данных экспериментов LHC было решено построить общую, географически распределенную, высокопропускную вычислительную инфраструктуру на ресурсах участников коллабораций. Данная инфраструктура была реализована в начале 2000-ых и является на данный момент, пожалуй, наиболее известной системой типа grid - Worldwide LHC Computing Grid (WLCG).

Система позволяет выполнять более миллиона задач в день, задействуя одновременно сотни тысяч процессоров. Рост системы ограничен бюджетными возможностями участников, и в основном происходит за счет технологического развития вычислительных систем. Это не всегда отвечает возрастающим

потребностям коллаборации. Ускорительный комплекс и экспериментальные установки постоянно совершенствуются. ЦЕРН, с поддержкой европейской комиссии по физике частиц, осуществляет многолетний план развития, предусматривающий 4 крупных модернизации, сопровождающиеся полной остановкой детекторов и ускорителя. Основной целью модернизации является проведение новых исследований, а результатом – увеличение объема получаемых и производимых данных. Это неотвратимо ведет к увеличению количества задач, необходимых для обработки и анализа экспериментальных данных.

Эксперимент ATLAS уже после первой плановой модернизации начал рассматривать возможность использования ресурсов, предоставляемых сторонними поставщиками и не подключенных в инфраструктуру WLCG. Такие ресурсы включают в себя: коммерческие и некоммерческие облачные инфраструктуры, вычислительные ресурсы, предоставляемые на волонтерской основе, высокопроизводительные вычислительные комплексы – суперкомпьютеры. Подобную систему, включающую разнородные вычислительные ресурсы, можно назвать гетерогенной вычислительной системой.

С 2012 года, в рамках национальных программ, некоторые научные группы эксперимента ATLAS, начали получать доступ к высокопроизводительным вычислительным системам. Объем таких выделенных ресурсов, как правило, значительный и слишком велик для задач индивидуального анализа. В таком случае вычислительные комплексы имеет смысл подключать к системе автоматизированной обработки данных эксперимента, обеспечивающий постоянный поток вычислительных задач.

До 60% вычислительных ресурсов в ATLAS используется для моделирования физических событий методом Монте-Карло. Это высокоавтоматизированный процесс, обеспечивающий долговременный постоянный поток задач, не зависящий от периода набора данных или пользовательской активности. Задачи моделирования ресурсоемкие и могут не требовать доступа к внешним ресурсам по глобальной сети в процессе выполнения, что делает их хорошими кандидатами для выполнения на вычислительных узлах суперкомпьютеров.

Несмотря на определенную схожесть, высокопроизводительные вычислительные центры имеют ряд очень существенных отличий от вычислительных центров WLCG, начиная с организации доступа к ресурсу и заканчивая архитектурой и политикой использования, ориентированными на выполнение больших параллельных задач.

Определенную заинтересованность в совместных исследованиях демонстрировали и представители высокопроизводительных вычислительных центров. Загрузка крупных суперкомпьютеров на данный момент составляет 85–90% от теоретически достижимой. Это обусловлено в большей степени политикой предоставления и использования ресурсов такой системы. Десять процентов загрузки современного суперкомпьютера – это сотни миллионов ЦПУ*часов в год. Такие ресурсы, как правило, распределены неравномерно во времени и имеют относительно короткий период доступности.

Технология, позволяющая повысить загрузку без существенного увеличения числа задач в системе, используя эти свободные ресурсы, вызывает большой интерес со стороны высокопроизводительных вычислительных центров. Некоторые крупные центры готовы были предоставлять такие ресурсы на безвозмездной (внеконкурсной) основе.

С 2013 года коллаборация ATLAS, в рамках развития собственной распределенной системы обработки данных, инициировала ряд научно-исследовательских проектов для интеграции в нее суперкомпьютеров.

***Цели и задачи***

Целью исследования является разработка методики и программного обеспечения для интеграции суперкомпьютеров с распределенной высокопропускной системой обработки данных эксперимента ATLAS.

Основными задачами, которые нужно было решить в рамках исследования, являлись:

- Исследование особенностей организации высокопроизводительных вычислительных центров
- Разработка методики интеграции суперкомпьютеров с распределенной высокопропускной системой обработки данных
- Создание на основе уже разработанных компонент прототипа сервиса, реализующего разработанную методику и анализ функционирования созданного прототипа
- Исследование особенностей поведения прикладных приложений, используемых в высокопропускной обработке данных, на суперкомпьютерах
- Разработка методик оптимизации поведения упомянутых прикладных приложений на суперкомпьютерах

- Исследование возможности увеличения эффективности использования высокопроизводительных систем (суперкомпьютеров)
- Практическая применимости проведенных исследований и разработок

***Научная новизна***

Предложена методика непосредственного подключения суперкомпьютеров в распределенную систему обработки данных, минуя промежуточное программное обеспечение грид.

Впервые для эксперимента ATLAS реализована методика подключения суперкомпьютеров к распределенной вычислительной системе под управлением PanDA WMS для высокопропускной обработки данных. Разработанная методика была впервые применена для подключения широкого спектра высокопроизводительных вычислительных систем, включая: суперкомпьютеры Titan (OLCF), Theta (ALCF), Cori (NERSC).

Предложена и реализована методика одновременного выполнения на суперкомпьютерах набора независимых задач обработки данных, использующих одинаковую среду выполнения, под контролем внешней системы управления нагрузкой.

Предложен и реализован алгоритм, позволяющий увеличить эффективность использования высокопроизводительных вычислительных систем, ориентированных на выполнение больших параллельных задач (использующих до нескольких тысяч вычислительных узлов), посредством динамической дозагрузки суперкомпьютера малыми задачами под контролем внешней системы управления нагрузки.

***Научно-практическая значимость работы***

В рамках диссертационной работы проведены исследования вычислительных систем, реализующих различные концепции осуществления большого объема вычислений. Предложена и реализована методика построения распределенных гетерогенных вычислительных инфраструктур. С использованием предложенной методики в распределенную систему обработки данных эксперимента ATLAS на БАК были интегрированы ряд высокопроизводительных вычислительных систем, включая ресурсы Ок-Риджской национальной лаборатории (суперкомпьютер Titan). Существенным практическим результатом стала эволюция архитектуры

системы управления нагрузкой PanDA, расширенной специализированным сервисом Harvester, ориентированным на эффективное взаимодействие с различного рода вычислительными инфраструктурами.

Предложенная методика и ее программная реализация позволила обеспечить практически беспрерывное выполнение задач обработки данных эксперимента ATLAS на суперкомпьютере Titan с 2016 по 2019 год.

Реализованный в программной среде PanDA алгоритм использования свободных, нераспределенных между другими задачами, ресурсов позволил увеличить загрузку суперкомпьютера Titan.

Разработанное программное обеспечение для интеграции с высокопроизводительными вычислительными системами использовалась, помимо эксперимента ATLAS, и для обеспечения расчетов и обработки данных и других научных групп. Благодаря возможности использования высокопроизводительных вычислительных систем, PanDA WMS была выбрана в качестве решения, обеспечивающего распределенную обработку данных в эксперименте COMPASS. Предложенные методы и программные решения были использованы специалистами COMPASS для доработки пилотного приложения и модулей Harvester с учетом специфики вычислительных задач этого эксперимента.

***Методология и методы исследования***

В исследованиях для данной диссертационной работы были использованы методы системного анализа, анализа программного обеспечения и программной инженерии, технологии программирования и организации взаимодействия программ в глобально распределенной среде.

***Положения, выносимые на защиту***

1. Методика и программное обеспечение, интегрирующее высокопроизводительные вычислительные ресурсы в систему распределенной обработки данных эксперимента ATLAS.
2. Методика выполнения набора независимых задач, генерируемых внешней системой управления нагрузкой, и ее реализация с использованием технологии MPI.

3. Алгоритм и программное обеспечение, позволяющее увеличить эффективность использования высокопроизводительной вычислительной системы, реализованные для суперкомпьютера Titan.

***Апробация и степень достоверности***

Ход и результаты исследований регулярно докладывались на целом ряде крупных международных конференций, в частности:

- Доклады на конференции CHEP (International Conference on Computing in High Energy and Nuclear Physics) с 2015 по 2019 год.
- Презентация работы на конференции Supercomputing с 2016 по 2019 год.
- Регулярные доклады на рабочих совещаниях ATLAS Distributed Computing с 2013 по 2019 год.
- Доклад на конференции PASC (The Platform for Advanced Scientific Computing) 2018.

Достоверность исследования подтверждается практически проверенной корректностью получаемых вычислительных результатов, работоспособностью и эффективностью использования подключаемых вычислительных инфраструктур, в том числе успешно проведенной полной интеграцией методики в распределенную систему обработки данных эксперимента ATLAS и подключением суперкомпьютерных центров OLCF, ALCF, NERSC, CSCS, Marenostrum к системе распределенной обработки данных ATLAS.

