# Информационные системы в экспериментах в ФЧВЭ

# Физические данные: События

- Частицы, ускоряемые коллайдером , группируются в "сгустки" (bunch), которые пересекаются друг с другом каждые с частотой 40 МГц (ATLAS) или 12.5 Мгц (SPD).
- Детектор находится в одной из областей пересечения
- Во время каждого столкновения пучков происходит несколько независимых взаимодействий, практически одновременно (в течении порядка наносекунды)
- Сигналы в детекторе от этих взаимодействий записываются вместе как одно “*событие*”
  - События могут определятся при помощи специальной системы (trigger) срабатывающей при обнаружении «интересных» комбинаций сигналов с детекторов. Такая схема реализована на установке ATLAS
  - На установке SPD события строятся на основании данных полученных детекторов в течении некоторого промежутка времени, с последующим отбором в “онлайн-фильтре”
- В одно событие могут попасть результаты нескольких взаимодействий (pile-up)
  - Для ATLAS — до 50-60, для SPD — не актуально

# Физические данные: Обработка

- **Изначально записи «сырых» (*RAW*) событий содержат:**
  - **Сигналы с детекторов**
  - **Триггерные решения либо результаты online фильтра**
- **Затем, в результате offline-обработки восстанавливаются физичесие параметры события:**
  - **частицы и их параметры, полная и недостающая энергии, поляризация, etc …**
- **Восстановленые события (AOD) используются для проведения физических анализов**
  - **В эксперименте ATLAS также создаются производные наборы событий под разные группы анализов**
- **События могут быть повторно обработаны**
  - **когда становятся доступными улучшенные алгоритмы восстановления или константы калибровки и выравнивания детектора**
- **Помимо “реальных” событий, методом Монте-Карло генерируются их аналоги (MC Events)**
  - **На ATLAS их ~ в три раза больше чем реальных**

# Физические данные: Организация

- **Число событий (ATLAS):**
  - Run2: (2015-2018): 10 млрд реальных событий и 35 млрд MC в год
  - Run3: (2022-?): 35 млрд реальных + 100 млрд MC /год
- **Число событий (SPD):**
  - Phase I (2028-2030): 100 млрд реальных событий в год + сравнимое число данных Монте-Карло
  - Phase II (2032-?): 1.5 трлн реальных событий в год + MC
- Размер события SPD: ~ 15 кб, на 2 порядка меньше чем на ATLAS

---

- Группы статистически эквивалентных событий хранятся в файлах на диске или на магнитной ленте.
- Каждый файл обычно содержит от 1000 до 10000 событий, в зависимости от формата
- На 2021 г. ATLAS хранил более 100 миллионов файлов на диске, чуть более 200 PB, + копии на магнитной ленте.
- SPD планирует набирать ~ 10 PB данных в год

# Физические данные: Хранение

- Файлы группируются в датасеты (наборы данных)
  - Датасет как правило содержит события относящиеся к одному Run (сеансу непрерывного набора данных)
- Каждое событие в датасете имеет уникальный идентификатор определяемый номером Run и номером события в Run
- Данные МС организованы сходным образом
- Датасеты хранятся и обрабатываются на серверах распределенной вычичлительной системы
  - ATLAS использует распределенную вычислительную инфраструктуру под названием WLCG grid
    - Сеть WLCG включает в себя более 170 компьютерных центров, предоставляющих вычислительные ресурсы, дисковое и ленточное хранилище для экспериментов на БАК
    - Помимо научных центров могут использоваться коммерческие сервера, облачные ресурсы, и LHC @ Home
  - Для SPD также предполагается использовать инфраструктуру Российской части WLCG и ресурсы участников

# ИС распределенных данных

- **Computing Resource Information Catalog (CIRC)**
  - собирает информацию обо всех вычислительных ресурсах и хранилищах данных, протоколах доступа, точках входа, etc...
  - распространяет эту информацию через API
  - эволюционировал из ИС ATLAS Grid (AGIS)

