

# Комплексная система управления данными и задачами в гетерогенной компьютерной среде

А.А Климентов Р.Ю Машинистов А.М Новиков А.А Пойда И.С Тертычный

Национальный Исследовательский Центр “Курчатовский Институт”,  
Москва

[alexei.klimentov@cern.ch](mailto:alexei.klimentov@cern.ch) [Ruslan.Mashinistov@cern.ch](mailto:Ruslan.Mashinistov@cern.ch) [novikov@wdcbr.ru](mailto:novikov@wdcbr.ru)  
[Poyda\\_AA@nrcki.ru](mailto:Poyda_AA@nrcki.ru) [itertychnyy@gmail.com](mailto:itertychnyy@gmail.com)

## Аннотация

В статье представлен подход к организации вычислений, основанный на опыте PanDA, и разработанная система управления заданиями и данными, позволяющие объединять вычислительные мощности различных инфраструктур, таких как Грид, облачные инфраструктуры и суперкомпьютеры, для проведения масштабных вычислений в областях науки, требующих высокоинтенсивных и высокопроизводительных вычислений.

*Данная работа выполнена в рамках мегагранта правительства РФ, контракт No 14.Z50.31.0024. Мы благодарны нашим коллегам из НИЦ “Курчатовский институт” и ЦЕРН за обсуждение результатов работы, особая благодарность группе разработчиков WMS PanDA и группам МИФИ и биоинформатики НИЦ “Курчатовский институт”.*

## Введение

Возрастающие объемы научных расчетов предъявляют все более высокие требования к используемым вычислительным мощностям. В качестве таковых продолжают использоваться грид-инфраструктуры, платформы облачных вычислений, суперкомпьютеры, университетские и исследовательские кластеры и т.д. Однако все чаще появляются задачи, для которых необходима интеграция и федерализация вычислительных ресурсов. Одной из таких задач стала задача по обработке, анализу и моделированию данных для экспериментов на Большом Адронном Коллайдере (БАК).

Например, для обработки данных эксперимента ATLAS во время первого рабочего запуска БАК были использованы сотни вычислительных центров и хранилищ мульти-петабайтного масштаба (общий объем данных составил 160ПБайт) по всему миру. Использование специальной системы управления потоком заданий (Workload Management System) PanDA позволило объединить гетерогенные вычислительные мощности (в основном входящие в инфраструктуру Грид - WLCG [1]) и предоставить ученым унифицированный доступ к компьютерному ресурсу.

В статье представлен подход к организации вычислений, основанный на опыте PanDA, и разработанная система управления заданиями и данными, позволяющие объединять вычислительные мощности различных инфраструктур, таких как Грид, облачные инфраструктуры и суперкомпьютеры, для проведения масштабных вычислений в областях науки, требующих высокоинтенсивных и высокопроизводительных вычислений.

## 1 Описание разработанного подхода и архитектуры реализующей его программной системы

### 1.1 Система управления потоком заданий PanDA

В качестве основы для разрабатываемой технологии и реализующей его программной системы был выбран подход, используемый в системе управления заданиями PanDA (“Production and Distributed Analysis”) [2], с 2007 года успешно используемой для обработки, анализа и моделирования данных эксперимента ATLAS, проводимого на БАК [3].

Программный комплекс PanDA обеспечивает “прозрачность” обработки данных в распределенной вычислительной инфраструктуре. Он предоставляет среду выполнения для широкого диапазона экспериментальных приложений, автоматизирует централизованную обработку данных, обеспечивает анализ данных для десятков групп физиков, поддерживает пользовательский поток операций,

---

Труды XVII Международной конференции DAMDID/RCDL’2015 «Аналитика и управление данными в областях с интенсивным использованием данных», Обнинск, 13-16 октября 2015