Разработанное программное обеспечение включено в состав стандартных компонент программного инструментария системы управления нагрузкой PanDA WMS.

***Публикации и личный вклад***

По теме диссертации опубликовано 19 научных работ, 12 из которых опубликованы в рецензируемых изданиях, соответствующих требованиям к публикациям Положения о присуждении ученых степеней в ОИЯИ (пр. ОИЯИ от 30.04.2019 № 320).

Предложенные методики интеграции суперкомпьютеров в высокопропускную распределенную систему обработки данных, оптимизации выполнения прикладных приложений, не разработанных для выполнения на суперкомпьютерах, и увеличения эффективности использования суперкомпьютера

были сформулированы при определяющем участии автора. Прототип интегрирующего программного комплекса, разработанного на основе указанных методик, реализован лично автором диссертации.

Соискатель принимал участие в создании и развитии, обеспечивающего взаимодействие с вычислительными инфраструктурами, сервиса Harvester, разработав компоненты для интеграции с ресурсами OLCF (Titan, Summit) и в разработке компонент Pilot 2.0 для работы на вычислительных узлах суперкомпьютеров.

***Соответствие диссертации паспорту специальности***

В диссертационной работе присутствуют результаты в трех областях, соответствующих следующим пунктам паспорта специальности:

3. Модели, методы, алгоритмы, языки и программные инструменты для организации взаимодействия программ и программных систем

8. Модели и методы создания программ и программных систем для параллельной и распределенной обработки данных, языки и инструментальные средства параллельного программирования

9. Модели, методы, алгоритмы и программная инфраструктура для организации глобально распределенной обработки данных

***Объем и структура диссертации***

Диссертационная работа состоит из введения, трех глав, заключения, списка публикаций, списка цитируемой литературы (85 пунктов). Работа содержит 95 страниц и включает в себя 27 рисунков и 4 таблицы.

**Содержание работы**

***Введение***

Во введении к данной работе показаны актуальность исследования, описаны основные цели, практическая значимость и научная новизна. Перечислены положения, выносимые на защиту, приведены сведения об апробации результатов и описан личный вклад автора.

***Первая глава***

В первой главе описывается проект BigPanDA: проблемная область, цели и задачи проекта, ставшие основой исследования. Описывается система управления нагрузкой PanDA WMS.

Проект BigPanDA стартовал в 2013 году, с основной целью – развитие системы управления нагрузкой PanDA WMS. Основными задачами проекта BigPanDA были:

- Исследование возможности использования системы управления нагрузкой PanDA WMS для других научных дисциплин, не только физики высоких энергий и широкого круга научных групп;
- Интеграция PanDA WMS с высокопроизводительными вычислительными системами;
- Исследование возможности увеличения эффективности использования суперкомпьютеров;

В проекте участвовали: Брукхейвенская Национальная Лаборатория, Университет Техаса в Арлингтоне, Ок-Риджская национальная лаборатория, университет Ратгерс. Проект был успешно реализован и закончился через 6 лет, в 2019 году.

Система управления нагрузкой PanDA WMS – одна из ключевых компонент системы обработки данных в распределенной вычислительной среде, используемой в коллаборации ATLAS. Разработка системы началась в 2005 году, изначально для объединения вычислительных систем, расположенных в университетах и национальных лабораториях США, а в 2009 году была принята коллаборацией ATLAS как единая система, объединяющая все вычислительные ресурсы участников. PanDA является мультиагентной системой, обеспечивающей полный цикл обработки заданного объема данных путем формирования и контроля выполнения необходимого числа задач, выполняемых на вычислительных узлах при помощи специализированных приложений – пилотов.

PanDA Pilot - исполнительная среда для задач PanDA. Пилот запрашивает описание задачи из центрального сервиса, подготавливает входные данные и окружение для выполнения, запускает задачу и контролирует ход исполнения с периодическим оповещением сервиса, осуществляет выгрузку выходных данных для долговременного хранения. Пилотная исполнительная среда обеспечивает концепцию “позднего связывания”, что позволяет эффективно управлять

приоритетами выполнения задач и игнорировать время ожидания в очереди локального планировщика.

На момент начала работ в рамках проекта BigPanDA, система управления нагрузкой PanDA обеспечивала единовременное выполнение более 100 тысяч задач, при этом в течение дня выполнялось более одного миллиона задач, использовавших более ста пятидесяти тысяч процессоров.

Особенностью системы обработки данных для эксперимента ATLAS является возможность управления "размером" задачи по времени выполнения, а также относительная равномерность времени выполнения задач в рамках обработки гомогенного набора данных. Данное свойство заинтересовало представителей OLCF (Ок-Риджскокого передового вычислительного центра) с точки зрения возможности динамического формирования размеров вычислительных задач, что, в теории, должно было повысить эффективность использования вычислительной инфраструктуры.

Для проведения совместных поисковых работ OLCF предоставил ресурсы и экспертизу в области высокопроизводительных систем, разработчики PanDA экспертизу в области организации распределенной обработки данных и функционирующую систему, обеспечивающую такую обработку.

Крупные высокопроизводительные вычислительные системы имеют целый ряд отличий, как архитектурных, так и организационных, от входящих в грид-инфраструктуру вычислительных центров, реализующих высокопропускную концепцию.

Ниже приведены основные особенности высокопроизводительного вычислительного комплекса OLCF, верные и для других суперкомпьютерных центров.

- Вычислительная система коллективного использования
- Конкуренция за получение ресурсов начиная с формулировки запроса на предоставления выделенного объема ресурсов
- Политика использования, ориентированная на выполнение больших вычислительных задач с использованием параллельных систем
- Ограниченный доступ к ресурсам, собственная система аутентификации и авторизации. Невозможность использования сторонних методов доступа
- Специализированная конфигурация вычислительных узлов. Ограниченный набор системного ПО.

- Общая файловая система как средство промежуточного хранения данных задач
- Специализированная файловая система для размещения прикладного ПО
- Отсутствие внешнего сетевого соединения с вычислительными узлами
- Ограниченное (небольшое) количество слотов в очереди локального планировщика для одного пользователя: такая политика исключает практическую возможность сделать большое количество запросов ресурсов малого размера, более того, есть обратная зависимость на период выделения ресурсов в зависимости от размера запроса (малый объем выдается на короткий срок)

Основными техническими проблемами, которые было необходимо решить для интеграции PanDA с OLCF, являлись:

- двухфакторная система авторизации, исключающая возможность автоматизировать удаленное подключение к системе
- доставка задач на вычислительные узлы суперкомпьютера, загрузка входных данных и выгрузка выходных данных в отсутствии обычных для грид-систем интерфейсов доступа к вычислительным системам – Computing Element (CE) и интерфейсов систем хранения – Storage Element (SE)
- выполнение большого количества задач в условиях крайне ограниченного количества слотов в локальном планировщике
- минимизация времени ожидания в очереди локального планировщика

Задача стояла шире, чем подключение единичного суперкомпьютера: нужно было решить задачу подключения нового типа вычислительных ресурсов и их эффективного использования. Для достижения этой цели требовалось предложить общую методику подключения и разработать адаптируемое к особенностям каждого ресурса промежуточное ПО.

***Вторая глава***

Во второй главе диссертации описывается предложенная методика интеграции, разработанный прототип интегрирующего сервиса, выявленные особенности организации работы на высокопроизводительных вычислительных системах.

Особенностью высокопроизводительных вычислительных систем является необходимость высокоскоростного обмена данными между вычислительными узлами и процессорами в процессе выполнения задачи. Это диктует как

особенности архитектуры суперкомпьютеров, так и предъявляет определенные требования к прикладному программному обеспечению.

Зачастую, по тем или иным причинам, невозможно адаптировать прикладное программное обеспечение, использующееся в высокопропускной обработке данных, для эффективной работы на суперкомпьютерах, даже несмотря на достаточную программно-аппаратную совместимость счетных узлов, используемых при реализации и тех и других систем.

Методика интеграции систем включает в себя следующие действия:

- организация связи с внешними сервисами для доставки описаний задач, загрузки входных и выгрузки выходных данных;
- осуществление выполнения задач посредством взаимодействия с локальным планировщиком;
- реализация возможности выполнения достаточного для высокопропускной обработки количества задач в условиях ограниченного количества слотов в очереди локального планировщика;
- выбор оптимального размера слотов или размера слотов в заданном диапазоне при возможности и необходимости их динамического формирования;
- разработка рекомендации по размещению и организации прикладного программного обеспечения для эффективного исполнения на высокопроизводительной вычислительной системе.

