

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ «НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЯДЕРНЫЙ
УНИВЕРСИТЕТ «МИФИ»

УДК 004.415.2

ОТЧЕТ
О ПРЕДДИПЛОМНОЙ ПРАКТИКЕ

**СИСТЕМА УПРАВЛЕНИЯ ПРОЦЕССАМИ
ОБРАБОТКИ В ГЕОГРАФИЧЕСКИ
РАСПРЕДЕЛЕННОЙ СИСТЕМЕ
ОБРАБОТКИ ДАННЫХ SPD**

Студент _____ Н.Г. Монаков

Научный руководитель _____ Е. Ю. Солдатов

Научный консультант _____ А.Ш. Петросян

Москва 2025

Содержание

Введение	3
1 Физическая установка	4
1.1 Комплекс NICA	4
1.1.1 Системы комплекса NICA	4
1.1.2 Основные направления научно-исследовательской программы	5
1.1.3 Ускорительный блок	8
1.2 Эксперимент SPD	13
2 Система управления заданиями и взаимосвязь IT-составляющих системы	14
2.1 Реализация подобной системы ATLAS	14
2.1.1 ProdSyS	15
2.1.2 Rucio	16
2.1.3 PanDA	19
2.2 Реализация подобной системы COMPASS	28
3 Анализ технологий для построения системы	30
3.1 Выбор фреймворка для приложения	30
3.2 OAuth 2.0	32
3.3 JWT	32
3.4 Apache2	32
4 Разработка прототипа веб-приложения	33
5 Создание веб-сервера	36
6 Механизм аутентификации и авторизации	37
Заключение	39
Список литературы	39

Введение

Любой физический эксперимент ставит перед собой амбициозные теоретические и технические задачи, но работа над экспериментами в рамках мега-сайнс проектов, таких как эксперимент SPD (Spin Physics Detector) на проекте NICA (Nuclotron based Ion Collider fAcility), поднимает и последующие этапы сбора, хранения и обработки полученных данных на самый высокий технический уровень. Системы и подсистемы, управляющие процессами на этих этапах, должны быть реализованы, кроме прочего, с учетом возможности изменения как ресурсов самих систем, так и объёмов поставляемых данных с физической части эксперимента. В ходе научно-исследовательской работы планируется изучить существующий мировой опыт построения систем, ответственных за управление процессами обработки данных, в том числе географически распределенных, встроиться в создание и последующее поддержание соответствующей системы для эксперимента SPD.

1 Физическая установка

1.1 Комплекс NICA

Цель проекта «NICA» [1] заключается в создании на территории Российской Федерации экспериментальной базы мирового уровня для проведения фундаментальных исследований по ряду ключевых направлений современной физики высоких энергий, а также прикладных исследований. Кроме того, проект предусматривает участие учёных из научных организаций стран-участниц и других государств, присоединившихся к проекту, что также делает эксперимент крупным и важным международным проектом в сфере теоретической физики.

1.1.1 Системы комплекса NICA

В состав комплекса входят следующие основные системы:

1. Ускорительный блок, включающий:

- сверхпроводящий синхротрон «Нуклотрон» с системами вывода и перенаправления пучков
- инжекционный комплекс (источники ионизированных и поляризованных частиц, линейные ускорители)
- бустер в виде сверхпроводящего синхротрона
- сверхпроводящий коллайдер тяжёлых ионов и поляризованных частиц

2. Экспериментальные установки:

- MPD (MultiPurpose Detector) — установка для исследования плотного барионного вещества в сталкивающихся пучках
- BM@N (Baryonic Matter at Nuclotron) — установка для исследования плотного барионного вещества с использованием выведенных из Нуклотрона пучков
- SPD (Spin Physics Detector) — установка для изучения спиновой структуры нуклона в опытах на поляризованных пучках

3. Инновационный блок, включающий экспериментальные зоны, каналы и стенды для:

- прикладных и инновационных исследований
- разработки микроэлектроники
- решения задач биомедицины и материаловедения

4. Вычислительно-информационный блок, включающий:

- распределённый информационно-вычислительный комплекс для моделирования, обработки, анализа и хранения накопленных данных
- сетевую инфраструктуру
- набор информационных сервисов

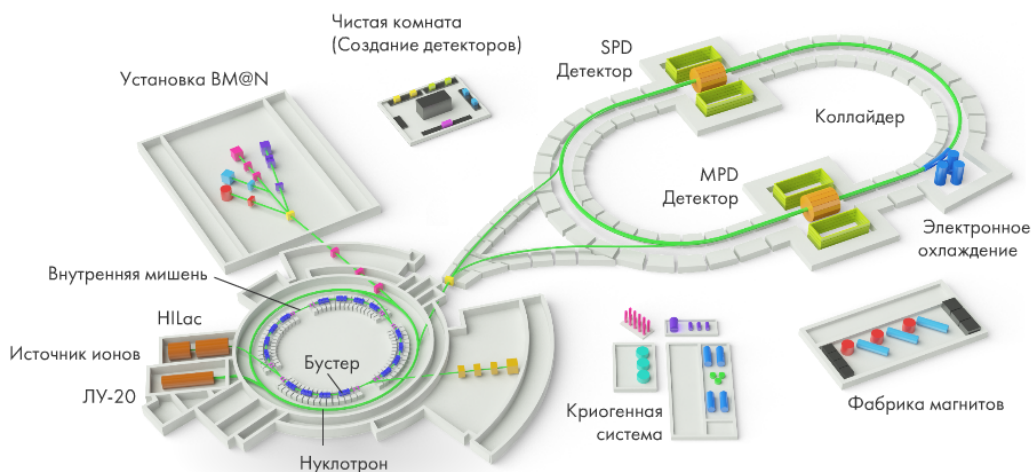


Рисунок 1 — Комплекс NICA

Для проведения фундаментальных исследований на комплексе NICA были налажены внутривосийские и международные научные взаимодействия, объединяющие учёных, инженеров и специалистов из стран-участниц проекта. В инновационной и прикладной деятельности задействованы также коллективы ОИЯИ, включающие специалистов из разных государств, что позволяет решать широкий спектр задач.

Говоря о фундаментальных исследованиях, коллаборация ставит перед собой цели, охватывающие широкий спектр явлений, проявляющихся в реакциях с тяжёлыми ионами, поляризованными адронами и лёгкими ядрами и связанных со структурой вещества, участвующего в сильных взаимодействиях.

1.1.2 Основные направления научно-исследовательской программы

1. Поиск и экспериментальное исследование фазовых переходов и критических явлений в сильно взаимодействующей ядерной материи при экстремальных барионных плотностях. [2]

Такая материя существовала лишь на ранних стадиях эволюции Вселенной, когда плотность и температура вещества были экстремально высоки, при невысоких же температурах она может возникать внутри ядер или нейтронных звёзд. Расчёты квантовой хромодинамики на решетках (КХДР) предсказывают фазовые переходы, связанные с деконфайнментом и восстановлением киральной симметрии при достаточной плотности энергии, что приводит к формированию кварк-глюонной плазмы. Однако современные КХДР-методы не охватывают область высоких барионных плотностей, характерную для столкновений тяжёлых ионов на комплексе NICA, что, несомненно, повышает значимость получения ожидаемых экспериментальных данных. Энергетический диапазон комплекса позволяет изучать как адронную, так и кварк-глюонную фазы, и рассматривая температуру и барионную плотность, нормированные к «нормальным» условиям, в качестве ключевых наблюдаемых характеристик, косвенно указывающих на статистические закономерности в процессах столкновений.

Кроме того, важно не только зарегистрировать сам фазовый переход, но и установить его природу: будет ли это непрерывный переход или фазовый переход первого рода с критической конечной точкой. До настоящего времени ни первое, ни второе не наблюдалось экспериментально, что делает поиск этого перехода одной из приоритетных задач программы. Поскольку теоретические предсказания в области конечных барионных плотностей пока ограничены, эксперименты начнутся с изучения диагностических наблюдаемых, уже исследованных на ускорителях RHIC и SPS: выходов частиц и их спектров, флуктуаций поперечного импульса, а также комбинированных распределений при различных энергиях и типах пучков. Будут исследоваться фемтоскопические корреляции, направленные, эллиптические и более высокие гармоники для разных адронов, лептонов и фотонов, а также асимметрии по заряду и спину.

2. Экспериментальное изучение спиновой структуры нуклона и лёгких ядер. [3]

Современное описание спиновой структуры адронов строится через набор распределённых функций, зависящих от спина и поперечного импульса, при этом особую роль играют корреляции между различными компонентами поперечного импульса. В программе комплекса будет исследоваться полный набор TMD-функций (transverse momentum-dependent) ведущего порядка с помощью жёстких зондов: дилептонов в Дрелл-Яновских процессах, квараквониев, прямых фотонов, адронов с большим поперечным импульсом. Кроме того, планируется изучение интегрированных по поперечному импульсу партонных распределений и кварк-глюонных корреляторов. Особое

внимание будет уделено тензорным распределениям поляризованного дейтрона – уникальной возможности комплекса.

3. Исследование поляризационных эффектов в столкновениях тяжёлых ионов и в системах с несколькими нуклонами.

Поляризация гиперонов является чувствительным индикатором динамики адронной и кварк–глюонной среды, включая такие характеристики, как гидродинамическая завихренность и спиральность. Программа включает изучение поляризации гиперонов в различных кинематических областях в зависимости от типа ядер, энергии и центральности столкновения. Планируется также сравнение поляризационных эффектов в реакциях с адронами, лёгкими и тяжёлыми ядрами, а также систематическое исследование тензорной поляризации векторных мезонов и дилептонов.

4. Изучение динамики реакций и модификаций свойств адронов в ядерной среде. [4] [5] [6]

Теоретические исследования указывают на возможность частичного восстановления хиральной симметрии, что приводит к изменению спектральных функций адронов в плотной среде. Наиболее чувствительными зондами таких модификаций для векторных мезонов являются дилептоны: их угловые распределения и тензорная поляризация отражают состояние среды в момент взаимодействия и не искажаются сильными взаимодействиями на более поздних стадиях. В настоящее время отсутствуют данные по дилептонам с инвариантной массой в диапазоне нескольких ГэВ; программа NICA устранил этот пробел, что особенно важно в связи с достижением максимальной плотности адронов именно в этой области.