- **Система управления данными Rucio***
  - Каталог содержимого датасетов:
    - Список файлов, суммарный размер, принадлежность, происхождениен, время жизни, состояние ...
  - Каталог файлов:
    - Размер, контрольная сумма, число событий ...
  - Каталог расположения датасетов:
    - Список копий датасетов на узлах сети Grid
  - Средства пересылки данных:
    - Очередь пересылаемых датасетов, состояние
  - Средства для удаления данных:
    - Список датасетов (или копий) для удалены, состояние
  - Списки и состояние ресурсов хранения (на узлах GRID)

* Название системы происходит от имени осла Санчо Пансы, персонажа “Дон Кохота” Сервантеса. Прототип назывался DQ2 (Don Quijote 2)

# ИС распределенных вычислений

- FTS / DTS: Обеспечивает массовую передачу данных

- ProdSys/JEDI/PanDA
  (Управление распределенными вычислениями)
  - Список запрошенных задач (tasks) а также датасеты на входе и выходе этих задач, версии программного обеспечения, …
  - Список заданий (jobs), их состояние и размещение, …
  - Списки ресурсов обработки, состояние …
  - Получение информации и управление задачами возможно через Web интерфейс, клиент в командной строке или API для использования в автоматических системах.

- ATLAS запускает более 1 миллиона заданий в день, используя 200 тыс. слотов
- В среднем в день 600 ТБ данных пересылаются между вычислительными центрами

Системы управления распределенными вычислениями и данными адаптируются для использования на SPD

# Физические метаданные / AMI

*Метаданные* – данные о данных

- Информация о датасетах, цепочках происхождения обработанных данных и ссылки на конфигурации задач
- Использованные при моделировании конфигурации и сечения
- Информация о триггерах, светимостях для реальных и моделированных данных

AMI (ATLAS Metadata Interface) информация по датасетам

- Позволяет находить подходящие данные для анализа и получать массу дополнительной информации
  - Количество и размеры файлов, число событий
  - Периоды и ссылки к условиям набора данных
- Информация берется из различных источников:
  - *Tier 0* — компьютерная ферма на которой производится первичная обработка данных с установки
  - Система управления и мониторинга распределенной вычислительной сети Grid (ProdSys/PanDA)
  - Rucio — система управления данными на узлах сети Grid

Информация доступна через Web интерфейс, клиент в командной строке или python API

# Данные состояний - Condition Data

- **Condition Data** – **информация о установке и параметрах набора данных, не входящая в конкретное событие**
  - **Параметры оборудования детектора**
    - **Температуры, токи, напряжения, давления газов etc..**
  - **Параметры сбора данных**
    - **Конфигурации триггера и съема данных детекторов**
  - **Калибровки детектора**
    - **Калибровка по энергии, временные пороги …**
  - **Позиционирование установки**
    - **Относительное и абсолютное позиционирование подсистем**
  - **Физические калибровки**
    - **Шкалы энергий и разрешения, веса, эффективности триггера**
- **Характеризуют состояние в течение определенных интервалов времени - периодов валидности (IOV)**
- **Используется для калибровки подсистем, в on-line обработке, при реконструкции событий и повторной обработке, а также при анализе данных**

# База данных состояний CREST

- **База данных состояний CREST - ИС с поддержкой интервалов валидности и версий для всех типов данных условий**
  - **Архитектура CREST разработана как модель клиент-сервер, с реляционной базой данных в качестве бэк-энда**
  - **Доступ к данным был реализован с помощью чистого REST API с поддержкой JSON.**
  - **Библиотека клиентского доступа C++ предоставляет интерфейс для HTTP-запросов**
  - **Данные об условиях обычно записываются один раз и часто считываются.**
    - **Например, одновременно может выполнятся ~ 10k задач реконструкции данных для каждой нужны данные о состоянии.**
  - **Кэширование результатов запросов с помощью SQUID прокси позволяет существенно снизить нагрузку на БД**
- **Разработана для ATLAS (и других экспериментов на БАК)**
- **Предполагается адаптация для экспериментов на NICA**

# Базы данных оборудования

- Каталог оборудования и компонент из которых состоит детектор
- Содержит информацию о детекторах и электронных компонентах, кабелях, стойках и крейтах а также историю их положения на установке
- Необходима для сборки, запуска и поддержания в рабочем состоянии систем установки
- Особенно полезна для передачи знаний между членами команды
- Тесно связана с ИС схем подключения, позволяющей отслеживать прохождение потоков данных и сигналов в установке и управлять ими
- Ведется разработка прототипов этих ИС для установки SPD