Связь с внешними сервисами на высокопроизводительных системах возможна только с интерактивных узлов, предназначенных для доступа пользователей к вычислительным ресурсам. Запуск счетных задач осуществляется на вычислительных узлах посредством планировщика и локального менеджера ресурсов. Как правило, не подразумевается какой-либо дополнительный обмен информацией между интерактивными и вычислительными ресурсами. Отличительной чертой является наличие общей файловой системы, объединяющей все типы узлов.

Пилотная среда выполнения задач PanDA предполагает наличие сетевой связи между пилотами и центральным сервером, поэтому использование PanDA Pilot на вычислительных узлах не имеет практического смысла. Взаимодействие пилотного приложения и задачи осуществляется через общее дисковое пространство и отслеживание ассоциированных процессов на уровне

операционной системы. Таким образом, для осуществления выполнения задач PanDA на суперкомпьютерах функциональность пилотного приложения должна быть разделена между интерактивными узлами и вычислительными узлами.

Реализация данной функциональности возможна с помощью системы приложений, состоящей из сервиса или набора сервисов на интерактивном узле, обеспечивающих связь с внешними компонентами распределенной системы обработки данных, локальной системой управления ресурсами и распределенной файловой системой суперкомпьютера (Рисунок 1).

Еще одной компонентой в этой программной системе является приложение, обеспечивающее выполнение задач на вычислительных узлах. В функции данного приложения входят: обеспечение связи с интерфейсной компонентой на интерактивном узле; настройка среды выполнения задачи, необходимые действия для обеспечения оптимизации работы задачи в условиях высокопроизводительного вычислительного комплекса.

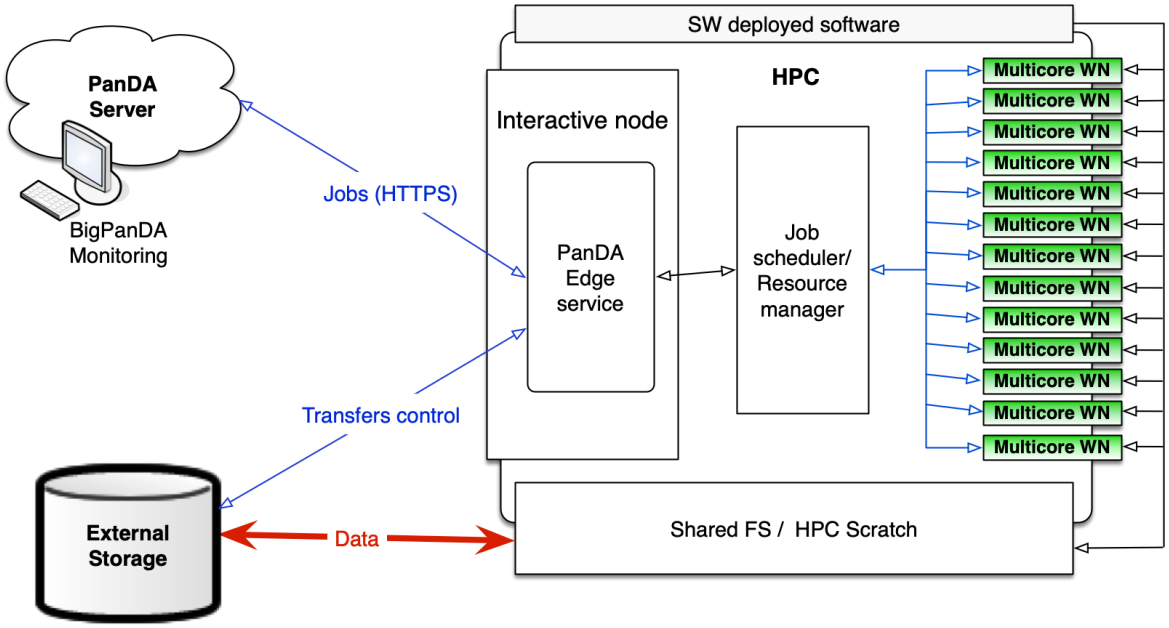


Рисунок 1. Концептуальная схема подключения высокопроизводительной вычислительной системы к распределенной высокопропускной системе обработки данных

В условиях наличия существенных ограничений на количество слотов (запросов на ресурсы у локального планировщика) и, одновременно, отсутствия лимита на объем ресурсов в рамках запроса, необходим механизм, позволяющий выполнять одновременно существенное количество "небольших" задач.

Для реализации этого механизма была выбрана технология MPI как наиболее распространенная на параллельных вычислительных системах. Инструментированное MPI приложение, выполняющееся на вычислительных узлах, позволяет запускать в рамках одного нумерованного процесса параллельного приложения одну задачу PanDA.

Использование пилотной среды выполнения в распределенных системах позволяет практически игнорировать время ожидания в очереди локального планировщика ресурсов. В условиях ограничений высокопроизводительного вычислительного комплекса и непосредственной работы с локальным планировщиком необходимо, по возможности, минимизировать это время ожидания. При "традиционной" организации работы на суперкомпьютере приоритет запроса ресурсов, а, соответственно, и время ожидания в очереди, существенно зависит от запрашиваемого объема ресурсов: у “больших” слотов приоритет выше. Можно сказать, что побочным эффектом такой политики является “освобождение” большого количества ресурсов на короткое время. Эвристическим методом было установлено, что запросы, наиболее подходящие под текущие свободные ресурсы по объему и, прежде всего, по времени доступности, имеют, как правило, минимальное время ожидания в очереди.

Система управления нагрузкой PanDA имеет возможность генерировать задачи с заданным в некоторых пределах временем выполнения на вычислительном ресурсе, что позволяет использовать вычислительные ресурсы, доступные даже на непродолжительное время.

Локальная система управления ресурсами в OLCF оснащена средствами мониторинга, предоставляющими информацию о доступных ресурсах с периодом их доступности. На основе этой информации можно сформировать соответствующий запрос ресурсов для слота. С высокой степенью вероятности запрошенные ресурсы выделяются планировщиком в течение непродолжительного (до двух минут) ожидания в очереди. В рамках работы проводились исследования о влиянии приоритета запросов на время пребывания в очереди локального планировщика. Было практически показано, что при использовании такой тактики, приоритет заявки становится ничтожным при принятии решения планировщиком о предоставлении ресурсов.

На основе вышеописанных возможностей был предложен следующий алгоритм формирования слота:

- запрашивается информация о доступных на текущий момент времени ресурсах;
- в соответствии с полученной информацией формируется заявка на ресурсы и передается планировщику;
- если в течение непродолжительного времени ресурсы не предоставляются, то запрос отменяется и процесс повторяется.

***MultiJob Pilot прототип интегрирующего сервиса***

Предложенная методика и алгоритмы были программно реализованы в виде специализированного интегрирующего сервиса. Ввиду того, что значительный объем функциональности: интерфейсы с серверной компонентой PanDA, загрузка/выгрузка данных, контроль выполнения задач и некоторые другие уже была реализована в PanDA Pilot, именно это приложение было взято за основу прототипа сервиса. Была пересмотрена изначальная идеология пилотного приложения: один пилот - одна задача, на один пилот - один слот. Это повлекло за собой изменение требований: сопровождение выполнения набора задач в рамках одного слота, что, соответственно, ведет к модификации процедур загрузки/выгрузки входных/выходных данных, форматов обмена информацией между процессами приложения, изменение процедур мониторинга. При этом организация рабочего процесса пилотного приложения со строгой последовательностью основных процедур: получение задач, загрузка входных данных, подготовка среды, запуск задач, контроль выполнения, постобработки и первичного анализа выполнения задачи, выгрузка результатов, осталась неизменной (Рисунок 2).

Исполнение задачи посредством PanDA Pilot осуществляется путем запуска дочернего процесса и контролируется через отслеживание данного процесса на уровне ОС и на основе отслеживания изменений в рабочей директории задачи. В разработанном прототипе эта функциональная часть была полностью изменена: на основе библиотек RADICAL-SAGA был разработан интерфейс взаимодействия с локальным планировщиком. Данный интерфейс позволял запускать и отслеживать выполнение задания на уровне локального планировщика ресурсов. В качестве задания выступает специально разработанное MPI приложение обеспечивающее выполнение набора задач PanDA.