5. Исследование структуры ядер на межнуклонных расстояниях, порогового образования странных гиперонов и поиски гиперядер в опытах с фиксированной мишенью и пучками, извлечёнными из Нуклотрона.

Флуктуации на коротких расстояниях, связанные с концепциями флуктонов и кумулятивных процессов, могут быть изучены в обратной кинематике при рассеянии ядерных пучков на водородной мишени, что позволяет детектировать все частицы в конечном состоянии. Рассеяние поляризованных дейтронов даст возможность прямой проверки тензорной природы флуктуаций. Усиленное образование странности в столкновениях тяжёлых ионов по сравнению с элементарными процессами является чувствительным признаком фазовых переходов; это явление особенно наглядно при образовании гиперядер, которое пла-

нируется исследовать как на столкновениях пучков при максимальной барионной плотности, так и на фиксированных мишенях.

6. Разработка теоретических моделей изучаемых процессов и теоретическое сопровождение экспериментов.

Теоретическая программа включает детальное моделирование процессов в конкретных экспериментальных условиях комплекса, а также исследовательские разработки новых методов, в частности непертурбативной и решётчатой квантовой хромодинамики.

1.1.3 Ускорительный блок

Ускорительный блок проекта NICA предназначен для генерации пучков заряженных частиц, которые используются экспериментальными установками комплекса для реализации научных программ в множестве областей физики. Как упоминалось ранее, речь идет о:

- релятивистской ядерной физике
- спиновой физике
- радиобиологии
- множестве прикладных направлений

Важно отметить тот факт, что комплекс обладает достаточно уникальными характеристиками, а именно:

- разнообразием состава и структуры ускоряемых ионных пучков
- гибкостью в реализации исследовательских программ
- высокой светимостью при изучении взаимодействий частиц в доступном энергетическом диапазоне
- множеством прикладных направлений

что делает работы по обеспечению его успешного функционирования важным вкладом в развитие современной российской и мировой науки.

Структура блока приведена на рисунке 2 и включает в себя следующие компоненты:

1. Инжекционная цепочка для лёгких ионов $\frac{q}{A} = \frac{1}{3}$

предназначена для ввода в Нуклотрон пучков различных ионов от протонов до магния, включая: поляризованные протоны с энергией до 12 МэВ и дейтроны с энергией до 6 МэВ/нуклон. Основные элементы цепочки:

- лазер
- дуоплазматрон
- SPI (источник поляризованных частиц)
- резонансный линейный ускоритель ЛУ-20М
- каналы распределения пучков между компонентами системы

В зависимости от задач эксперимента может использоваться один из трёх перечисленных источников. Параметры источников и соответствующих пучков приведены в таблице 1.

Параметр	Источник			
	Лазер	Дуоплазматрон	SPI	
Частицы	ионы до Mg^{10+}	H^+, D^+	He^{2+}	$\uparrow H^+, \uparrow D^+$
Частиц на импульс	$\sim 10^{11}$	$\sim 5 \cdot 10^{12}$	$\sim 10^{11}$	$5 \cdot 10^{11}$
Частота импульсов, Гц	0.5	1	1	0.2

Таблица 1 — Параметры источников лёгких ионов

Созданный пучок попадает в модернизированный ускоритель ЛУ-20М, который представляет собой совокупность форинжектора на основе трех секций ускорителя с пространственно-однородной квадрупольной фокусировкой, ускорителя типа Альварез ЛУ-20, а также промежуточных линий транспортировки пучка низкой и средней энергии.

Отношение заряда иона к его массе q/A	1.0	0.5	≥ 0.3
Энергия инжекции, [КэВ]	31	61.8	103
Максимальный ток, [мА]	10	20	10
Выходная энергия [МэВ/нуклон]	0.156		
Эмиттанс (на выходе) [$\pi \cdot \text{см} \cdot \text{мрад}$]	≤ 0.5		
Передача, %	> 85	> 89	> 93
В аксептансе ЛУ-20, %	70	71	80

Таблица 2 — Основные параметры форинжектора

Номер гармоники	$2\beta\lambda$
Энергия инжекции	156 КэВ/нуклон
Выходная энергия	5 МэВ/нуклон
Рабочая частота	145 МГц
Диаметр промежутка	1,4 м
Длина промежутка	14,4 м
Число дрейфовых трубок	57 + 2 полу-трубки
Фактор качества	40000
Синхротронная фаза	31, 5°
Фокусирующая структура	FODO
Градиент магнитного поля в квадрупольных линзах	(58.4 ÷ 7.4) Тл/м
Акseptанс	220 $\pi \cdot \text{мм} \cdot \text{мрад}$

Таблица 3 — Основные параметры ЛУ-20М

2. Инжекционная цепочка для тяжелых ионов $\frac{q}{A} = \frac{1}{6}$

предназначена для ввода в Бустер пучков ионов вплоть до золота с энергией до 3.24 МэВ/нуклон. Основные элементы цепочки:

- криогенный источник многозарядных тяжёлых ионов КРИОН-6Т
- высокочастотный резонансный линейный ускоритель НЛАС
- каналы распределения пучков между компонентами системы

Источник разработан в ОИЯИ и основан на явлении, обнаруженном при изучении режимов работы электронно-лучевого источника EBIS [7], работающего в отражательном режиме. Суть явления заключается в проявлении облаком многократно отраженных электронов, заключенных в сильное магнитное поле соленоида, свойств, характерных для фазового перехода; что имеет результатом постепенное увеличение плотности электронной плазмы и переходу в стационарное состояние, называемое электронной струной. В таблице 4 приведены параметры источника и пучка, создаваемого данным источником.

Частицы в основном режиме	Au^{31+}
Частиц на импульс	$\sim 2,5 \cdot 10^9$
Рабочая частота, Гц	0.6
Максимальная частота, Гц	10

Таблица 4 — Параметры источника тяжёлых ионов

Испущенный пучок через линию транспортировки пучка низкой энергии попадает в секцию с пространственно-однородной квадрупольной

фокусировкой, где ускоряется до 300 КэВ/нуклон. Оттуда по линии транспортировки пучка средней энергии пучок попадает в две последовательно расположенные секции линейных ускорителей с дрейфовыми трубками с квадрупольным дублетом, обеспечивающие выходную энергию 3.2 МэВ/нуклон. Ускоренный пучок по линии транспортировки отправляется в Бустер для последующего ускорения.

3. Синхротроны

Бустер

Данный синхротрон выступает промежуточным ускорителем тяжелых ионов и их инжектором в Нуклотрон. Исходя из этого он выполняет следующие задачи:

- **аккумуляция** ионов, ускоренных НІАс, до уровня 2×10^9 ионов Au^{31+}
- **ускорение частично ионизированных ионов** благодаря обеспечению сверхвысокого вакуума в камере пучка
- **формирование необходимого фазового объема пучка** посредством системы электронного охлаждения
- **ускорение тяжёлых ионов** до энергии, необходимой для их полной ионизации
- **полная ионизация тяжелых ионов** на мишени, установленной в канале
- **инжекция подготовленного пучка в Нуклотрон** с предварительным отсеиванием ионов с нецелевым зарядом

Сам синхротрон представляет из себя установку периметром 210.96 метров, состоящую из 4 сверхпроводящих арок, каждая из которых имеет угол поворота 90° и состоит из 5 периодов, каждый из которых, свою очередь, содержит фокусирующую и дефокусирующую квадрупольные линзы, 2 дипольных магнита и 4 малых свободных промежутка, предназначенных для размещения мультипольных корректоров, коллиматоров и диагностического оборудования. Между собой арки соединены прямыми участками, на которых размещаются системы ввода и вывода пучка, система электронного охлаждения, а также высокочастотная система ускорения и управления пучком.

Проектные возможности установки позволяют ей выводить в Нуклотрон пучок тяжелых ионов с энергией до 578 МэВ/нуклон.

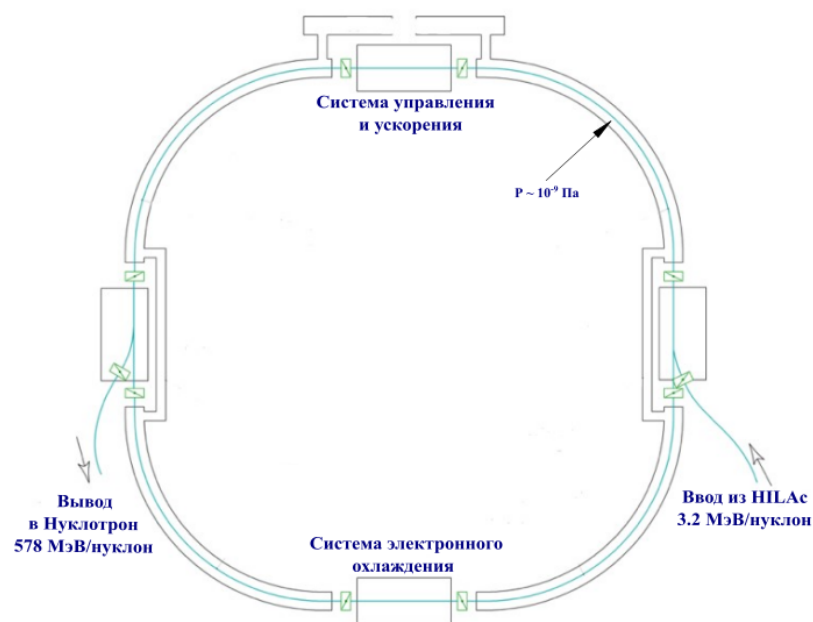


Рисунок 2 — Схема Бустера

Нуклотрон

Второй синхротрон является основным ускорителем всего комплекса и инжектором в накопительные кольца. Для самого же Нуклотрона, как было указано, инжекторами являются ЛУ-20М для легких и Бустер для тяжелых ионов соответственно. Из этого вытекают основные задачи, поставленные перед установкой, а именно:

- ускорение лёгких ионов для экспериментов на фиксированных мишенях
- ускорение тяжелых ионов для экспериментов на фиксированных мишенях
- ускорение различных ионов для экспериментов на внутренней мишени
- ускорение лёгких ионов для последующей инжекции в накопительные кольца
- ускорение тяжелых ионов для последующей инжекции в накопительные кольца
- ускорение различных ионов для последующей инжекции в накопительные кольца при работе в несимметричной моде

Совокупность 96 дипольных и 64 квадрупольных сверхпроводящих магнитов, разделенная на 8 суперпериодов, создаёт в кольце Нуклотрона периметром 251.5 м магнитное поле силой 1.8 Тл с магнитной жесткостью 38.5 Тл·м, что соответствует кинетической энергии:

- для протонов - 10.7 ГэВ
- для дейтронов и легких ионов с $Z/A = 1/2$ - 4.95 ГэВ/нуклон
- для ионов с $Z/A = 1/3$ - 2.6 ГэВ/нуклон
- для ионов золота - 4.95 ГэВ/нуклон

После ускорения пучки выводятся посредством системы медленно-го вывода, работающей по двухступенчатой схеме с эффективностью превышающей 95%, в павильоны для экспериментов на фиксированных мишенях и системы быстрого вывода, состоящей из магнита Ламбертсона и импульсного ударного магнита, в накопительные кольца Коллайдера.