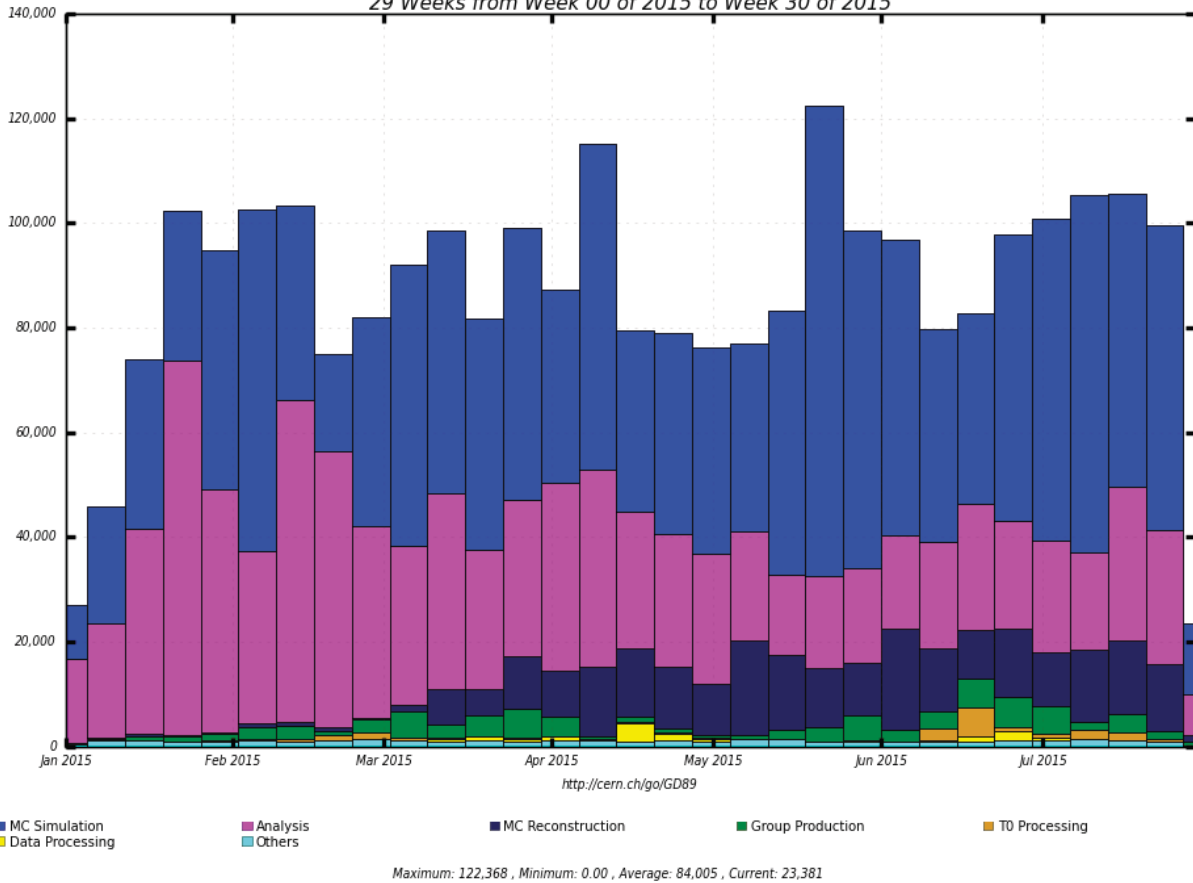


Рис. 1. Число выполняющихся заданий ATLAS на ресурсах WLCG превосходило 100000 в 2015 г.

обеспечивает единый доступ к распределенным глобальным ресурсам, предоставляет состояние и историю выполняемых операций через интегрированную систему контроля и управляет распределением данных.

Масштабируемость PanDA была продемонстрирована в процессе работы коллаборации ATLAS при быстром росте числа выполняемых заданий во время первого рабочего запуска БАК (2010-2013 гг.). На рисунке 1 представлена статистика по количеству запущенных задач ATLAS на ресурсах WLCG с начала 2015 года. Программный комплекс PanDA был разработан достаточно гибким для адаптации новых технологий обработки и хранения данных и сетевых технологий.

Мы воспользовались опытом разработчиков PanDA и постарались обобщить их подходы с целью разработки подхода и системы, способных объединить различные вычислительные инфраструктуры.

### 1.2 Архитектура разработанной системы

Схема разработанного подхода и общая архитектура реализующей его системы представлены на рисунке 2.