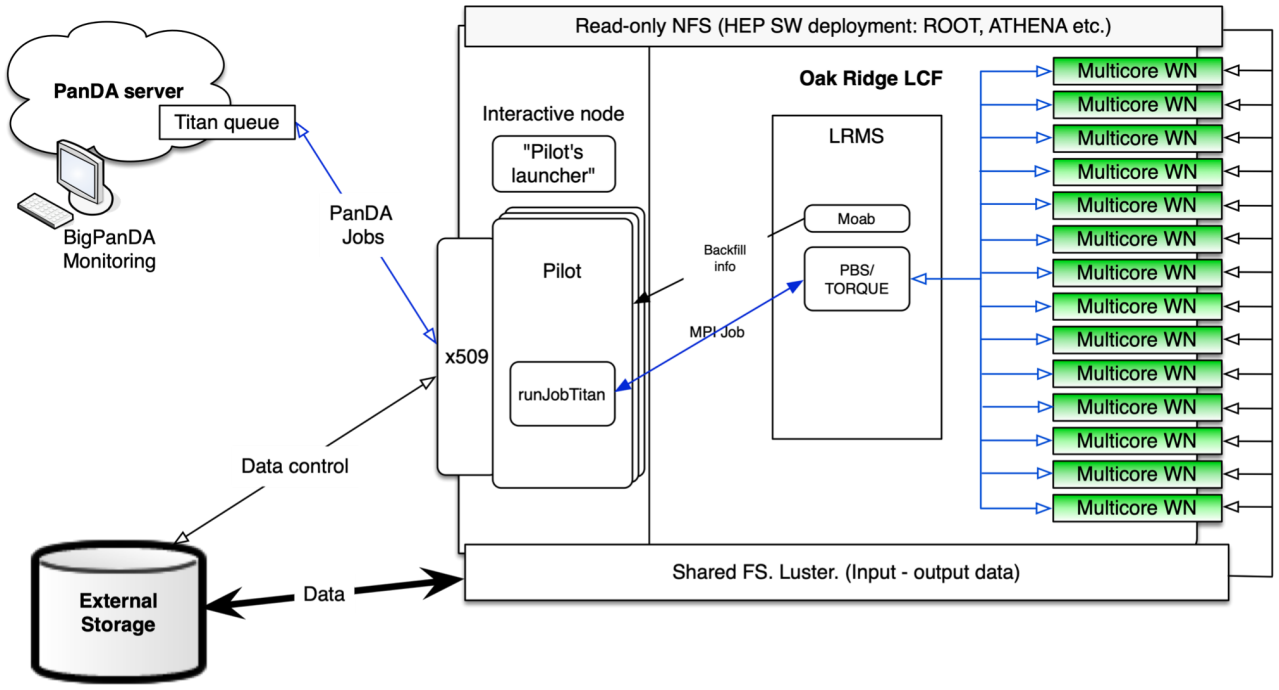


Рисунок 2. Схема прототипа интегрирующего сервиса MultiJob Pilot

Прототип сервиса был дополнен процедурами, позволяющими получать информацию о доступных ресурсах. На основе этой информации формировался запрос необходимого количества задач от серверной компоненты PanDA. Поскольку перед запуском задачи необходим целый ряд подготовительных процедур, занимающий некоторое время, в течение которого объем доступных ресурсов может измениться, то перед отсылкой запроса локальному планировщику осуществляется контрольный сбор информации о доступных ресурсах и, при необходимости размер запроса корректируется.

В случае, если в течение непродолжительного времени (2 - 3 минут) запрошенные ресурсы не выделяются планировщиком и запрос все еще находится в стадии ожидания в очереди, заявка отменяется и приложение заканчивает свою работу.

Для осуществления обработки потока задач необходимо, чтобы пилотные приложения регулярно запускались и работали. В рамках распределенной инфраструктуры рассылкой пилотов занимается специализированный сервис, однако ввиду особенностей доступа к высокопроизводительным вычислительным центрам использование этого сервиса не представлялось возможным. Для решения этой проблемы было разработано постоянно работающее фоновое приложение, осуществляющее запуск пилотов и контроль числа работающих

пилотов для эффективного использования доступных слотов локального планировщика.

На начальном этапе программная система, разработанная в рамках создания прототипа, позволяла обслуживать четыре слота в очереди локального планировщика. Максимальное количество доступных слотов является ограничением исходящим из политики использования ресурсов. Размер каждого слота варьировался от 15 до 450 вычислительных узлов, выполняя по одной задаче PanDA на вычислительный узел (16 процессоров). В дальнейшем, по согласованию с экспертами OLCF, максимальное количество слотов доступных в рамках проекта было увеличено до двадцати, но максимальный размер слота пришлось уменьшить до 300 узлов ввиду определенных архитектурных недостатков прототипа.

Разработанный прототип был интегрирован с системой обработки данных ATLAS, что позволило получить практически постоянный поток задач (несколько тысяч в день) и проводить продолжительные и нагрузочные испытания (Рисунок 3).

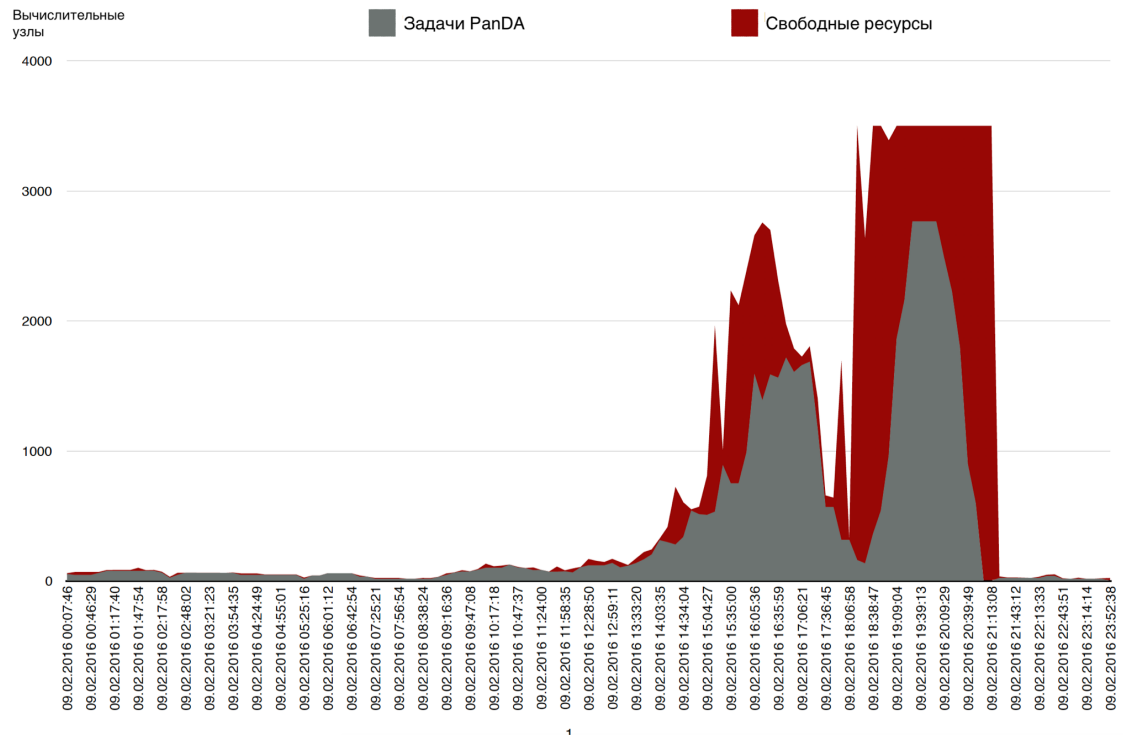


Рисунок 3. Пример захвата свободных ресурсов при помощи прототипа интегрирующего сервиса

***Выявленные особенности, недостатки и методы устранения***

Разработанный прототип показал принципиальную возможность интеграции высокпоропускной системы обработки данных с высокопроизводительными вычислительными системами. Однако был выявлен и ряд недостатков, ограничивающих масштабируемость решения. Пожалуй, самым существенным недостатком было наследие архитектуры PanDA Pilot с достаточно жесткой последовательностью: получение задачи, загрузка входных данных, подготовку к исполнению и исполнение задачи, выгрузка результатов и выходных данных (Рисунок 4). С увеличением количества выполняемых задач увеличивалось и время необходимое на их подготовку, во время которого ресурсы не только не используются, но могут быть и “упущены”, в то же время увеличивались пиковые нагрузки на внешние сервисы: серверную компоненту PanDA, удаленную систему хранения.