4. **Сверхпроводящие накопительные кольца**
5. **Транспортные каналы для передачи пучков между элементами комплекса**

1.2 Эксперимент SPD

Экспериментальная установка SPD [8,9], строящаяся в одной из двух точек пересечения пучков коллайдера NICA, предназначена для всестороннего изучения спиновой структуры протона и дейтрона.

Основное внимание будет уделено изучению их поляризованной глюонной компоненты в реакциях инклюзивного рождения чармониев, открытого чарма и прямых фотонов, а также прочих спинзависимых явлений в столкновениях поляризованных пучков протонов и дейтронов с энергией в системе центра масс до 27 ГэВ и светимостью до $10^{32} \text{ см}^{-2} \text{ с}^{-1}$, которая обеспечит поток данных $\sim 20 \text{ ГБ/с}$ [10]. Посредством измерения соответствующих спиновых асимметрий будут получены данные по корреляциям между направлениями спина протона (дейтрона), его импульса, а также направлением спина, продольным и поперечным импульсами глюонов внутри протона (дейтрона). На первом этапе работы установки основное внимание планируется уделить изучению спиновых эффектов в упругих p-p и d-d рассеяниях, поиску мультипартонных корреляций и новых связанных состояний.

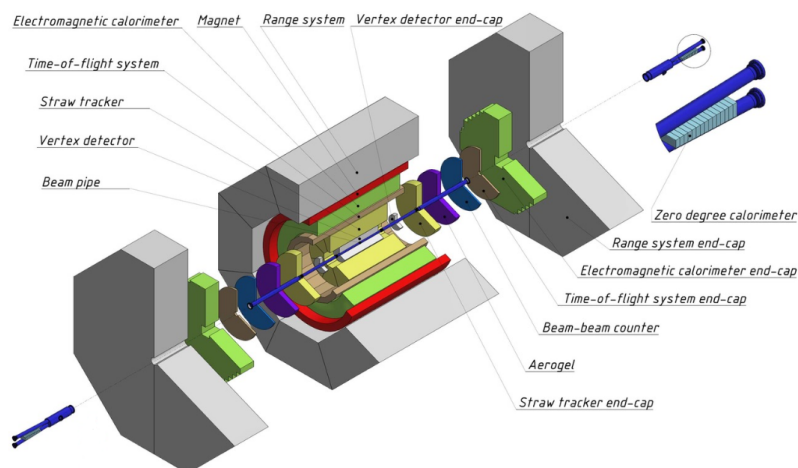


Рисунок 3 — Схема SPD

2 Система управления заданиями и взаимосвязь ИТ-составляющих системы

Как было упомянуто, ожидаемая скорость поступления данных будет порядка 10 петабайт в год, что даже с учетом онлайн-фильтра, отсеивающего около 90% данных, будет приводить к постоянному накоплению данных для обработки, кроме того, данные будут использоваться множеством разных физических групп, с соответственно разными программами исследований, разными нуждами в программном обеспечении и необходимых ресурсах. Ввиду этого возникает необходимость существования система высокого уровня, которая позволит наиболее эффективным образом организовывать создание, обработку и отслеживание заданий в условиях постоянного дефицита производительных мощностей, ожидаемого исходя из количества событий/год $\sim 10^{12}$ размером по 10-15 КБ каждое, с затратами на обработку от 1 до 10 секунд, и ведущего к необходимости постоянной работы 6-60 тысяч процессоров.

Проблема нехватки мощностей при работе с большими объемами данных с экспериментальных установок, конечно, не уникальна для SPD и возникала, практически, во всех экспериментах в физике элементарных частиц или физике высоких энергий с начала этого века. Основой этих решений, как правило, выступают вычислительные кластеры и модульный подход к созданию сервисов. Для начала рассмотрим некоторые примеры технических решений, разработанных и используемых в рамках экспериментов на SPS(Super Proton Synchrotron) и LHC(Large Hadron Collider).

2.1 Реализация подобной системы ATLAS

Коллаборацией ATLAS(A Toroidal LHC Apparatus) [11] разработан набор взаимодействующих систем и сервисов для хранения и обработки данных. Верхним уровнем системы является ProdSys(Production System) [12]. Это система, ответственная за управление процессами обработки данных, управляющая и/или взаимодействующая с прочими имплементированными сервисами, вроде системы управления данными Rucio [13], системы управления нагрузкой PanDA(Production and Distributed Analysis) [14], информационной системы CRIC(Computing Resource Information Catalogue) [15].

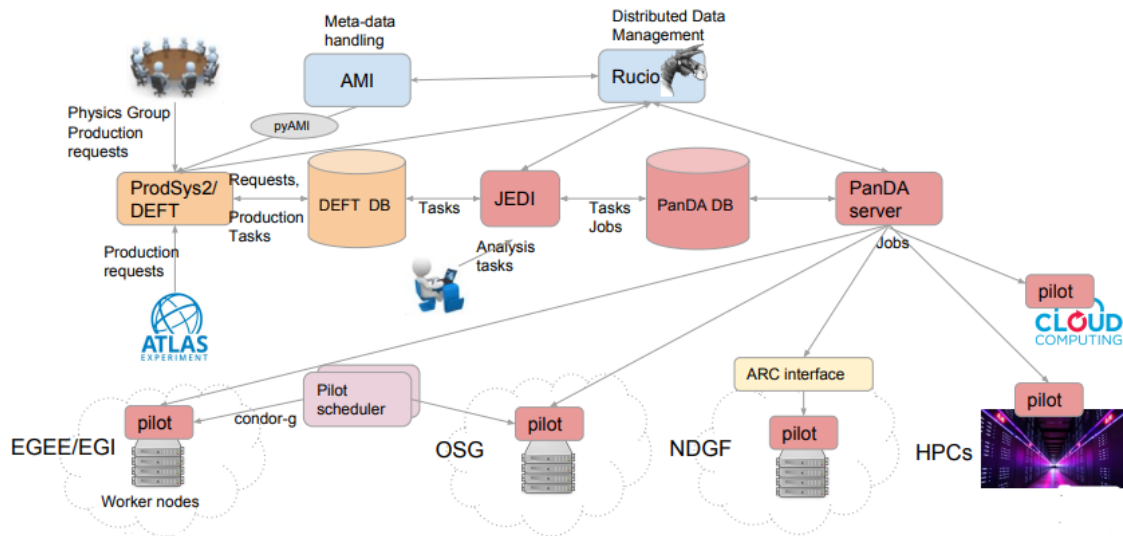


Рисунок 4 — Взаимосвязь сервисов и систем работы с данными ATLAS

2.1.1 ProdSyS

Система разработана для запуска заданий на разнообразном наборе вычислительных ресурсов для большого числа пользователей с множеством различных комплексных рабочих процессов. Чтобы успешно справляться с вытекающими из вышеперечисленного вызовами, быть способной развиваться и приспосабливаться к новым требованиям, система разделена на несколько независимых централизованных сервисных уровней.

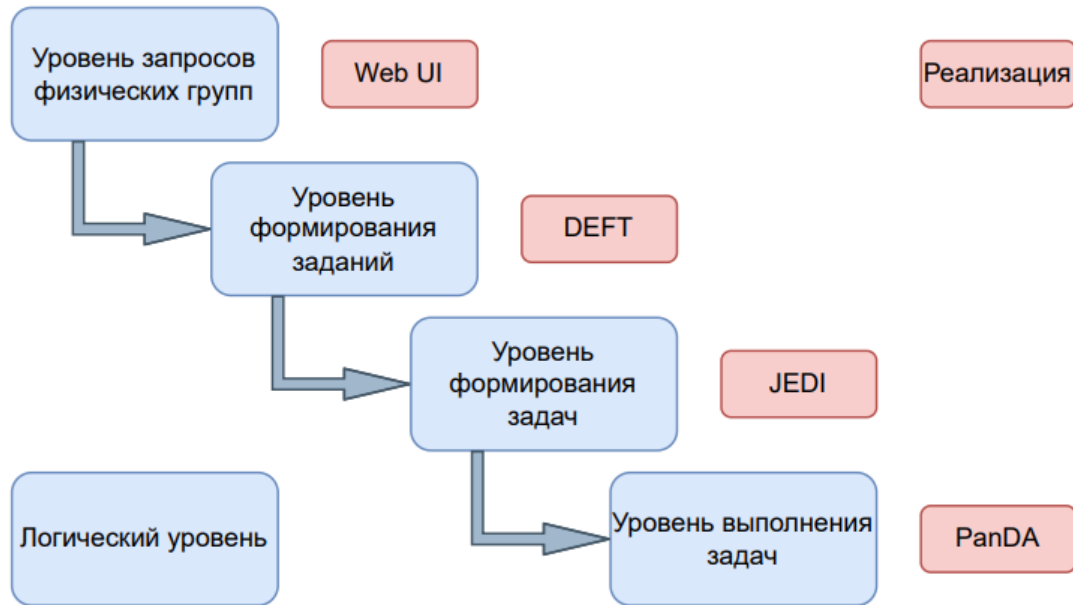


Рисунок 5 — Компоненты ProdSyS

При этом, иницилируя процесс обработки данных, физические группы взаимодействуют со всей системой ProdSyS только на верхнем уровне и работают не с конкретными задачами и файлами, а с их совокупностями: заданиями, датасетами и контейнерами соответственно. В рамках работы группы система предоставляет определённым представителям группы роль Production Manager для обеспечения возможности определять поток обработки данных для заданий группы, управлять заданиями, определять требуемые ресурсы и прикладное программное обеспечение. Именованное создаваемых объектов реализовано в специальном формате, представленном на рисунке 6, для упрощения, унификации работы и взаимодействий подсистем, а также для удобства физических групп.