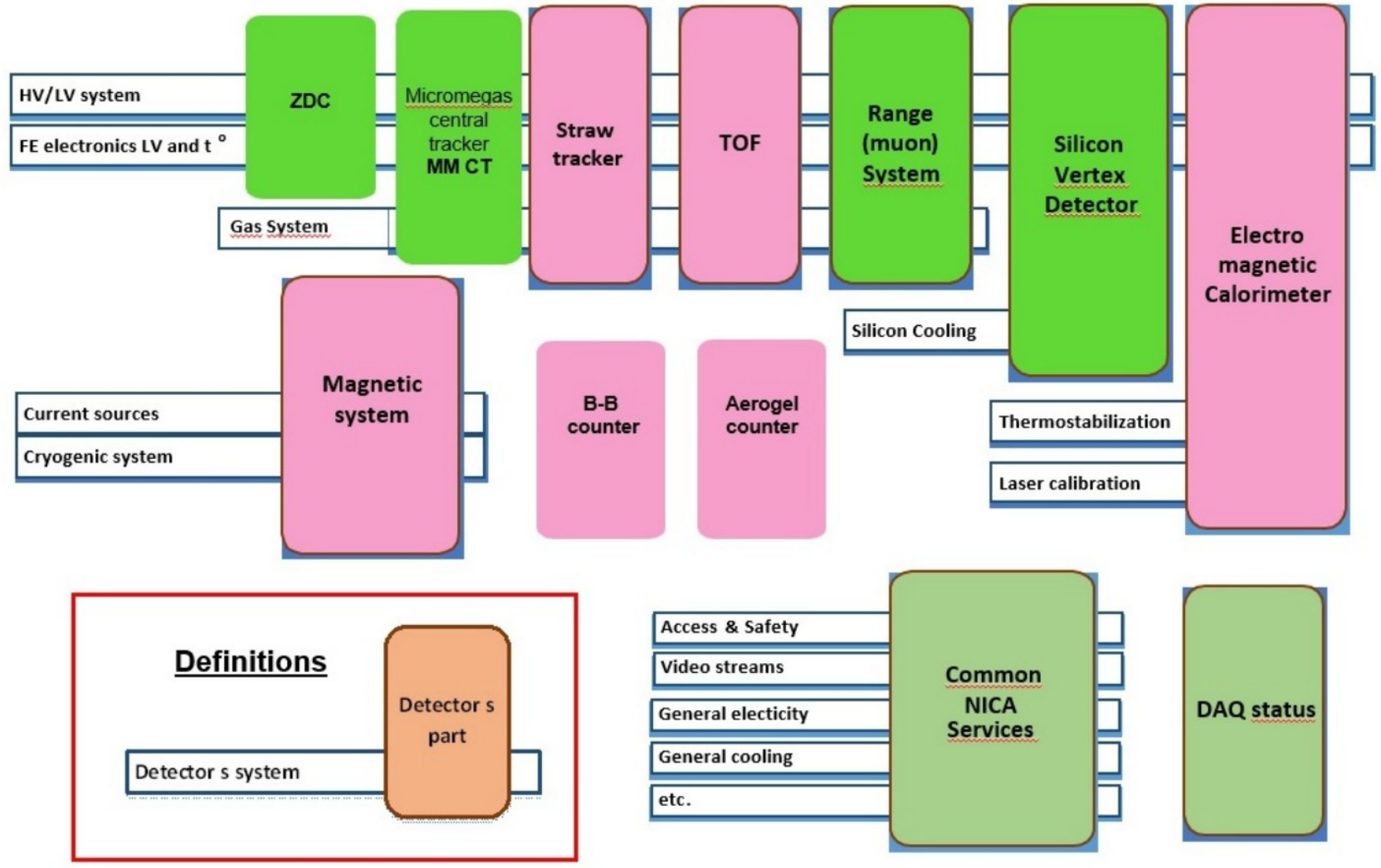
ATLAS EXPERIMENT

Cables: **83529**
Crates: **7714**
Equipment: **167491**
Racks: **929**
Boards: **11534**
Channels: **26585**
Cable Trays: **1360**
Zones: **21**
Detector Parts: **1618**
Equipment Models: **1984**
Measurements: **10413**