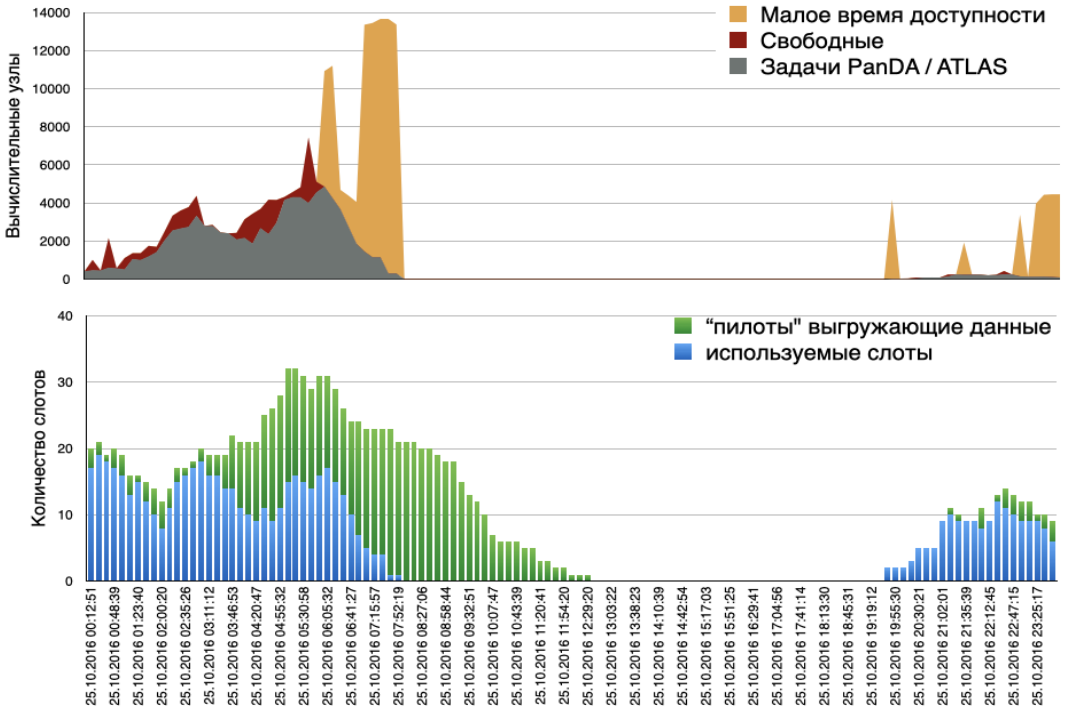


Рисунок 4. “Упущенные” ресурсы в результате не оптимальной архитектуры прототипа

Был проведен ряд оптимизаций: в интерфейс серверной компоненты PanDA были добавлены групповые операции для получения и обновления одним запросом набора задач PanDA, оптимизация загрузки входных данных - один входной файл для группы задач. Тем не менее во время пиковых нагрузок возникал неприемлемый объем сбоев, выше 10% от выполненных задач.

В то же время возникла необходимость существенного увеличения масштабируемости (производительности) разрабатываемого программного комплекса для обеспечения возможности одновременного выполнения существенно большего количества задач PanDA.

Дальнейшая разработка на основе программной реализации PanDA Pilot не представлялась возможной ввиду ее принципиальных недостатков. За десятилетия развития этого приложения качество кода сильно упало, что вызывало существенные сложности с его последующей поддержкой. Для радикального избавления от архитектурных проблем было принято решение разработать абсолютно концептуально новый сервис, отвечающий за взаимодействие с вычислительной инфраструктурой. Такой сервис был реализован и получил название Harvester.

Harvester – многопотоковое масштабируемое приложение, включающее в себя набор слабосвязанных подпрограмм, выполняющихся одновременно в различных процессах и использующее базу данных для временного хранения и обмена информацией о выполняемых задачах и их статусах. Каждая подпрограмма выполняет определенную функциональную часть и изменяя статус задачи в локальной БД. При необходимости увеличения производительности для определенной функциональности можно увеличить количество процессов с определенными подпрограммами (Рисунок 5).

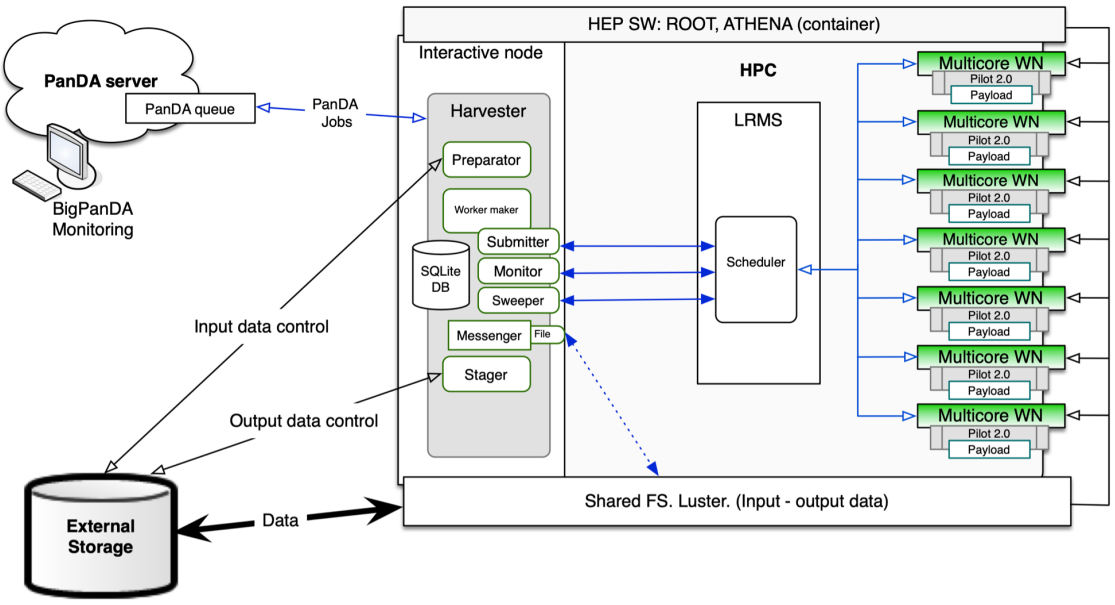


Рисунок 5. Схема сервиса Harvester при работе с суперкомпьютером

С применением Harvester величину слота можно было увеличить до 3800 узлов (задач PanDA), что практически решило проблему масштабируемости для суперкомпьютера Titan.

При увеличении нагрузок возникали проблемы не только с масштабируемостью прототипа, но и с нагрузками, приходящихся на подсистемы суперкомпьютера.

Для выполнения задач ATLAS была очевидна необходимость использования прикладного программного обеспечения ATLAS. Ввиду специфики архитектуры суперкомпьютера Titan, не было возможности обеспечить доступ к сервису дистрибуции прикладного программного обеспечения в WLCG – CVMFS. Прикладное программное обеспечение ATLAS было размещено и сконфигурировано для работы с сетевой файловой системой Spider II (Lustre). Данная файловая система общего назначения, объемом 32 петабайта и пропускной способностью 1 терабайт в секунду, была основной системой промежуточного хранения данных для задач пользователей.

При увеличении количества выполняемых задач PanDA в рамках слота и увеличение количества используемых слотов, стала заметна возрастающая нагрузка на файловую систему, связанная с архитектурой прикладного программного обеспечения ATLAS, и особенностями организации выполнения задач. Поскольку, по ряду причин, архитектура прикладного программного не могла быть адаптирована для высокопроизводительных систем, необходимо было провести ряд исследований по возможности оптимизации использования ресурсов OLCF. Первым шагом, еще до проведения специальных исследований, был перенос прикладного программного обеспечения ATLAS с файловой системы общего назначения на сетевую файловую систему NFS, ориентированную на чтение. На какое-то время это решило проблему, но, тем не менее нагрузки, оказываемые на файловую систему, оставались существенным ограничивающим фактором. Уязвимой частью файловой системы являлись севера метаданных, хранящих информацию о структуре директорий и физическом размещении файлов. Необходимо было проанализировать поведение прикладного программного обеспечения и выявить основные источники большого количества запросов метаданных: чтение директорий, открытия/закрытия файлов. Это потребовало более детального изучения особенностей архитектуры прикладного программного обеспечения, использующегося в ATLAS начиная с программной среды. Программная среда прикладных приложений ATLAS, осуществляющих

обработку и анализ данных, называется ATHENA. Для создания ATHENA использовались два языка программирования: C++ для разработки основного кода алгоритмов, и Python используемый для конфигурирования модулей и формирования последовательности выполнения алгоритмов. AthenaMP это развитие начального фреймворка с поддержкой многопроцессорности.