Рисунок 6 — Структура имен заданий и датасетов

DEFT(Database Engine For Tasks) отвечает, как следует из рисунка 5, за формирование заданий, а также за хранение их состояний для систем обработки данных и общей системы мониторинга.

2.1.2 Rucio

Rucio [16]— это фреймворк с открытым исходным кодом, разработанный для организации, управления и обеспечения доступа к данным в рамках

данного эксперимента и в дальнейшем распространившийся в своем использовании как на другие эксперименты на БАК(Большой Адронный Коллайдер), так и на сторонние эксперименты.

Основные используемые понятия:

- **Пространство имён** отвечает за адресацию данных различными способами
- **Учётные записи** обеспечивают авторизацию, аутентификацию и управление правами доступа
- **Хранилище** предоставляет единый интерфейс к распределённым дата-центрам
- **Подписки** используются для реализации политик передачи данных на масштабах всей системы
- **Правила репликации** гарантируют консистентное распределённое состояние пространства имён на хранилище

Ключевые особенности Rucio:

- **масштабируемость:** поддержка распределённого хранения данных в гетерогенных центрах обработки данных, расположенных географически удалённо. В рамках эксперимента система продемонстрировала успешную работу с более чем двумя сотнями элементов хранения.
- **оптимизация потока данных:** хотя в общем случае Rucio не взаимодействует напрямую с физической сетевой инфраструктурой, оно может использовать накопленные за историю работы сетевые метрики — пропускную способность, потерю пакетов и задержку — для выбора оптимальных путей передачи данных
- **гибкость:** поддержка разнородных систем хранения, таких как дисковые массивы, ленточные библиотеки, облачные хранилища. Это достигается благодаря встраиванию существующих или добавлению любых кастомных протоколов в систему посредством плагинов, использующих внутренние интерфейсы взаимодействия Rucio, имитирующие стандартные POSIX-операции
- **приближенное удовлетворение ACID:** глобальный учёт данных с поддержкой транзакционности

- **интегрируемость:** бесшовная интеграция с ProdSyS и PanDA, возможность интеграции сторонних сервисов и систем
- **разграничение доступа:** реализация системы идентификации, аутентификации и авторизации, в том числе с использованием сертификатов и сторонних систем, в частности SSO
- **хранение метаданных:** для обеспечения иерархической организации данных с использованием правил репликации

Данные в Rucio организуются с помощью идентификаторов данных (Data Identifiers), которые представляют собой многоуровневую структуру, приведенную на рисунке 7.

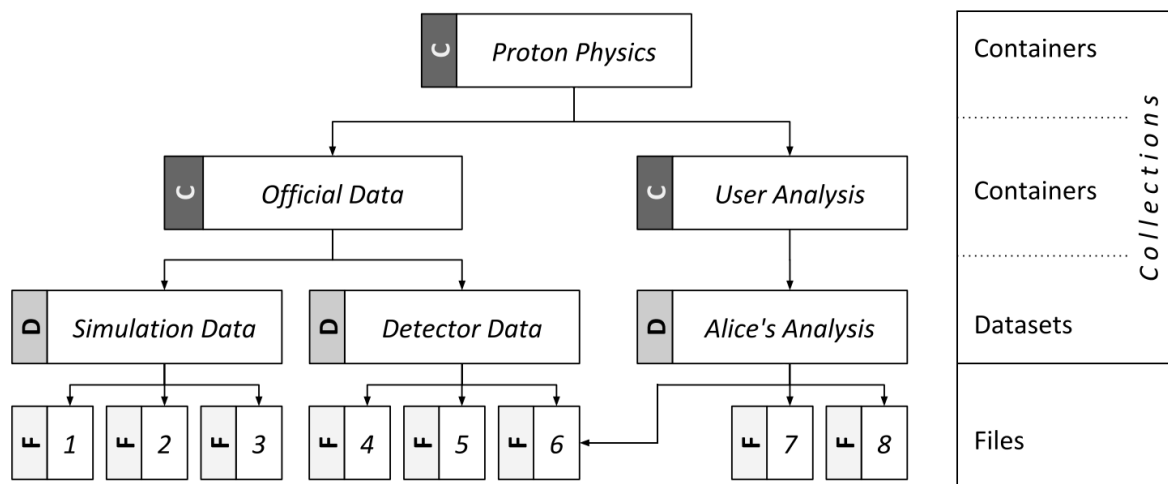


Рисунок 7 — Структура пространства имен

Наименьшая единица, которой оперирует система - реальный файл на хранилище. Файлы объединены в логические группы - датасеты, предназначенные для упрощения массовых операций и структурирования данных. Важным пунктом для распределенной системы тут является возможность объединения в датасет как сугубо локальных файлов, так и файлов с разных точек хранения. Датасеты, в свою очередь, также могут быть объединены в связный логический объект - контейнер, что позволяет единообразно оперировать еще большими объемами данных и организовывать эти самые данные в иерархичную структуру произвольной глубины.

В целях унификации подхода к хранению и удобства оперированием файлами и коллекциями (датасетами и контейнерами) их именование подчиняется согласованной структуре, в формате `<scope>:<name>`, где `<scope>` разделяет глобальное пространство имён, что повышает гибкость организации и предоставления прав, а `<name>` - уникально идентифицирует сам объект внутри `<scope>`.

Как было сказано ранее, Rucio хранит метаданные для обеспечения корректной организации файлов, в частности: чек-сумму, вычисляемую с

использованием алгоритмов MD5 или Adler32, каждого файла, зарегистрированного в системе, проверяемую при любом обращении к файлу, будь то использование для анализа, перемещение или добавление в коллекцию, что позволяет поддерживать консистентность хранимых и используемых данных; статус доступности, отображающий наличие правил репликации и хотя бы одной реплики на хранилище; флаг отображения файла в коллекциях и прочие метаданные.

Также имеются встроенные средства мониторинга, используемые для отслеживания состояния системы, объемов и типов управляемых ресурсов, доступности систем хранения и эффективности взаимодействия между этими системами.

2.1.3 PanDA

Система распределения и обработки рабочих нагрузок, спроектированная для работы на масштабах эксперимента LHC. PanDA была разработана для удовлетворения комплексных требований по обработке данных, моделированию детектора и физическому анализу в рамках эксперимента ATLAS, после чего, аналогично Rucio, применялась на ряде других установок по исследованию в области физики высоких энергий и не только.

В рамках своего дебютного эксперимента система успешно справлялась с:

- оркестрацией выполнения заданий на более чем двух сотнях вычислительных центров в 40 странах по всему миру
- распределением нагрузки, поступающей от тысяч ученых
- объемом нагрузки на уровне экзобайт
- встраиванием сторонних прикладных программ и сценариев выполнения обработки

Одним из ключевых принципов проектирования PanDA стала тесная интеграция управления нагрузкой и данными — в частности, с вышеописанной системой распределённого управления данными Rucio — что позволило эффективно обрабатывать около 100 ПБ данных ежегодно. Также проектирование велось с учётом гибкости к адаптации новых технологий в области обработки, хранения данных, сетей и распределённых вычислительных сред. Система справилась с бесшовной интеграцией широкого спектра ресурсов: от Всемирной вычислительной сети LHC (Worldwide LHC Computing Grid или WLCG) и волонтерских вычислений до независимых от WLCG систем высокопроизводительных вычислений, лидирующих вычислительных центров и коммерческих облачных сервисов.

Краеугольным понятием в работе PanDA является термин "**Задание**" или "**Task**". Это объект с чётко определёнными состояниями, соответствующий одной **рабочей нагрузке**. По сути, задание есть экземпляр рабочей нагрузки вместе с инструкциями на её выполнение, ссылками на необходимые данные и набором метаданных. Задание принимает на вход данные и выдаёт результат обработки. Соответственно, цель задания — полностью обработать входные данные. Как правило, входными и выходными данными являются коллекции файлов, однако могут использоваться и другие форматы, такие как: набор последовательных чисел, метаданные, коллекции чанков (sub-file data) или уведомления.

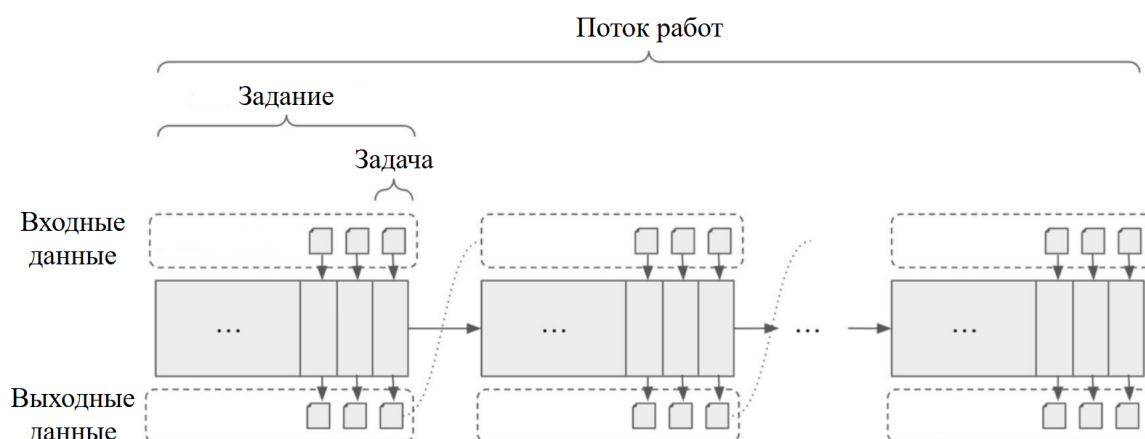


Рисунок 8 — Иерархия объектов в процессе обработки

Поток работ представляет собой верхнеуровневую единицу в иерархической организации обработки данных, объединяющую несколько рабочих нагрузок и описывающую их топологию. Каждый поток соответствует научной цели, разбитой на промежуточные шаги, необходимые для её достижения. **Рабочая нагрузка** — это программный модуль или приложение, использующее вычислительные ресурсы для выполнения одного из промежуточных шагов потока работ. Пользователь может запускать все шаги вместе или их часть в составе конкретного потока.