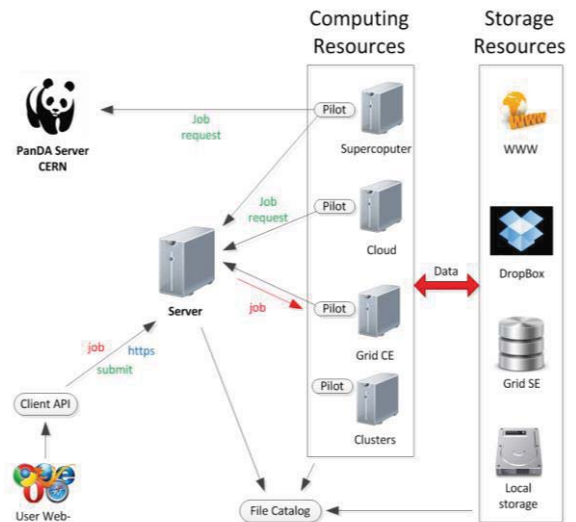


Рис. 2. Архитектура разработанной системы

Основными компонентами системы являются:

- Сервер. Сервер принимает от пользователей задачи и управляет их выполнением: сортирует задачи по разным очередям в зависимости от требований, распределяет их на доступные вычислительные ресурсы (при этом сервер отслеживает, чтобы объемы и состав ресурсов удовлетворял требованиям задачи), отслеживает состояние задачи и ее жизненный цикл,

осуществляет дополнительные проверки успешности выполнения задачи, поддерживает метаинформацию о всех активных и завершенных задачах и т.п.

- **Пилотные задания.** Пилотные задания осуществляют механизм поздней привязки задачи к среде вычислений. Они запускаются на ресурсах, предназначенных для выполнения пользовательских задач, проверяют их состояние, резервируют, собирают информацию и передают на сервер, который в ответ присылает задачу. Пилотное задание инициирует запуск присланной задачи непосредственно на ресурсах и контролирует процесс ее выполнения.

Система пилотных задач позволяет не только осуществлять позднюю привязку, но и скрыть неоднородность различных вычислительных сред с помощью их унифицированного интерфейса “сервер-пилот”. Таким образом, можно интегрировать в единую вычислительную среду различные вычислительные инфраструктуры, например, Грид-инфраструктуру, инфраструктуру облачных вычислений, относительно обособленные машинные кластеры и даже суперкомпьютеры.

Еще одной функцией пилотных заданий является подготовка входных данных для задачи и выгрузка результатов. И здесь как и в случае вычислительной среды наблюдается большая неоднородность систем хранения (разные протоколы, разный функционал), которую пилотные задания могут унифицировать для вышестоящих слоев.

Пилотные задания могут принимать принимать пользовательские задачи с разных серверов, позволяя таким образом разделять один ресурс между различными системами.

Более подробно про пилотные задания написано в разделе 2.2.

- Пользовательские API и веб-интерфейс, позволяющие отправлять задачи в систему и выполняющие функции мониторинга.

- Общий файловый каталог, позволяющий унифицировать доступ к данным, хранимым на распределенных системах.

Основными нововведениями разработанной схемы по сравнению с классической схемой PanDA заключаются в следующем:

- Расширен список интегрируемых ресурсов за счет расширения схем работы пилотных заданий.
- Расширен список интегрируемых источников данных за счет отказа от ATLAS-специфических программных компонент, в частности, файлового каталога и системы управления данными.
- Расширена область применения системы за счет отказа от сертификатов CERN и расширения клиентских графических приложений для работы с системой.

Преимуществом разрабатываемой системы является отсутствие жесткой привязки к программному обеспечению, специфичному для эксперимента ATLAS. Файловый каталог Rucio, информационная система AGIS, система безопасности, использующая механизм виртуальных

организаций CERN, - компоненты системы, не позволяющие работать с PanDA разработчикам и ученым, не связанным с ATLAS. Благодаря изначально заложенным принципам универсальности и расширяемости, в нашей разработке мы постарались объединить наиболее общие подходы с целью получения вычислительной системы, способной объединить различные вычислительные инфраструктуры.

## 2 Реализация

### 2.1 Сервер

Сервер - центральная часть системы. Основой задачей сервера является управление потоком пользовательских задач, сопоставление их конкретным вычислительным узлам, а также формирование запросов на перемещение и регистрацию данных. На первом этапе своей работы сервер получает задачу от пользователя с помощью одного из нескольких доступных клиентов и распределяет ее в очередь конкретного вычислительного элемента, выбор которого обусловлен множеством факторов, таких как приоритет задачи, доступность ресурса и наличие необходимого ПО на нем, расположение входных данных. На следующем этапе при появлении очередного запроса от пилота на получение задачи, сервер передает всю необходимую для запуска задачи информацию и начинает отслеживать ее жизненный цикл. По окончании работы пилота сервер регистрирует полученные файлы результатов и сообщает пользователю, создавшему задачу, об окончании ее выполнения.