Анализ журнальных записей типовой задачи полной симуляции, реализованной на ATHENA MP и использующей один вычислительный узел Titan (16 ядер ЦПУ), показал, что в процессе исполнения происходит открытие более 6800 файлов, включая около 1900 подключаемых библиотек и более 1400 интерпретируемых файлов Python. Суммарное количество обращений к файловой системе на основе данных ОС приведено в таблице 1. Учитывая, что в рамках одного слота выполняется от десятков до сотен задач, суммарный фактор воздействия получается крайне высоким.

| | **Open ok** | **Open fail** | **Stat ok** | **Stat fail** |
|---|---|---|---|---|
| **Основной фреймворк (NFS)** | **109987** | **371688** | **124522** | **318206** |
| **Spider / Lustre** | 2902 | 5860 | 2502 | 2356 |
| **Рабочая директория задачи (Lustre)** | 471 | 2605 | 365 | 995 |

Таблица 1. Количество обращений к файловым системам в процессе выполнения одной задачи ATHENA MP

В результате проведенного анализа были выявлены два фактора, наиболее воздействующие на файловую систему:

- Поведение интерпретатора Python при поиске внешних модулей (библиотек). Интерпретатор выполняет "сканирование" текущей рабочей директории и затем по списку директорий указанных в ряде переменных окружения в процессе поиска необходимого модуля
- Не оптимизированное поведение прикладного программного обеспечения для работы на распределенной файловой системе: каждая задача в процессе выполнения создает множество временных файлов и директорий

Проведенная первичная оптимизация позволила снизить количество избыточных вывозов к файловой системе в 4 раза, но общее число все еще было слишком высоким, для удовлетворения требованиям совместной работы на высокопроизводительной вычислительной системе.

Перенос рабочей директории задачи в RAM диск на вычислительном узле должен был существенно сократить количество обращений к распределенной файловой системе в процессе выполнения задачи. До внесения изменений в процесс запуска задачи необходимо было оценить время, необходимое для копирования рабочей директории и входных данных. Были проведены нагрузочные тесты: копирование файла ~4Gb на каждый рабочий узел для слота на: 1,5,150,400 и 800 узлов. Результаты проведенных тестов показали очень незначительное время, требующееся на подготовку (в среднем до 40 секунд для слота на 800 узлов), что являлось очень незначительным увеличением относительно времени выполнения задачи. Данная функциональность была реализована.

Путем последовательных шагов оптимизации было достигнуто ~30 кратное уменьшение количества запросов к общей файловой системе для слота из 350 узлов. Это позволило без изменения архитектуры прикладного ПО достичь желаемой эффективности в достаточно сжатые сроки.

Резюмируя содержание второй главы, можно отметить, что, реализуя предложенную методику интеграции путем развития промежуточного ПО, в частности компонент системы управления нагрузкой удалось решить поставленную задачу и интегрировать высокопроизводительный вычислительный комплекс: суперкомпьютер Titan (OLCF) в распределенную высокопропускную систему обработки данных эксперимента ATLAS.

***Третья глава***

Третья глава посвящена достигнутым практическим результатам работы. Условно, результаты можно разделить на три группы:

- успешная интеграция суперкомпьютера Titan в систему распределенной обработки данных эксперимента ATLAS и применение методики и разработанных компонент промежуточного программного обеспечения для подключения как других высокопроизводительных систем, так и для использования в обеспечении обработки данных других научных групп;
- увеличение эффективности использования суперкомпьютеров на примере Titan, набор рекомендаций по организации высокопропускной обработки данных программным обеспечением неориентированным для исполнения на высокопроизводительных вычислительных системах;

- вклад в развитие распределенных вычислительных систем и практическая реализация гетерогенной вычислительной системы, объединяющей ресурсы реализующие различные парадигмы;

Отдельно можно вынести набор методических рекомендаций по организации высокопропускной обработки данных с использованием прикладного программного обеспечения, не оптимизированного для работы на высокопроизводительных вычислительных системах.

Очевидным результатом работы стала методика и разработанные средства, позволяющие использовать высокопроизводительные вычислительные центры, как элементы распределенной системы обработки данных. Результаты проводимых исследований вызвали необходимость пересмотра концептуальной архитектуры системы управления нагрузкой для работы в распределенной гетерогенной вычислительной среде. Система была расширена специализированным, ориентированным на вычислительные инфраструктуры, сервисом Harvester, который скрывает архитектурные особенности вычислительных систем от сервиса распределения задач, в то же время обеспечивая их эффективное использование (Рисунок 6).

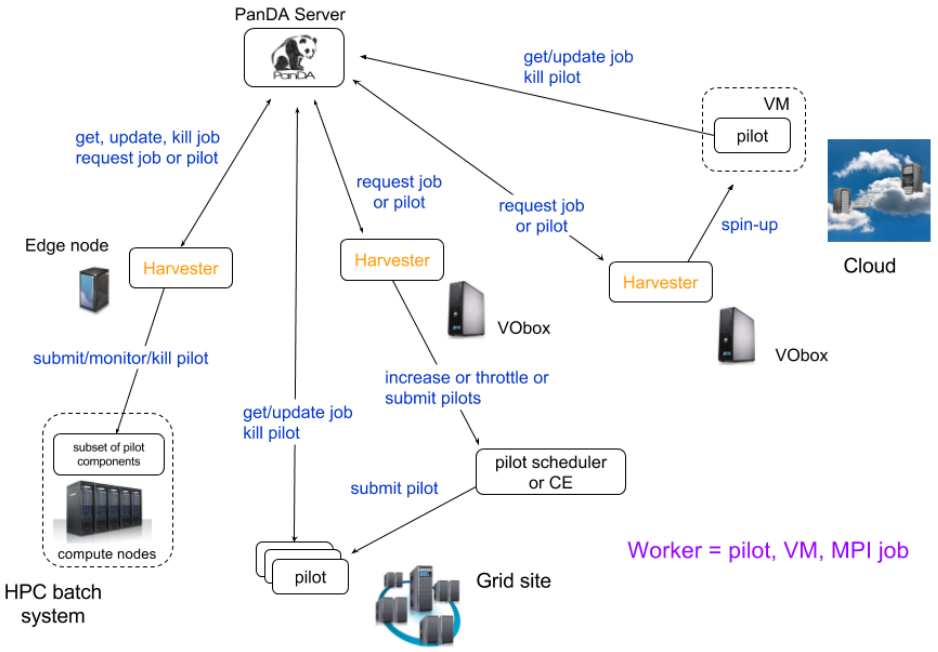


Рисунок 6. Инфраструктурно-ориентированный сервис Harvester в системе управления нагрузкой

Детальный анализ архитектуры и имплементации PanDA Pilot, осуществленный в процессе разработки прототипа интегрирующего сервиса MultiJob Pilot, стал

мотивацией к полной переработке пилотного приложения. Архитектура PanDA Pilot была заложена на этапе начального проектирования системы управления нагрузкой, и не пересматривалась в течение следующих лет эксплуатации. Несмотря на неизменность архитектуры, приложение расширялось, чтобы покрывать, возникающие в процессе развития системы, требования. Сопровождение выполнения задач на вычислительных узлах суперкомпьютеров и на ресурсах облачных вычислительных систем потребовала полного пересмотра архитектуры этого приложения. Основным требованием к новой архитектуре стала максимально возможная адаптивность приложения не только широкому набору промежуточных сервисов распределенной среды, но и возможность имплементации специфичных процессов выполнения задач. Разработка новой версии пилотного приложения PanDA Pilot 2.0 заняла более двух лет и еще около полугода понадобилось на полномасштабное внедрение в эксплуатацию.

Практическую пользу для коллаборации ATLAS данная работа начала приносить, как только прототип интегрирующего сервиса MultiJob Pilot был подготовлен для устойчивой работы. С 2015 года практически до вывода суперкомпьютера Titan из эксплуатации в 2019 году, задачи ATLAS постоянно выполнялись в OLCF. Ввиду специфики вычислительной системы, было принято решение, что только задачи определенного типа будут на ней выполняться, а именно: полное моделирование физического события методом Монте-Карло - как наиболее ресурсоемкий тип задач. Осуществляя выполнение сотен тысяч вычислительных задач в месяц, интегрирующий сервис обеспечивал моделирование сотен миллионов событий. До осени 2017 года задачи ATLAS выполнялись только в режиме использования ресурсов, не распределенных между другими пользователями Titan. Тем не менее практически в каждый момент времени выполнялась полезная работа для ATLAS, используя от нескольких десятков до нескольких сотен вычислительных узлов.

В 2017 году, в дополнение к уже имеющимся условиям работы в OLCF, ATLAS получил первый грант в размере 80 миллионов ЦПУ/часов на суперкомпьютере Titan. Для использования этого объема ресурсов было необходимо изменить тактику работы, чтобы они больше удовлетворяли принятым на высокопроизводительных системах условиям. Требовалось запрашивать больший объем ресурсов, чтобы воспользоваться преимуществом повышения первоначального приоритета и по возможности уменьшить время ожидания в очереди (Таблица 2).

| Приоритетные группы | Минимум узлов в запросе | Максимум узлов в запросе | Максимальное время в запросе (Часы) | Повышение приоритета (Дни) |
|---|---|---|---|---|
| 1 | 11250 | 18688 | 24 | +15 |
| 2 | 3750 | 11249 | 24 | +5 |
| 3 | 313 | 3,749 | 12 | 0 |
| 4 | 126 | 312 | 6 | 0 |
| 5 | 1 | 125 | 2 | 0 |

Таблица 2. Организация работы очереди заданий на суперкомпьютере Titan

Разработанное программное обеспечение позволяет реализовать и подобную тактику использования ресурсов, обеспечивая при этом выполнение до нескольких тысяч задач PanDA, одновременно используя несколько десятков тысяч ЦПУ (Рисунок 7).