В рамках PanDA реализована возможность выполнения нагрузок в двух режимах:

1. Управляемая нагрузка, запускаемая централизованно на уровне виртуальной организации (VO). Это позволяет эффективно задействовать значительный объём вычислительных ресурсов, выделенных конкретной VO.
2. Пользовательская нагрузка, инициируемая отдельным исследователем, использующим свои квоты, но выполняющая задачи в интересах VO.

Один поток может включать как управляемые, так и пользовательские нагрузки. Общая структура потока приведена на рисунке 8, но важно отметить, что поток не обязательно должен быть линейным, он может реализовывать как параллельное выполнение, так и наложение времени выполнения конкретных заданий, что является дополнительным инструментом оптимизации процесса обработки. Минимальным же объектом в работе PanDA является, как можно видеть на том же рисунке 8, **задача** или **job** — объект с набором состояний, представляющий собой часть разбиения одного задания. Каждая задача получает на вход свою часть данных и по завершении формирует часть выходного результата. Сама задача выполняется на **работнике** - абстракции вычислительного ресурса, на физическом уровне реализованной в виде кластеров или рабочих узлов с разным набором процессоров, графических ускорителей, операционной памяти и средств хранения. Во многом при назначении работников на определенные задачи система старается реализовывать подход "Алгоритмы к данным", однако нередко приходится прибегать к куда менее эффективным с точки зрения передачи трафика стратегиям "Данные к алгоритмам" или вообще пересылать и то, и другое на третий ресурс ввиду политики глобального распределения ресурсов и приоритетов. **Глобальное распределение ресурсов** — механизм, определяющий, как совокупные вычислительные ресурсы делятся между различными группами VO и типами пользовательских заданий. Управляемые системой ресурсы динамически разбиваются на несколько глобальных доменов, которые назначаются каждому заданию в зависимости от запускающего его субъекта. Внутри системы многие компоненты оперируют именно этими доменами для планирования, приоритизации и учёта использования ресурсов в соответствии с заданными квотами и политиками. Говоря об упомянутой уже не раз приоритизации, скажем, что **приоритет** задания или задачи определяет, какой из конкурирующих процессов в рамках одного домена будет обслужен раньше. Как правило, задачи наследуют приоритет задания, которое их породило, однако первые несколько из них получают повышенный приоритет для оперативного сбора метрик, необходимых для дальнейшей обработки. Данный приоритет является универсальным, то есть распространяется на любое событие, связанное с процессом, будь то переходы состояния, обращение к внешним системам или непосредственное выполнение.

Описав объекты, которыми оперирует система, перейдем к описанию её управляющих компонентов.

асинхронную связь с пользователями, пилотами и сборщиками через HTTPS, а также набор планировщиков задач, выполняемых по расписанию. Сервер управляет задачами на всём их жизненном цикле. Основные функции сервера:

- приём задач, генерируемых JEDI, и задач из прочих источников, генерируемых, в основном, для тестирования
- обеспечение доступности входных данных на хранилищах, связанных с вычислительными ресурсами на которых запланировано выполнение задачи
- отправка задачи на рабочий узел после проверки доступности на нем входных данных
- отслеживание состояний задач на рабочих узлах
- передача метаданных чанков и отслеживание их обработки на соответствующем уровне разбиения, если задача сконфигурирована для подобной обработки
- выполнение пост-обработки после завершения задачи на узле
- отслеживание команд на ручное управление задачами
- отправка обратной связи по задачам, сгенерированных JEDI

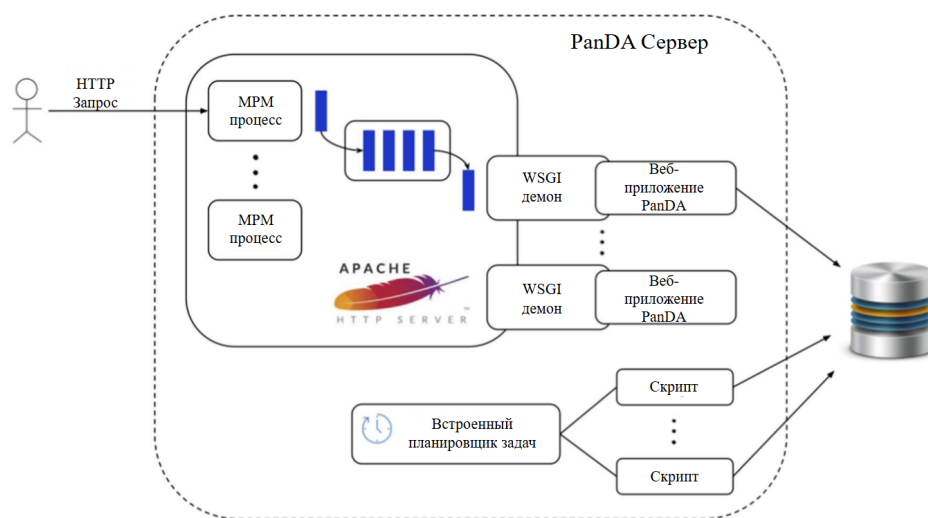


Рисунок 10 — Архитектура экземпляра PanDA-сервера

Благодаря архитектуре PanDA-сервера, приведённой на рисунке 10, его можно легко масштабировать за счёт добавления новых экземпляров. Архитектура включает Apache HTTP-сервер, который управляет WSGI-демонами и динамически масштабируемыми дочерними процессами для обработки запросов от пользователей и управляемых

агентов, таких как **пилоты** и **сборщики**. Поддерживаются синхронные и асинхронные запросы: первые блокируют отправителя до завершения, вторые — сразу возвращают ответ, а обработка продолжается в фоновом режиме. Встроенный планировщик задач заменяет cron, исключая наложения запусков, обеспечивая контроль за параллельным выполнением скриптов и синхронизацию между экземплярами сервера.

- **Пилот** (Pilot) — временный агент, выполняющий задачу на работнике и периодически отчитывающийся перед PanDA-сервером о различных метриках в течение всего своего жизненного цикла.
- **Сборщик** (Harvester) — компонент, генерирующий и отправляющий пилоты с использованием подходящего протокола для каждого поставщика ресурсов.
- **Система мониторинга PanDA** (BigPanDAmon) — веб-интерфейс для мониторинга задач и заданий в PanDA, предоставляющий единый портал для конечных пользователей, администраторов и технических специалистов. На портале приведены совокупные отчёты, дашборды и графические представления основных объектов системы (задания, задачи, площадки, пользователи). Интерфейс позволяет как детально исследовать причины отказов на узлах обработки, так и отслеживать общую картину — производительность площадок или ход масштабных кампаний по обработке. Сервис реализован на основе фреймворка Django с ядром из базовых представлений и набором подключаемых модулей. Внутренний ORM-слой Django позволяет сервису поддерживать Postgre и Oracle, используемые для получения и хранения промежуточных данных. Сами данные система аккумулирует из разных источников, среди них: база данных PanDA, CRIC, Rucio, Elasticsearch-кластер с логами PanDA/JEDI-приложений, MONIT Grafana с учётными и статистическими данными для ATLAS.

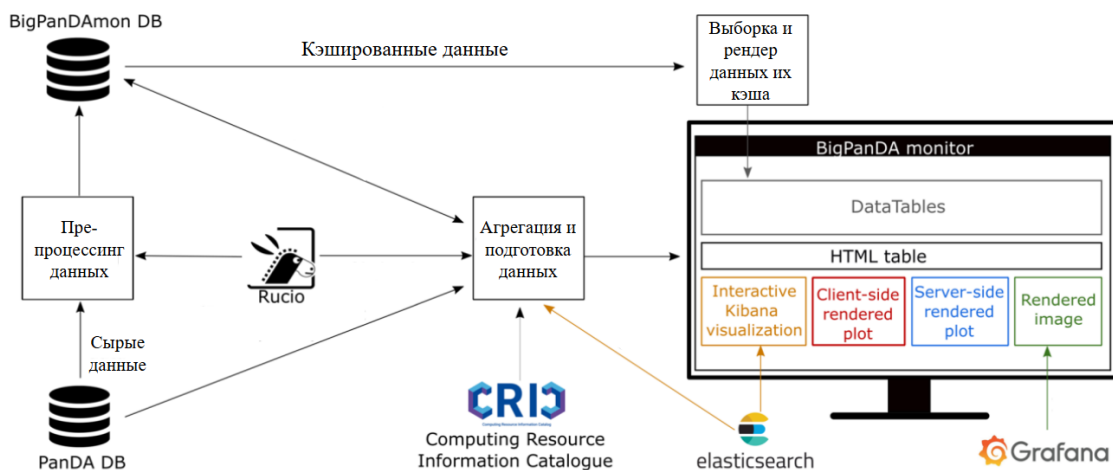


Рисунок 11 — Поток данных для системы мониторинга

Для обеспечения контролируемого доступа к данным используется аутентификация посредством протокола OAuth2 с SSO-провайдерами в лице ЦЕРНа, GitHub и Google. Также имплементированы механизмы кэширования и предварительной обработки: кэш на уровне БД хранит готовые к рендерингу данные, регулярные фоновые задачи подготавливают отчёты заранее, общее CephFS хранилище содержит логи и отрисованные графики Grafana, что позволяет улучшить производительность конечных запросов в момент обращения пользователей. Серверная отрисовка статического контента производится встроенными средствами Django, а динамика и интерактивность обеспечиваются благодаря использованию AngularJS. В общей сложности система включает более 70 различных представлений, доступных в виде интерактивных веб-страниц или JSON, в дополнение к 54 эндпоинтам для запросов дополнительных данных по требованию. Некоторые представления приведены ниже.