В качестве программной основы для сервера мы взяли актуальную программную реализацию сервера системы PanDA, расширив ее и отказавшись от использования специфичных для ATLAS компонент, в том числе:

- расширили возможности аутентификации пользователей, в том числе не имеющих личных сертификатов Грид.

отказались от ATLAS-специфической системы управления данными и каталога, для чего были разработаны свои реализации каталога размещения данных и системы управления данными.

Также была проведена работа по переходу к использованию СУБД MySQL вместо Oracle.

### 2.2 Подсистема пилотных заданий

Одним из ключевых компонентов системы является подсистема пилотных заданий (Pilot), реализующих операцию поздней привязки. После запуска пилотные задания собирают информацию о каждом конкретном вычислительном ресурсе. Если ресурс прошел валидацию на соответствие минимально-необходимым требованиям, то пилот считается успешно активированным, после чего сервер по запросу пилота передает ему рабочие задачи на основе критериев выбора ресурса. Таким

образом, задача начинает работу в уже проверенной и, при необходимости, подготовленной среде, привязываясь к ресурсу на максимально близкой к запуску стадии. 'Поздняя привязка' рабочих задач к месту вычислений предотвращает задержки и отказы, и максимизирует гибкость выделения ресурсов задачам при помощи динамического состояния обрабатывающих ресурсов и приоритетов задач. Пилотные задания являются также основным 'изолирующим слоем' системы, инкапсулирующим сложные неоднородные среды и интерфейсы, с которыми взаимодействуют слои более высокого уровня.

Запуск пилотных заданий на рабочих узлах обеспечивается независимой от центрального сервера подсистемой планировщиков ('фабрики пилотных заданий'). Количество запущенных пилотных заданий и политика их запуска (когда запускать пилотные задания, сколько пилотных заданий одновременно поддерживать и т.п.) определяются в конфигурационных файлах администраторами конкретной вычислительной среды (например, компьютерного кластера). Каждое пилотное задание "обслуживает" только одну очередь пользовательских задач на сервере. При этом Pilot имеет еще ряд дополнительных параметров, в соответствии с которыми фильтруются задачи в очереди. Некоторые параметры являются статическими (например, набор тегов, которые должна иметь задача), а некоторые формируются динамически (например, объем доступных ресурсов). Запущенное пилотное задание связывается с сервером, получает доступную пользовательскую задачу из требуемой очереди, удовлетворяющую заданным параметрам (например, по требованиям к объему свободного места на диске или к количеству доступных процессоров), инициирует полученную задачу и контролирует весь процесс ее выполнения.

В качестве программной основы была взята система пилотных заданий PanDA, расширенная для поддержки трех схем работы, представленных на рисунках 3-5.

Схема, представленная на рисунке 3, является классическим вариантом, используемым в системе PanDA для выполнения задач эксперимента ATLAS. Основными компонентами модуля пилотного задания являются:

Mover, осуществляющий загрузку в локальное хранилище входных данных, необходимых для выполнения пользовательской задачи, а также выгрузку и регистрацию результирующих данных.

RunJob, запрашивающий и получающий пользовательскую задачу с сервера и запускающий ее на выполнение на ресурсах вычислительной среды.

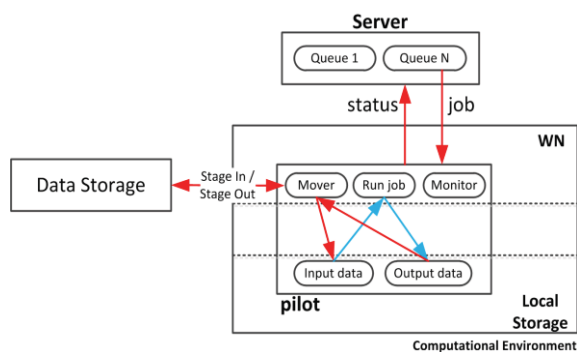


Рис. 3. Схема работы пилотных заданий на узлах вычислительной среды с доступом к WAN

— Monitor, предназначенный для контроля выполнения задачи и работой всех компонент пилотного задания.