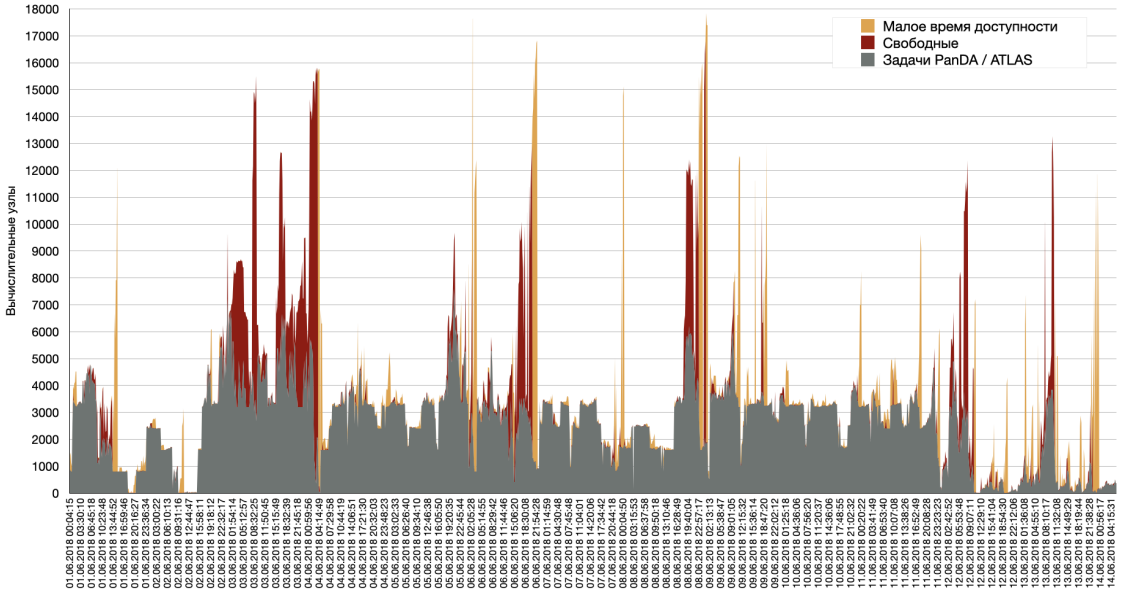


Рисунок 7. Использование суперкомпьютера Titan для задач PanDA/ATLAS с использованием сервиса Harvester

За период с мая 2015 года по июнь 2019 на суперкомпьютере Titan было выполнено более 18 миллионов задач ATLAS, позволивших осуществить моделирование более чем миллиарда физических событий.

Предложенная методика и разработанное ПО было использовано для интеграции и других высокопроизводительных вычислительных систем, например, ALCF

(суперкомпьютер Theta), NERSC (суперкомпьютер Cori), ресурсы TACC и многих других в систему обработки данных эксперимента ATLAS. Ориентировочная оценка показала, что до 10-15% от всех объемов моделирования физических процессов для эксперимента ATLAS стали выполняться на ресурсах высокопроизводительных вычислительных центров в рамках выделенных грантов или национальных программ.

Отдельно стоит отметить успешное использование данной методики и компонент разработанного ПО, специалистами коллаборации COMPASS, при подключении суперкомпьютеров Blue Waters (NCSA) и Frontera (TACC) в систему обработки данных эксперимента. Ввиду определенной специфики архитектуры прикладного ПО, прогнозируемой доработке подверглось пилотное приложение обеспечивающее выполнение задач на вычислительных узлах. Применение рекомендации, полученных в рамках исследований в OLCF позволило добиться и достаточной производительности на Frontera.

Применение техники работы на свободных, не распределенных между другими пользователями, узлах суперкомпьютера, предположительно должно увеличить эффективность его использования. Данный эффект был продемонстрирован в ходе представляемой работы. Предустановленный минимально возможный приоритет для запросов ресурсов и их минимальное время ожидания, практически исключали вариант конкуренции за эти ресурсы с другими пользователями. По измерениям с января 2016 года по сентябрь 2018 в таком режиме, при помощи разработанного программного комплекса на Titan было использовано 370 миллионов ЦПУ*часов Это соответствует 2,8% от всего объема вычислительной мощности Titan в этот период(Рисунок 8).

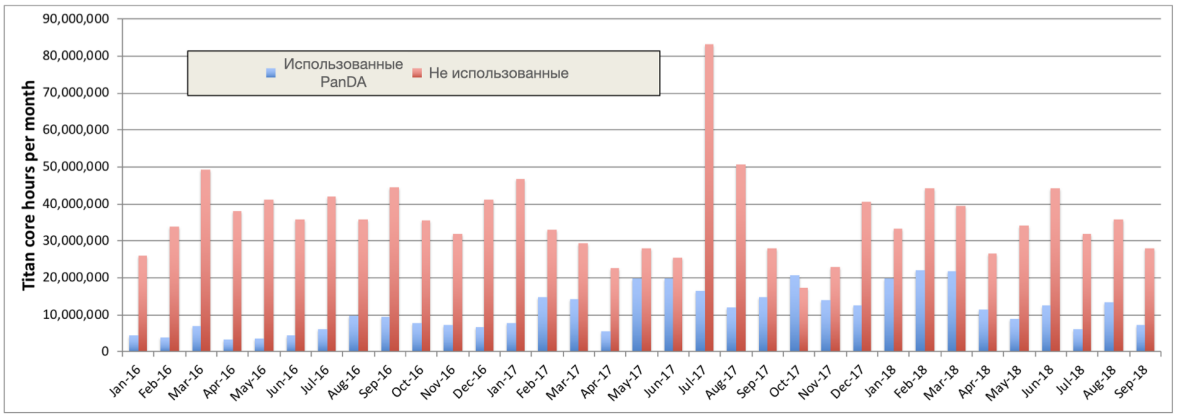


Рисунок 8. Потребление свободных ресурсов суперкомпьютера Titan при помощи PanDA

Рассматривая исторический тренд за несколько лет использования суперкомпьютера Titan, можно оценить увеличение утилизации суперкомпьютера на ~2%, что подтверждает повышение эффективности использования этого высокопроизводительного вычислительного комплекса (Рисунок 9). Проведенный анализ возможного увеличения времени ожидания ресурсов для задач других пользователей не показал существенного воздействия в результате использования предлагаемой методики.


Рисунок 9. Увеличение эффективности использования суперкомпьютера Titan при помощи PanDA

По результатам исследований, направленных на оптимизацию работы высокопропускных прикладных приложений на высокопроизводительных вычислительных системах при помощи промежуточного программного обеспечения, были сформулированы следующие рекомендации:

- Система управления нагрузкой должна быть способна выдерживать пиковые нагрузки, зависящие от объемов выделяемых ресурсов. Для крупных высокопроизводительных вычислительных комплексов это могут быть десятки тысяч ЦПУ;
- Желательно, чтобы на суперкомпьютерах выполнялись наиболее ресурсоемкие или даже специализированные типы задач;
- Уязвимым местом высокопроизводительной вычислительной системы является система ввода/вывода. Общая файловая система между узлами суперкомпьютера требует контролируемых нагрузок, чтобы минимизировать негативное воздействие на задачи других пользователей;

- При наличии технической возможности, код прикладного ПО нужно размещать на специально выделенной файловой системе, как правило, ориентированной на чтение;
- Для хранения промежуточных данных, возникающих в ходе выполнения задачи, рекомендуется использовать ресурсы вычислительного узла, а не общей файловой системы.

Несмотря на специфичность архитектуры и политики использования суперкомпьютеров, показана возможность их интеграции в распределенные вычислительные системы, построенные в рамках высокопропускной парадигмы. Интеграция осуществляется путем развития промежуточного программного обеспечения, управляющего процессом обработки больших объемов данных. Необходимо принимать во внимание архитектурные особенности каждого суперкомпьютера и, зачастую, дорабатывать отдельные компоненты для эффективной возможности использования его ресурсов.