Cloud	GOC site name	Tier	Queue name	Status	Type	Workflow	System	Copytools	Associated Harvester	Min RSS [GB]	Max RSS [GB]	Max time [hours]	Max input size [GB]	Last comment
RU	JINR-LCG2	T2	JINR	online	production	pull_ups	---	rucio	CERN_central_A	0	16.4	72	---	no active blacklisting rules defined

Рисунок 12 — Статус центров обработки в РФ

Workflows attribute summary

status (5)

Finished (2068)

Failed (1)

Transforming (155)

SubFinished (9)

New (1)

username (6)

pandasv1 (2169)

James Smith (10)

cimcp (39)

Kevin Michael Nelson (10)

Aryan Borkar (3)

Peyan ShaoZho Hu (3)

Requests:

Show 10 entries

Search:

request id	username	workflow status	graph	workflow name	created on (UTC)	total tasks	tasks	transform type
1028345	pandasv1	New	plot	group.proj-evind.64fa5d6c-5da1-4b45-abe8-2dc48700bbae_0001	2025-05-20 15:13:05	0	New(0)	Other
1028343	pandasv1	Transforming	plot	data24_13p6TeV.00475035.physics_Main.daq.RAW	2025-05-20 14:25:40	1	Running(1)	Stageln
1028341	pandasv1	Transforming	plot	mc16_13TeV.302524.Pythia8EvtGen_A14NNPDF23LO_2DP20_Mass_400_650.simul.HITS.e4045_e6174_s3126_tid35253183_00	2025-05-20 14:25:35	1	Running(1)	Stageln
1028339	pandasv1	Transforming	plot	data24_13p6TeV.00475341.physics_Main.daq.RAW	2025-05-20 14:04:50	1	Running(1)	Stageln

Рисунок 13 — Общий мониторинг заданий

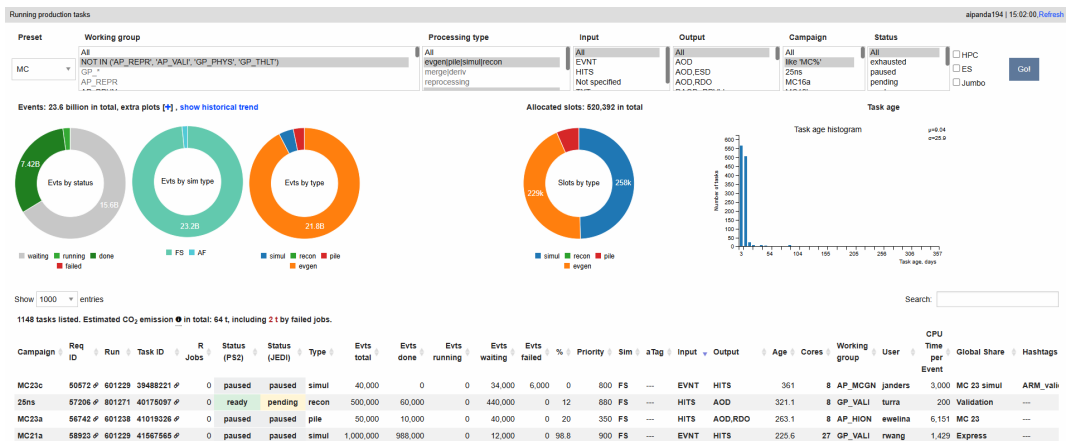


Рисунок 14 — Мониторинг выполняемых заданий

PanDA поддерживает выполнение множества типов обработки, перечень которых представлен на рисунке 15 и доступен посредством функционала, предоставляемого специализированными Python модулями и инструментами командной строки из библиотеки **PanDa Client**. Данная библиотека отвечает за обеспечение связи между пользовательским веб-интерфейсом и PanDA-сервером, что позволяет ускорить процесс создания заданий путем автоматизации ряда процессов и внедрения валидации. В случае, если пользователь хочет запустить задание вручную или использовать нестандартное для текущих процессов обработки прикладное ПО, то он может как использовать методы из модуля Client, так и инструменты командной строки, которые приведены ниже.

Основные компоненты и функции PanDa client включают:

- Отправка заданий; автоматизация процесса подготовки и запуска заданий на распределённых ресурсах [prun ...]

Monte-Carlo Production
Data Reprocessing
High Level Trigger
Tier-0 spill-over
SW Validation
Physics groups production
Derivation production in trains
Open-ended production
Users Analysis

Рисунок 15 — Типы заданий

- `-inDS`: Входной набор данных
 - `-outDS`: Выходной набор данных
 - `-exec`: Команда для выполнения
 - `-bexec`: Создание бинарных файлов
 - `-nFiles`: Количество файлов для обработки
 - `-writeInputToTxt`: Запись входных файлов в текстовый файл
 - `-outputs`: Спецификация выходных файлов
 - `-containerImage`: Образ контейнера для выполнения
 - `-architecture`: Запуск на GPU
 - `-match`: Фильтр входных данных
 - `-nJobs`: количество задач в задании
 - `-nEventsPerJob`: количество событий в задаче
 - ...
- Интеграции с Athena [`pathena ...`]
 - `-inDS`: Входной набор данных
 - `-outDS`: Выходной набор данных
 - `-trf`: Трансформационный скрипт
 - `-containerImage`: Образ контейнера для выполнения
 - `-nJobs`: количество задач в задании
 - `-nEventsPerJob`: количество событий в задаче
 - ...
- Интерфейс для управления заданиями и мониторинга их статуса [`rbook ...`]
 - `kill`
 - `finish`
 - `retry`
 - `show`
 - `showl`
 - ...
- Настройка параметров отправки заданий и управления ресурсами

- взаимодействие с вычислительными центрами стала осуществлять посредством сервиса PanDA, с использованием x509-сертификатов для авторизации, протокола передачи данных XRootD и сервиса FTS
- мониторинг заданий и задач в созданном интерфейсе

Так же в 2022 году был осуществлён переход на новую систему хранения СТА [19], заменившую CASTOR, однако с точки зрения объектов хранения изменений не произошло, база данных системы все ещё хранит историю выполнения задач и является каталогом обработанных данных с соглашением об именовании: год/период/название обработки/номер сброса детектора-номер файла-параметры обработки, пример приведён на рисунке 17.

`/data/2016/oracle_dst/P11/slot2/mDST/mDST-276384-2-7.root.001`

Рисунок 17 — Структура имен заданий и датасетов

На рисунке 18 можно видеть форму создания и таблицу отслеживания, а на рисунке 19 - интерфейс мониторинга состояний текущих заданий.

The screenshot displays the COMPASS interface. On the left is a form for creating a new job. The 'Name' field is '2016P09S5.1_mu-'. The 'Type' dropdown is set to 'test production'. The 'Site' is '/cmfss/compass.cern.ch/'. The 'Path' is 'data/2016/'. The 'Soft' is 'P09/slot5.1'. The 'Production' is '2016P09-5.1'. The 'Year' is '2016'. The 'Period' is 'P09'. The 'Prodsit' is '51'. The 'Phastver' is '8'. The 'Template' is 'template_mu-opt'. The 'Files source' is 'runs list'. On the right is a table of jobs. The table has columns: RUN NUMBER, CHUNK NUMBER, PANDA ID, ATTEMPT, and STATUS. The table contains 10 rows of job data, all with a status of 'finished'.

RUN NUMBER	CHUNK NUMBER	PANDA ID	ATTEMPT	STATUS	
2016/raw/W14/cdr11091-	275678	11091	2182400	1	finished
2016/raw/W14/cdr11082-	275678	11082	2182399	1	finished
2016/raw/W14/cdr11080-	275678	11080	2182398	1	finished
2016/raw/W14/cdr11089-	275678	11089	2182397	1	finished
2016/raw/W14/cdr11086-	275678	11086	2182396	1	finished
2016/raw/W14/cdr11063-	275678	11063	2182395	1	finished
2016/raw/W14/cdr11049-	275678	11049	2182394	1	finished
2016/raw/W14/cdr11016-	275678	11016	2182393	1	finished
2016/raw/W14/cdr11094-	275678	11094	2182392	1	finished
2016/raw/W14/cdr11092-	275678	11092	2182391	1	finished
2016/raw/W14/cdr11088-	275678	11088	2182390	1	finished
2016/raw/W14/cdr11076-	275678	11076	2182389	1	finished

Рисунок 18 — Панель для формирования заданий COMPASS

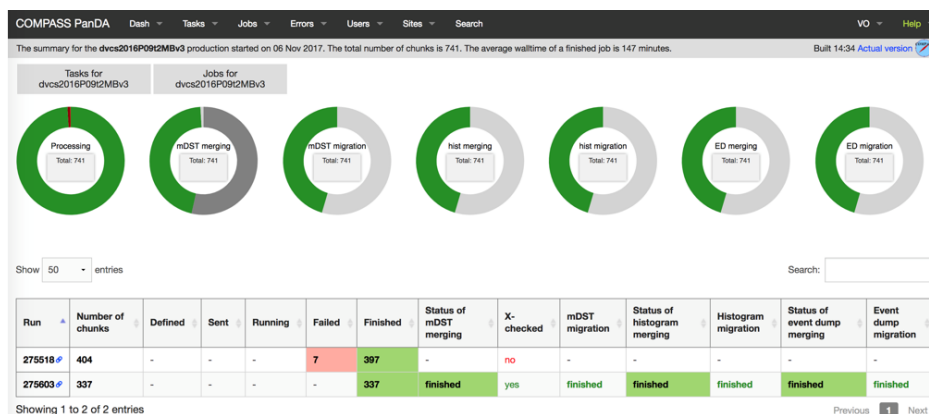


Рисунок 19 — Представление статистики

3 Анализ технологий для построения системы

Проанализировав поставленные перед системой требования, обусловленные областью её применения, а также рассмотрев аналогичные системы, которые успели зарекомендовать себя в рамках других масштабных физических экспериментов, была выработана следующая схема составляющих (рисунок 20).