Схема работы пилотного задания, представленная на рисунке 3, предполагает, что вычислительная среда, в которой будет выполняться пользовательская задача, имеет доступ к внешней сети (что позволяет напрямую обращаться к серверу и хранилищам данных). Такая конфигурация характерна для отдельных машин, кластеров, инфраструктур облачных вычислений. В этом случае пилотное задание запускается прямо на рабочих узлах вычислительной среды и работает с процессами и данными локально: данные загружаются в локальную файловую систему, обрабатываются как локальные файлы, пользовательская задача запускается локально. Отрицательным моментом схемы является сложность запуска параллельных задач, использующих, например, интерфейс передачи сообщений MPI, так как пользовательская задача запускается на ресурсах, уже выделенных пилотному заданию. Когда пилотное задание резервирует ресурсы, информация о том, какая пользовательская задача будет делегирована данному пилотному заданию и сколько узлов ей требуется, еще недоступна.

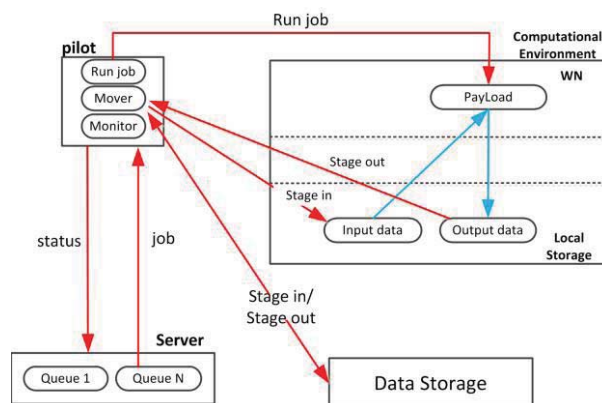


Рис. 4. Схема работы пилотных заданий на удаленном узле (вычислительная среда без доступа к WAN)



Схема, представленная на рисунке 4, предназначена для работы с вычислительными средами, в которых рабочие узлы не имеют доступа к WAN. Такая конфигурация характерна для большинства суперкомпьютеров. В этом случае для запуска пилотного задания требуется промежуточный узел, выступающий в роли шлюза между сервером и вычислительной средой. Проверка наличия ресурсов и их доступность проверяются до получения пользовательской задачи, но захват ресурсов для нее осуществляется после ее получения. В результате, пилотное задание может зарезервировать требуемый объем ресурсов при запуске пользовательской задачи, что позволяет запускать параллельные задачи. Входные и выходные файлы пересылаются через узел пилотного задания, на котором буферизуются. Отрицательными моментами данной схемы является резервирование ресурсов только после получения задачи от сервера (что увеличивает вероятность сбоя), а также повышенная нагрузка на узел пилотных заданий, для которого требуется большой объем ресурсов, в частности свободного места на жестком диске.

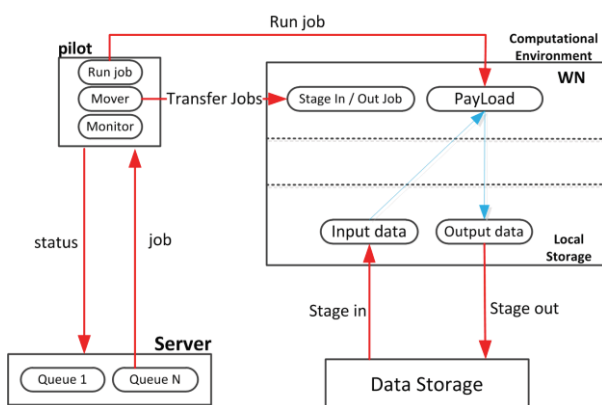


Рис. 5. Схема работы пилотных заданий на удаленном узле (вычислительная среда имеет доступ к WAN)

Схема, представленная на рисунке 5, является промежуточным вариантом между первыми двумя. В ней вычислительная среда также, как и в первой схеме, имеет доступ к WAN. Это дает возможность перемещать входные данные и результаты вычислений между хранилищами и вычислительной средой напрямую, что практически устраняет нагрузку на узел пилотных заданий. При этом сохраняется возможность запуска параллельных задач.

Первая из трех приведенных схем (Рисунок 3) является основной схемой в системе PanDA. Вторая схема (Рисунок 4) в несколько модифицированном частном виде также используется в PanDA для подключения суперкомпьютеров, например Titan [4]. Последняя схема (Рисунок 5) - оригинальная разработка, предложенная в данной работе.