***Заключение***

Можно выделить следующие основные результаты представленной диссертационной работы:

1. Предложена и реализована в программном виде, как расширение системы управления нагрузкой PanDA WMS, методика интеграции высокопроизводительных вычислительных ресурсов в глобально распределенную высокопропускную систему обработки данных.
2. Реализована методика выполнения не связанных высокопропускных приложений на суперкомпьютерах. В рамках методики выработаны рекомендации для оптимизации выполнения таких приложений в высокопроизводительной вычислительной среде.
3. Реализован оригинальный алгоритм для использования свободных, нераспределенных между другими запросами, ресурсов суперкомпьютера, позволяющий повысить эффективность его использования.
4. Разработанное программное обеспечение позволило осуществить интеграцию ресурсов Ок-Риджского передового вычислительного центра (OLCF), в частности суперкомпьютера Titan, и ряда других высокопроизводительных вычислительных ресурсов в глобально распределенную систему обработки данных эксперимента ATLAS.

5. Продемонстрирована возможность использования методики и разработанного ПО для широкого круга научных дисциплин:
    - LSST/DESC - построение прототипа распределенной среды обработки астрономических данных с комбинацией грид систем и суперкомпьютеров
    - nEDM - использование суперкомпьютера Titan для задач моделирования электрического дипольного момента нейтрона
    - IceQube - использование суперкомпьютеров Titan/Summit для обработки данных с нейтронного телескопа
    - Использование свободных ресурсов суперкомпьютера Titan для задач генетики
    - Построение прототипа распределенной вычислительной системе на высокопроизводительных вычислительных ресурсах национальных лабораторий США для обеспечения расчетов квантовой хромодинамики на решетках
6. Проведенные исследования привели к эволюции компьютерной модели обработки данных крупных экспериментов, расширив экосистему используемых вычислительных ресурсов

**Список публикаций по теме диссертации**


1. Barreiro Megino F., De K., Jha S., Klimentov A., Maeno T., Nilsson P., Oleynik D., Padolski S., Panitkin S., Wells J., Wenaus T., «Integration of Titan supercomputer at OLCF with ATLAS Production System», Journal of Physics: Conference Series, 898, 9, 92002, 10.1088/1742-6596/898/9/092002, 17426588, (2017)
2. Oeynik D., Panitkin S., Turilli M., Angius A., Oral S., De K., Klimentov A., Wells J.C., Jha S., «High-throughput computing on high-performance platforms: A case study», Proceedings - 13th IEEE International Conference on eScience, eScience 2017, 8109148, 295 -304, 10.1109/eScience.2017.43, 9781538626863, (2017)
3. P. Nilsson, A. Anisenkov, D. Benjamin, D. Drizhuk, W. Guan, M. Lassnig, D. Oleynik, P. Svirin, T. Wegner, "The next generation PanDA Pilot for and beyond the ATLAS experiment", EPJ Web of Conferences, 214, 03054, DOI:10.1051/epjconf/201921403054, (2019)
4. Pavlo Svirin, Kaushik De, Alessandra Forti, Alexei Klimentov, Rasmus Larsen, Peter Love, Tadashi Maeno, Ruslan Mashinistov, Swagato Mukherjee, Andrei

Nomerotski, Danila Oleynik, Sergey Panitkin, Hye Yun Park, Erin Sheldon, Anze Slosar, Jack Wells, Torre Wenaus, "BigPanDA: PanDA Workload Management System and its Applications beyond ATLAS", EPJ Web of Conferences, 214, 03050, DOI:10.1051/epjconf/201921403050, (2019)

5. Tadashi Maeno, Fernando Harald Barreiro Megino, Doug Benjamin, David Cameron, John Taylor Childers, Kaushik De, Alessandro De Salvo, Andrej Filipcic, John Hover, FaHui Lin, Danila Oleynik, "Harvester: an edge service harvesting heterogeneous resources for ATLAS", EPJ Web of Conferences, 214, 03030, DOI: 10.1051/epjconf/201921403030, (2019)
6. Anisenkov A., Drizhuk D., Guan W., Lassnig M., Nilsson P., Oleynik D., "Global heterogeneous resource harvesting: The next-generation PanDA Pilot for ATLAS", Journal of Physics: Conference Series, 1085, 3, 32031, 10.1088/1742-6596/1085/3/032031, 17426588, (2018)
7. Barreiro F., Oleynik D., Benjamin D., Childers T., De K., Elmsheuser J., Filipcic A., Klimentov A., Lassnig M., Maeno T., , Panitkin S., Wenaus T., The Future of Distributed Computing Systems in ATLAS: Boldly Venturing Beyond Grids // EPJ Web of Conferences, 214, 03030, DOI: 10.1051/epjconf/201921403047, (2019)
8. Megino F.H.B., De K., Klimentov A., Maeno T., Nilsson P., Oleynik D., Padolski S., Panitkin S., Wenaus T., «PanDA for ATLAS distributed computing in the next decade», Journal of Physics: Conference Series, 898, 5, 52002, 10.1088/1742-6596/898/5/052002, 17426588, (2017)
9. Megino F.B., Bejar J.C., De K., Hover J., Klimentov A., Maeno T., Nilsson P., Oleynik D., Padolski S., Panitkin S., Petrosyan A., Wenaus T., «PanDA: Exascale Federation of Resources for the ATLAS Experiment at the LHC», EPJ Web of Conferences, 108, 1001, 10.1051/epjconf/201610801001, 21016275, 9782759819447, (2016)
10. De K., Klimentov A., Maeno T., Mashinistov R., Nilsson P., Oleynik D., Panitkin S., Ryabinkin E., Wenaus T., «Accelerating Science Impact through Big Data Workflow Management and Supercomputing», EPJ Web of Conferences, 108, 1003, 10.1051/epjconf/201610801003, 21016275, 9782759819447, (2016)
11. Klimentov A., Buncic P., De K., Jha S., Maeno T., Mount R., Nilsson P., Oleynik D., Panitkin S., Petrosyan A., Porter R.J., Read K.F., Vaniachine A., Wells J.C., Wenaus T., «Next generation workload management system for big data on heterogeneous distributed computing», Journal of Physics: Conference Series, 608, 1, 12040, 10.1088/1742-6596/608/1/012040, 17426588, (2015)

12. Calafiura P., De K., Guan W., Maeno T., Nilsson P., Oleynik D., Panitkin S., Tsulaia V., Van Gemmeren P., Wenaus T., «Fine grained event processing on HPCs with the ATLAS Yoda system», Journal of Physics: Conference Series, 664, 9, 92025, 10.1088/1742-6596/664/9/092025, 17426588, (2015)

13. De K., Klimentov A., Maeno T., Nilsson P., Oleynik D., Panitkin S., Petrosyan A., Schovancova J., Vaniachine A., Wenaus T., «The future of PanDA in ATLAS distributed computing», Journal of Physics: Conference Series, 664, 6, 62035, 10.1088/1742-6596/664/6/062035, 17426588, (2015)

14. Calafiura P., De K., Guan W., Maeno T., Nilsson P., Oleynik D., Panitkin S., Tsulaia V., Van Gemmeren P., Wenaus T., «The ATLAS event service: A new approach to event processing», Journal of Physics: Conference Series, 664, 6, 62065, 10.1088/1742-6596/664/6/062065, 17426588, (2015)

15. Nilsson P., De K., Filipcic A., Klimentov A., Maeno T., Oleynik D., Panitkin S., Wenaus T., Wu W., «Extending ATLAS computing to commercial clouds and supercomputers», Proceedings of Science, 23-28-March-2014, 34, 18248039, (2014)

16. Maeno T., De K., Klimentov A., Nilsson P., Oleynik D., Panitkin S., Petrosyan A., Schovancova J., Vaniachine A., Wenaus T., Yu D., «Evolution of the ATLAS PanDA workload management system for exascale computational science», Journal of Physics: Conference Series, 513, TRACK 3, 32062, 10.1088/1742-6596/513/3/032062, 17426588, (2014)

17. Baginyan A.S., Balandin A.I., Dolbilov A.G., Golunov A.O., Gromova N.I., Kadochnikov I.S., Kashunin I.A., Korenkov V.V., Mitsyn V.V., Oleynik D.A., Pelevanyuk I.S., Petrosyan A.S. "Grid at JINR", CEUR Workshop Proceedings, 2507, c. 321-325, (2019)

18. Chudoba J., Elias M., Fiala L., aHorky J., Kouba T., Kundrat J., Lokajicek M., Schovancova J., Svec J., Belov S., Dmitrienko P., Dolbilov A., Korenkov V., Mitsyn V., Oleynik D., Petrosyan A., Rusakovich N., Tikhonenko E., Trofimov V., Uzhinsky A., Zhiltsov V. "JINR (Dubna) and Prague Tier2 sites: Common experience in the WLCG grid infrastructure", Physics of Particles and Nuclei Letters, 10(3), c. 288-294, (2013)

19. Belov S., Kadochnikov I., Korenkov V., Kutouski M., Oleynik D., Petrosyan A., "VM-based infrastructure for simulating different cluster and storage solutions used on ATLAS Tier-3 sites", Journal of Physics: Conference Series, 396(PART 4), 042036, (2012)