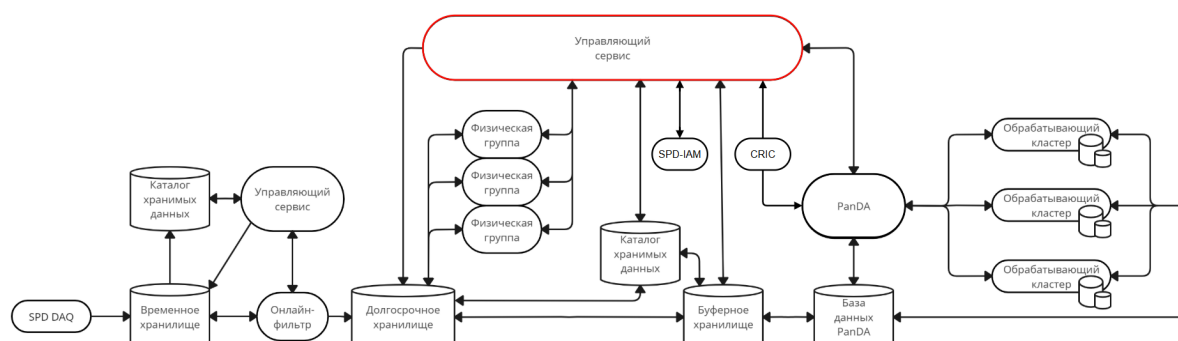


Рисунок 20 — Схема IT-составляющих

3.1 Выбор фреймворка для приложения

В ходе работы рассматривались два фреймворка, для реализации веб-приложения. Наиболее популярными в данной сфере решениями являются Django и Angular, ниже приведены их ключевые особенности.

Django

- MVT-архитектура (Model View Template)

- разделение обязанностей
- повторное использование кода
- масштабируемость
- ORM (Object Relational Mapper)
 - оптимизация часто используемых операций с базой данных
 - экранирование параметров запросов
 - автоматическая обработка подключений к базе данных и управление соединениями
 - поддержка миграций
- Реализованный admin-интерфейс
- Встроенные функции обеспечения безопасности
 - аутентификация и управление сессиями пользователей
 - защита от CSRF (Cross-site Request Forgery)
 - автоматическая обработка подключений к базе данных и управление соединениями
 - защита от XSS (Cross-site Scripting)
 - защита от SQL-инъекций
 - управление паролями
 - ограничение доступа на уровне представлений
 - HTTPS (HyperText Transfer Protocol Secure)

Angular

- MVT-архитектура (Model View Template)
 - разделение обязанностей
 - повторное использование кода
 - масштабируемость
- SPA (Single Page Application)
 - маршрутизация и навигация между компонентами
 - поддержка Lazy Loading для оптимизации загрузки приложений
- Dependency Injection

- HTTP клиент и обработка запросов
 - встроенный HTTP клиент для выполнения запросов к серверу
 - интерцепторы для обработки HTTP запросов и ответов
- Формы и валидация
 - модуль `ReactiveForms` для создания и валидации форм
 - встроенные механизмы валидации данных на стороне клиента
- Тестирование
 - модульное тестирования компонентов и сервисов
 - инструменты для автоматизации тестирования
- Встроенные функции обеспечения безопасности
 - защита от CSRF (Cross-site Request Forgery)
 - защита от XSS (Cross-site Scripting)
 - HTTPS (HyperText Transfer Protocol Secure)

Из приведенных выше особенностей видится, что для нашей задачи разумнее будет использовать фреймворк Django, ввиду более широких возможностей обеспечения безопасности, встроенного функционала для работы с базами данных и инструментов аутентификации.

Однако в случае необходимости внедрения широких возможностей для взаимодействия с пользователями в будущем можно рассмотреть совместное использование фреймворков для обеспечения наилучшей реализации веб-приложения: Django для серверной части и Angular для клиентской. Это позволит разделить логику приложения на независимые части, что облегчит разработку, тестирование и поддержку, а также позволит использовать все преимущества обоих фреймворков.

3.2 OAuth 2.0

3.3 JWT

3.4 Apache2

4 Разработка прототипа веб-приложения

В ходе работы был изучен фреймворк Django, с использованием которого реализовано веб-приложение для создания, управления и отслеживания заданий на обработку.

Средствами ORM-слоя реализованы модели заданий (общая, Simulation, Reconstruction), формы для создания соответствующих заданий и их цепочек, что позволяет собирать необходимые параметры, формировать из этих параметров словарь и отправлять его в очередь PanDA для дальнейшей обработки.

The image shows two side-by-side screenshots of a web application interface for creating tasks. Both forms are titled "Task Creation".

Left Form (Simulation):

- Task name:
- Output dataset name: [Naming convention here](#)
- Total events:
- Events per job:
- Cloud:
- Data disk:
- Skip scout: ☐
- Offset:
- Path to execution files: smth like -> /cvmfs/spd.jinr.ru/production/MC/minbias-P8-sdroot417-dev.10GeV.V01
- Path to container: smth like -> /cvmfs/spd.jinr.ru/images/spdroot-dev-4.1.7.sif
-

Right Form (Reconstruction):

- Task name:
- Input dataset name: [Naming convention here](#), note that no extension expected
- Output dataset name: [Naming convention here](#)
- Files per job:
- Cloud:
- Data disk:
- Skip scout: ☐
- Offset:
- Path to execution files: smth like -> /cvmfs/spd.jinr.ru/production/MC/minbias-P8-sdroot417-dev.10GeV.V01
- Path to container: smth like -> /cvmfs/spd.jinr.ru/images/spdroot-dev-4.1.7.sif
-

Рисунок 21 — Создание заданий Simulation(слева), Reconstruction(справа)

Внедрены средства просмотра всех заданий (рисунок 22), завершённых заданий (рисунок 24). Реализованы возможности сортировки и фильтрации, а для завершённых - дополнительно существует возможность загрузки данных для отчётности. Для представленных таблиц реализованы модели на стороне сервера и представления на стороне базы данных.

Select field▼Enter valueFilter										
Task ID	Task name ↑ ↓	Parent ID	Creator	Status	Done jobs	Default/Current priority	Total events	Submit time ↑ ↓	Start time ↑ ↓	End time ↑ ↓
370	PROD2025-017.SIM	370	Elena Zemlyanichkina	running	2	900/900	0	17:03, 20 May 2025	17:11, 20 May 2025	None
369	PROD2025-016.RECO	369	Elena Zemlyanichkina	finished	9999	900/900	0	03:31, 10 May 2025	08:33, 10 May 2025	22:12, 13 May 2025
368	PROD2025-016.SIM	368	Elena Zemlyanichkina	done	10000	900/900	0	15:08, 08 May 2025	15:16, 08 May 2025	22:07, 09 May 2025
367	PROD2025-015.RECO	367	Elena Zemlyanichkina	finished	9992	900/900	0	13:16, 07 May 2025	13:24, 07 May 2025	18:58, 09 May 2025
366	PROD2025-015.SIM	366	Elena Zemlyanichkina	done	10000	900/900	0	04:26, 06 May 2025	07:01, 06 May 2025	12:52, 07 May 2025
365	PROD2025-014.RECO	365	Elena Zemlyanichkina	finished	9991	900/900	0	23:13, 04 May 2025	23:21, 04 May 2025	20:37, 06 May 2025
364	PROD2025-014.SIM	364	Elena Zemlyanichkina	done	10000	900/900	0	14:21, 03 May 2025	14:28, 03 May 2025	21:35, 04 May 2025
363	PROD2025-013.RECO	363	Elena Zemlyanichkina	finished	9992	900/900	0	10:03, 02 May 2025	10:11, 02 May 2025	00:51, 04 May 2025
362	PROD2025-013.SIM.2	362	Elena Zemlyanichkina	done	10000	900/900	0	23:35, 30 Apr 2025	23:43, 30 Apr 2025	08:57, 02 May 2025
361	PROD2025-013.SIM	361	Elena Zemlyanichkina	aborted	869	900/900	0	10:59, 30 Apr 2025	11:10, 30 Apr 2025	15:13, 30 Apr 2025
Page: 1 2 3 ... 27 28 29										

Рисунок 22 — Задания, отсортированные по убыванию Task Id

Select field▼Enter valueFilter										
Task ID	Task name ↑ ↓	Status	Start date	End date	Walltime	Total events	Events per job	Total jobs	Out DS size, GB	Out Log size, GB
369	PROD2025-016.RECO	finished	10 May 2025	13 May 2025	30	None	None	9999	18557.43	5.07
368	PROD2025-016.SIM	done	08 May 2025	09 May 2025	22931	40000000	4000	10000	18365.39	1.92
367	PROD2025-015.RECO	finished	07 May 2025	09 May 2025	23	None	None	9992	18543.30	5.05
366	PROD2025-015.SIM	done	06 May 2025	07 May 2025	24486	40000000	4000	10000	18362.72	1.97
365	PROD2025-014.RECO	finished	04 May 2025	06 May 2025	24	None	None	9991	18540.20	5.05
364	PROD2025-014.SIM	done	03 May 2025	04 May 2025	24612	40000000	4000	10000	18358.86	1.96
363	PROD2025-013.RECO	finished	02 May 2025	04 May 2025	20	None	None	9992	18536.20	5.06
362	PROD2025-013.SIM.2	done	30 Apr 2025	02 May 2025	24899	40000000	4000	10000	18357.95	1.93
359	PROD2025-012.RECO	finished	28 Apr 2025	29 Apr 2025	24	None	None	9993	18546.74	5.08
358	PROD2025-012.SIM	done	25 Apr 2025	26 Apr 2025	23316	40000000	4000	10000	18360.80	1.89
Page: 1 2 3 4 5 6 7										
Export csv										

Рисунок 23 — Завершенные задания

Возможен просмотр данных о конкретном задании (рисунок ??), его предварительное завершение или отмена.