## 2.3 Интерфейс формирования и запуска задач

Для более эффективного взаимодействия пользователей с системой нами был разработан специальный интерфейс, позволяющий загружать входные файлы из различных хранилищ, определять и запускать пользовательские задачи, контролировать процесс и отслеживать статус их выполнения (мониторинг). Данный интерфейс состоит из нескольких программных модулей. Непосредственное взаимодействие с пользователем берет на себя веб-интерфейс, который предоставляет унифицированную веб-форму для определения новых пользовательских задач. Преимуществом данного способа взаимодействия является отсутствие необходимости использования сертификатов CERN и механизма виртуальных организаций.

Перемещением пользовательских и системных файлов между распределенными вычислительными узлами занимается модуль транспортировки файлов. Также в круг задач этого модуля входит поддержка механизма репликации и обеспечение согласованности реплик. Все файлы, задействованные в работе, их реплики и правила доступа к ним описываются в специальном файловом каталоге. Совместное использование модуля транспортировки файлов и файлового каталога обеспечивают возможность быстрого подключения сторонних систем хранения (например, Dropbox, Google Drive и т.д.), что пока недоступно для основной версии PanDA.

С целью интеграции с внешними пакетами программ был создан API (интерфейс прикладного программирования), позволяющий производить действия, доступные пользователям веб-интерфейса, с помощью HTTP запросов.

Таким образом, разработанный интерфейс позволяет запускать задачи пользователям, не обладающим сертификатами, выданными центром сертификации CERN, и использовать системы хранения, не поддерживаемые основной версией PanDA.

## 3 Экспериментальные исследования

### 3.1 Экспериментальная версия

Мы развернули экспериментальную версию разработанной системы в НИЦ “Курчатовский институт”, подключив несколько вычислительных сред, включая:

- Суперкомпьютер НИЦ “Курчатовский институт”. Высокопроизводительный вычислительный кластер второго поколения с пиковой производительностью 122,9 TFLOPS сдан в эксплуатацию с сентября 2011 года. Кластер состоит из 1280 счётных двухпроцессорных узлов, объединенных высокопроизводительной сетью передачи данных и сообщений InfiniBand DDR, имеет суммарную оперативную память 20,5 Тбайт и систему хранения данных на 144 Тбайт. На счётных узлах кластера установлена операционная система

Linux (CentOS). Система хранения данных построена на параллельной файловой системе Lustre 2.0. Для управления распределением ресурсов и выполнением счетных заданий используется менеджер ресурсов SLURM. Характеристики узлов приведены в таблице 1.

Таблица 1. Характеристики узлов суперкомпьютера НИЦ “Курчатовский институт”

|   |       |
|---|-------|
| Счётные узлы на процессорах Intel Xeon E5450 (3,00 ГГц, 4 ядра) |       |
| Количество процессоров на узел                                  | 2     |
| Количество ядер на узел   | 8     |
| Оперативная память на узел (Гбайт)                              | 16    |
| Оперативная память на ядро (Гбайт)                              | 2     |
| Локальная дисковая память на узел (Гбайт)                       | 120   |
| Общее количество узлов  | 1280  |
| Общее количество процессоров                                    | 2560  |
| Общее количество ядер   | 10240 |
| Общая пиковая производительность (TFLOPS)                       | 122,9 |

Так как суперкомпьютер НИЦ “Курчатовский институт” имеет доступ во внешние сети, то для его интеграции мы использовали схему, представленную на рисунке 5. Более детальная схема интеграции представлена на рисунке 6.

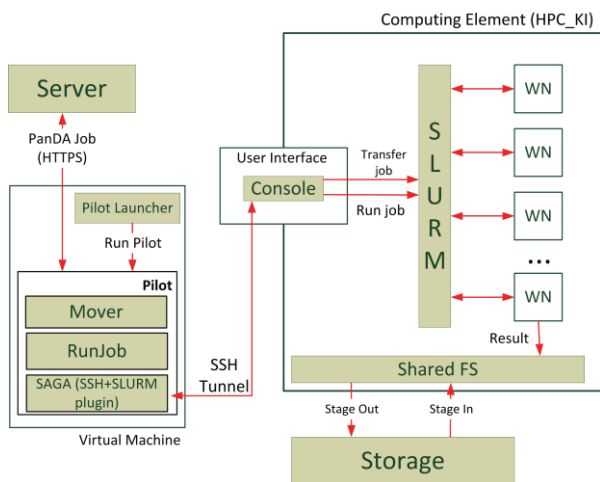


Рис. 6. Детальная схема интеграции суперкомпьютера НИЦ “Курчатовский институт” в режиме «один пилот на много ядер»

Пилотные задания запускаются на виртуальной машине, с которой доступ на суперкомпьютер осуществляется по протоколу SSH через машину пользовательского интерфейса, с которой можно ставить задачи в очередь SLURM. Для поддержки

SSH-канала и постановки задач в очередь SLURM используется интерфейс SAGA (Simple API for Grid Applications) [5], предназначенный для унификации использования различных распределенных промежуточных систем и сервисов. Для загрузки входных данных из удаленного хранилища и выгрузки результатов обработки используются отдельные задачи.

В то же время мы использовали классическую схему работы пилотных заданий (Рисунок 3) для работы с сервером PanDA, установленным в CERN (см. раздел 3.2 “Задания эксперимента ATLAS”). Более детальная схема подключения суперкомпьютера НИЦ “Курчатовский институт” по классической схеме для выполнения задач эксперимента ATLAS представлена на рисунке 7.

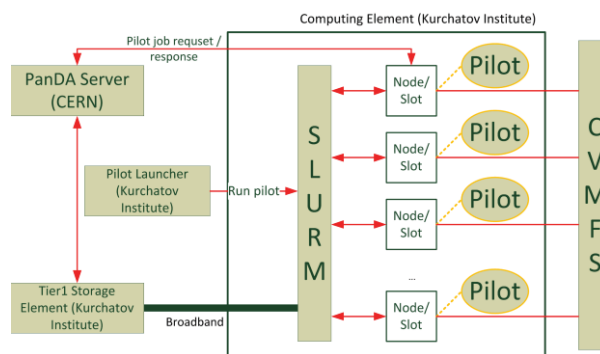


Рис. 7. Схема интеграции суперкомпьютера НИЦ “Курчатовский институт” для выполнения задач эксперимента ATLAS в режиме «один пилот на одно ядро»

Pilot Launcher запускает пилотные задания последовательно из расчета - одно задание на одно ядро. Для того чтобы выполнялись пользовательские задания по обработке данных эксперимента ATLAS, требуется программное обеспечение, доступное на подключаемой удаленной файловой системе CVMFS. Входные и выходные данные загружаются непосредственно в Grid Storage Element Tier1 НИЦ “Курчатовский институт”.

- Инфраструктура облачных вычислений, развернутая в НИЦ “Курчатовский институт”, состоящая из 16 узлов и дополнительного хранилища на 60 ТБ. Для ее интеграции мы использовали классическую схему работы пилота (Рисунок 3).

### 3.2 Задания эксперимента ATLAS

Начиная с ноября 2014 года проводился эксперимент, в котором на развернутой системе обрабатывались задачи эксперимента ATLAS. Для этого экспериментальный образец был адаптирован для работы с сервером системы управления задачами эксперимента ATLAS PanDA, установленным в CERN.

Адаптация была осуществлена путем создания

на центральном сервере PanDA в CERN отдельной очереди задач, к которой обращались пилотные задания, запускаемые на ресурсах НИЦ «Курчатовский институт». В ходе эксперимента примерно 200 пользовательских задач запускалось и успешно выполнялось ежедневно. Одним из наиболее важных исследований, решаемых на суперкомпьютере является реконструкция событий протон-протонового взаимодействия с высоким числом взаимодействия для изучения производительности Трекового Детектора Переходного Излучения (TRT).

На рисунке 8 приведена статистика выполнения задач эксперимента ATLAS на ресурсах НИЦ «Курчатовский институт» (служебные задачи на графике представлены светлым оттенком, пользовательские - темным).

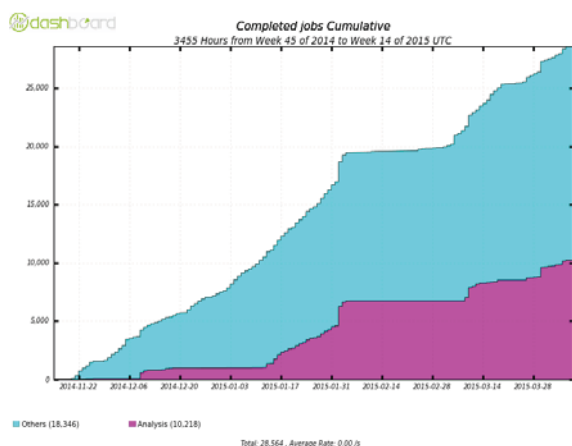


Рис. 8. Статистика успешно завершённых задач эксперимента ATLAS на суперкомпьютере НИЦ «Курчатовский институт»

Как видно из рисунка, суперкомпьютер НИЦ «Курчатовский институт» полностью интегрирован в рабочую схему эксперимента ATLAS. Задания поступают ежедневно. За время проведения эксперимента было обработано порядка 10-15 тысяч пользовательских заданий.