Finish taskKill task

Task ID	370
Parent task ID	370
Chain task ID	None
Name	PROD2025-017.SIM
Type	managed
VO	spd.nica.jinr
Creator	Elena Zemlyanichkina
State status	running
Total events	0
Total jobs done	2
Total requested jobs	5003
Priority	900
Current priority	900
Submit time	17:03 May 20, 2025
Start Time	17:11 May 20, 2025
Time stamp	19:25 May 20, 2025
End Time	
JEDI parameters	[{"taskName": "PROD2025-017.SIM", "nEvents": 20000000, "nEventsPerJob": 4000, "userName": "Elena Zemlyanichkina", "vo": "spd.nica.jinr", "taskPriority": 900, "architecture": "x86_64", "cloud": "JINR", "workingGroup": "spd.nica.jinr", "skipScout": true, "transUses": "A", "transHome": null, "transPath": "http://159.93.221.125:8080/spd_simu_VA_transform.sh", "processingType": "step1", "prodSourceLabel": "managed", "taskType": "test", "ramCount": 1900, "noInput": true, "log": [{"dataset": "MC2025_S1.minbias-P8-spdroot4172-dev.27GeV-UU.PROD2025-017.SIM.1.log", "type": "template", "param_type": "log", "token": "SPDDATADISK", "offset": 0, "value": "MC2025_S1.minbias-P8-spdroot4172-dev.27GeV-UU.PROD2025-017.SIM.1.\${SN}.log.tgz"}, {"jobParameters": [{"type": "constant", "value": "singularity run --bind /cvmfs/spd.jinr.ru/production/MC/minbias-P8-spdroot4172-dev.27GeV.V02_D0sig:/prod -H ./WORKDIR /cvmfs/spd.jinr.ru/images/spdroot-dev-4.1.7.2.sif spdroot.py -b -q /prod/simu.C(4000, *)", {"type": "constant", "value": ""}, {"type": "template", "param_type": "output", "token": "SPDDATADISK", "value": "s.MC2025_S1.minbias-P8-spdroot4172-dev.27GeV-UU.PROD2025-017.SIM.1.\${SN}/P", "dataset": "MC2025_S1.minbias-P8-spdroot4172-dev.27GeV-UU.PROD2025-017.SIM.1.S", "offset": 0}, {"type": "constant", "value": ""}, {"type": "template", "param_type": "output", "token": "SPDDATADISK", "value": "p.MC2025_S1.minbias-P8-spdroot4172-dev.27GeV-UU.PROD2025-017.SIM.1.\${SN}/P", "dataset": "MC2025_S1.minbias-P8-spdroot4172-dev.27GeV-UU.PROD2025-017.SIM.1.P", "offset": 0}, {"type": "constant", "value": ""}, {"type": "template", "value": "\${RNDMSEED}", "param_type": "number", "offset": 0}, {"type": "constant", "value": ""}]]}]

Рисунок 24 — Информация о задании

5 Создание веб-сервера

Создание веб-сервера осуществлено посредством серверного ПО Apache2, связанного с проектом через `mod_wsgi`.

Apache2 используется для обработки HTTP-, FTP-, SMTP-запросов и отправки ответов на них пользователю. Основан на модульной архитектуре, что ведет к гибкости и масштабируемости. Архитектура Apache2 включает в себя ядро сервера, модули, конфигурационные файлы и виртуальные хосты.

Создан виртуальный хост для обработки запросов, отредактированы некоторые файлы конфигураций для их обработки, например конфигурационный файл `"apache2/sites-available/***.conf"` для связывания Django-приложения с Apache через `wsgi`. Внесли соответствующие правки в `wsgi.py` и `settings.py`, добавив каталог `apache2` в PATH приложения и прописав `ip-адрес` соответственно.

Создана виртуальная машина на AlmaLinux OS 9, после настройки веб-сервер будет функционировать на ней. Контроль версий будет осуществляться посредством git-репозитория ОИЯИ `git.jinr.ru`.

6 Механизм аутентификации и авторизации

В целях обеспечения сквозной аутентификации и, как следствие, беспшовного использования сервиса в рамках общей системы была произведена интеграция с SSO-провайдером SPD-IAM (spd-iam.jinr.ru). Взаимодействие с сервисом авторизации происходит по протоколу OAuth 2.0 со сценарием Authorization Code Flow; все сообщения передаются через HTTPS-соединение. При передаче конфиденциальная информация содержится только в теле запроса, а на хранении - в переменных окружения или локальном сессионном хранилище, что обеспечивает защиту функционала сервиса от несанкционированного использования. Получаемый при этом соединении JSON-объект содержит самодостаточный набор метаданных и нагрузки в виде JWT, позволяющий как валидировать неподдельность пакета, так и управлять доступом к защищенным частям сервиса.

Часть полезной нагрузки помещается в сессионное хранилище для обеспечения работоспособности системы, а именно:

- `userName`
- `logged_in_timestamp | expiration_timestamp` - для контроля времени предоставления доступа
- `production_member` - для контроля возможности создания заданий
- `id_token` - предоставляется стороне PanDA WMS для проверки доступа

На текущий момент матрица прав доступа(таблица 5) достаточно простая, однако она полностью удовлетворяет текущим требованиям, а в случае необходимости, её легко модифицировать посредством изменения реализованного промежуточного ПО и атрибутов сессии. Упомянутое промежуточное ПО отвечает за контроль доступа к внутреннему функционалу сервиса, проверку срока действия токена доступа и его обновление при условии валидности рефреш-токена.



Welcome to **SPD**

Sign in with your SPD credentials

Sign in

[Forgot your password?](#)

Or sign in with

JINR Keycloak

Your institutional account

Not a member?

Apply for an account

Рисунок 25 — Аутентификация через SPD-IAM

Роль \ Действие	Роль		
	Пользователь	Участник группы Production	Администратор
Просмотр заданий	✓	✓	✓
Создание заданий	×	✓	✓
Изменение состояния заданий	×	Только если он создатель задания	Только если он создатель задания
Просмотр конфигурации сервера	×	×	✓

Таблица 5 — Матрица прав доступа

Заключение

На данном этапе научной работы:

- изучены соответствующие системы, реализованные в рамках других мега-сайнс экспериментов. Анализ показал, что в эксперименте ATLAS более сложные процессы обработки и другое прикладное ПО в виде Athena; в COMPASS - нет системы управления данными, ввиду чего большое количество дополнительного задач выполнялось системой, аналогичной разрабатываемой. Это привело нас к созданию собственной системы управления процессами обработки, но с учётом преимуществ обоих рассмотренных экспериментов
- изучены технологии, необходимые для реализации сервиса
- произведено полное развертывание и настройка приложения в рамках облачного сервиса ОИЯИ
- имплементированы механизмы коммуникации с прочими сервисами в рамках эксперимента
- внедрена возможность создания/мониторинга/завершения и отмены заданий
- реализована сквозная аутентификация на JWT

Список литературы

- [1] Технический проект комплекса nica. https://nica.jinr.ru/documents/TDR_spec_Fin0_for_site_eng.pdf/. Дата последнего обращения: 04.08.2024.
- [2] NICA Collaboration. Searching for a qcd mixed phase at the nuclotron-based ion collider facility. White paper, Joint Institute for Nuclear Research (JINR), 2010.
- [3] Эффекты поляризации в Дрелл-Яновских процессах. http://www1.jinr.ru/Репан/2010-v41/v-41-1/04_sis.pdf. Дата последнего обращения: 04.08.2024.
- [4] Takumi Iritani, Guido Cossu, and Shoji Hashimoto. Partial restoration of chiral symmetry inside hadrons. *PoS, LATTICE2014*:338, 2014.
- [5] Takumi Iritani, Guido Cossu, and Shoji Hashimoto. Lattice QCD study of partial restoration of chiral symmetry in the flux-tube. *PoS, Hadron2013*:159, 2013.
- [6] Wei Kou and Xurong Chen. Exploring quantum entanglement in chiral symmetry partial restoration with 1+1 string model. *Phys. Lett. B*, 853:138675, 2024.
- [7] В.П. Овсянников. *Техника и применение электронно-лучевых ионных источников*. дис. ... д-ра техн. наук, 1996. РГБ ОД, 9 96-2/1954-2.
- [8] V. P. Ladygin. Spin Physics Detector at NICA. *JPS Conf. Proc.*, 37:011012, 2022.
- [9] Alexey Guskov. Spin Physics Detector project at JINR. *PoS, PANIC2021*:344, 2022.
- [10] V. Abazov et al. Technical Design Report of the Spin Physics Detector at NICA. *Natural Sci. Rev.*, 1:1, 2024.
- [11] G. Aad et al. The ATLAS Experiment at the CERN Large Hadron Collider. *JINST*, 3:S08003, 2008.
- [12] 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(6):062005, 2015.

- [13] Vincent Garonne, R. Vigne, G. Stewart, M. Barisits, T. Beermann, M. Lassnig, C. Serfon, L. Goossens, and A. Nairz. Rucio - The next generation of large scale distributed system for ATLAS Data Management. *J. Phys. Conf. Ser.*, 513:042021, 2014.
- [14] Tadashi Maeno. PanDA: Distributed production and distributed analysis system for ATLAS. *J. Phys. Conf. Ser.*, 119:062036, 2008.
- [15] Alexey Anisenkov, Julia Andreeva, Alessandro Di Girolamo, Panos Paparrigopoulos, and Boris Vasilev. CRIC: Computing Resource Information Catalogue as a unified topology system for a large scale, heterogeneous and dynamic computing infrastructure. *EPJ Web Conf.*, 245:03032, 2020.
- [16] Martin Barisits et al. Rucio - Scientific data management. *Comput. Softw. Big Sci.*, 3(1):11, 2019.
- [17] COMPASS (Common Muon and Proton Apparatus for Structure and Spectroscopy). <https://home.cern/science/experiments/compass>.
- [18] Петросян А.Ш. *Методика и программная инфраструктура глобально распределенной обработки данных эксперимента COMPASS*. дис. ... канд. тех. наук, Объединённый институт ядерных исследований, Дубна, 2008. Режим доступа: https://issc.jinr.ru/files/59/f_2_1466.pdf.
- [19] CTA (CERN Tape Archive). <https://cta.web.cern.ch/cta/>. Дата последнего обращения: 04.08.2024.
- [20] J. Iven, M. Lamanna, and A. Pace. CERN's AFS replacement project. *J. Phys. Conf. Ser.*, 898(6):062040, 2017.
- [21] CASTOR (CERN Advanced STORage). <https://castor.web.cern.ch/castor/>.
- [22] IBM LSF. https://www.ibm.com/support/knowledgecenter/SSETD4/product_welcome_platform_lsf.html.
- [23] HTCondor. <https://htcondor.org/>.
- [24] A. Sh. Petrosyan. Panda for compass at jinr. *Physics of Particles and Nuclei Letters*, 13(5):708–710, Sep 2016.
- [25] Artem Petrosyan. Compass production system overview. *EPJ Web of Conferences*, 214:03039–03039, Jan 2019.