### 3.3 Задачи в области биоинформатики

Одной из целей данного проекта было осуществление запуска пользовательских задач, не связанных с Физикой Высоких Энергий и наиболее полно проявляющих преимущества разработки. В качестве первых задач были выбраны задачи выравнивания и сборки генома с помощью программного обеспечения Bowtie2 [6] и Abyss[7]. Данные программные пакеты в отличие от стандартных пакетов программ эксперимента ATLAS позволяют запускать задачи в параллельном режиме. Непосредственные вычисления осуществляются на суперкомпьютере НИЦ «Курчатовский институт». Первые тесты показали успешные результаты. С помощью пакета Quast[8] удалось расширить возможности мониторинга по пост-обработке выходных файлов - были сгенерированы специальные страницы отчетов успешности выполнения задач.

## Заключение

Представленный в статье подход к вычислениям и реализующая его система позволяют объединять вычислительные мощности различных инфраструктур для проведения масштабных вычислений в высокоинтенсивных областях науки, таких как Физика Высоких Энергий, биоинформатика и других. Данный подход применим в биологии для задач обработки данных геномного секвенирования, в астрофизике для изучения состава космических лучей, поиска антиматерии и тёмной материи, и т.п.

В рамках работы по развертыванию экспериментальной версии в НИЦ «Курчатовский институт» была произведена установка сервера системы PanDA, доработка и оптимизация которого позволила произвести интеграцию с суперкомпьютером НИЦ «Курчатовский институт». Во время тестирования ежедневно выполнялось более 200 пользовательских задач. Также были разработаны, обобщены и реализованы новые схемы работы пилотных заданий. Был разработан и реализован специальный веб-интерфейс и API для запуска задач и мониторинга их жизненного цикла. Тестирование всех компонентов системы проводилось на задачах по обработке данных эксперимента ATLAS и пользовательских задачах по биологии, которые достаточно требовательны к вычислительным ресурсам. В свою очередь, наши наработки применяются в работах по унификации компонентов PanDA.

## Литература

- [1] Worldwide LHC Computing Grid. Сайт проекта. Электронный ресурс. URL: <http://wlcg.web.cern.ch/>
- [2] Maeno, Tadashi. "PanDA: distributed production and distributed analysis system for ATLAS." *Journal of Physics: Conference Series*. Vol. 119. No. 6. IOP Publishing, 2008.
- [3] Collaboration, A. T. L. A. S., and G. Aad. "The ATLAS experiment at the CERN large hadron collider." *J. Instrum* 3 (2008): S08003.
- [4] T. Maeno et al. "Evolution of the ATLAS PanDA workload management system for exascale computational science" 2014 *J. Phys.: Conf. Ser.* 513 032062.
- [5] Goodale, Tom, et al. "SAGA: A Simple API for Grid Applications. High-level application programming on the Grid." *Computational Methods in Science and Technology* 12.1 (2006)
- [6] Программа для выравнивания последовательностей. Сайт проекта. Электронный ресурс. URL: <http://bowtie-bio.sourceforge.net/bowtie2/manual.shtml>
- [7] Simpson, Jared T., et al. "ABySS: a parallel assembler for short read sequence data." *Genome research* 19.6 (2009): 1117-1123.

- [8] Программа для оценки качества сборок. Сайт проекта. Электронный ресурс. URL: <http://bioinf.spbau.ru/ru/quast>

### **Integrated Workload and Data Management System in a Heterogeneous Computing Infrastructure**

Alexei Klimentov, Ruslan Mashinistov, Alexandr Novikov, Alexey Poyda, Ivan Tertychnyy

In this paper an approach to manage computing powers of grids, cloud platforms, supercomputers as well as university and research clusters for computational tasks in data intensive domains is described. This approach uses the experience of the workload management system PanDA (Production and distributed analysis) which was developed at CERN for the ATLAS experiment. Based on this approach, we designed and implemented Workload and Data Management System (WDMS) at NRC “Kurchatov institute”. WDMS is currently in production for High Energy and Nuclear Physics and bioinformatics applications and it can be used for other data- and compute-intensive scientific workflows.