# БД системы управления установкой

- **Задача системы контроля детектора (DCS) - согласованное и безопасное управление и мониторинг установки, его поддетекторов и технической инфраструктуры.**
- **DCS должна переводить детектор в любое требуемое рабочее состояние, постоянно контролировать и архивировать рабочие параметры, сигнализировать о любом ненормальном поведении.**
- **DCS должна служить единым интерфейсом для всех поддетекторов и технической инфраструктуры эксперимента**
- **DCS должна поддерживать связь с другими системами, которые управляются независимо, такими как КС ускорителя и технические службы.**

# БД системы управления установкой




- **Масштабы для некоторых систем**
- **DSSD (vertex): 1.5 млн каналов  Range System: 200 000 каналов  ECAL: 60 000 каналов**

# БД персонала и публикаций

- **В больших проектах как правило участвуют сотни и тысячи людей**
  - **В ATLAS ~ 6000 участников (~3000 авторов), в SPD ~ 400**
- **Для организации эффективного взаимодействия с совместным использованием компьютерных и других ресурсов нужна информационная система реализующая**
  - **Управление данными персонала и организаций**
  - **Поддержка рабочих групп**
  - **Учет вклада участников эксперимента**
  - **Получение отчетов с разбивкой по параметрам**
- **ИС реализует процедуры создания, утверждения и редактирования связанных документов**
  - **Регистрация и изменение членства в коллаборации**
  - **Создание и редактирование списков групп и привилегий**
  - **Включение в авторские списки**

# БД персонала и публикаций

- Крупный научный эксперимент может производить десятки и сотни публикаций в год
- Подготовка и публикация статей
  - Отслеживание прохождения публикаций
  - Организация обмена сообщениями между авторами, рецензентами и кураторами
  - Поиск по публикациям
  - Отслеживание публикаций  во внешних ИС
- Организация презентаций и докладов на конференциях
  - Составление списка конференций и доступных докладов
  - Организация звонка для спикеров и отбор
  - Прием, рецензирование и утверждение названий, тезисов и слайдов
  - Отслеживание публикации материалов
- В ATLAS эти функции выполняет ИС Glance
- На SPD предполагается централизованная ИС для использования совместно с  другими экспериментами

# Организаци баз данных ATLAS

- Данные размещены в нескольких БД
  - Три рабочих (production) базы данных:
    - Онлайн, с доступом из сети эксперементальной зоны ATLAS
    - Оффлайн, для автономной обработки данных
    - Распределенные вычисления, для Grid
  - Две полных копии БД для он-лайн и распределенных вычислений
  - Две базы для разработки приложений и тестирования
- Для удобства пользователей и во избежание перегузки доступ к БД производится через интерфейсы (Front-end)
  - Frontier для доступа из заданий обработки и анализа
    - обеспечить одновременное выполнение более 300 тыс. заданий
  - Сервера DDM и PanDA для доступа к датасетам и информации о задачам обработки и анализа
  - Клиентские сервера AMI и COMA для доступа к метаданным

# Базовая технология БД на ATLAS

- На момент начала разработки программного обеспечения эксперимента ATLAS для хранения большого количества структурированных данных была выбрана СУБД Oracle:
  - поддерживался CERN IT, включая оплату стоимости лицензий
- Многие критические по времени приложения размещаются в инфраструктуре Oracle
  - Condition database, AMI, COMA, ProdSys/PanDA, Rucio, AGIS (ИС Grid), Glance (персонал, авторство, доклады.)
- Приложения ATLAS разрабатывались используя инструменты Oracle для оптимизации производительности
- Такой подход — наличие только одной базовой технологии — позволил обеспечить надежную и производительную централизованную систему баз данных, управляемую совместно CERN (системный уровень) и ATLAS (прикладной уровень)

# Проблемы

- Приложения на Oracle со временем увеличились в размерах и сложности и показали хорошую надежность и производительность
  - Oracle может работать очень быстро, если схемы баз данных и запросы хорошо разработаны и оптимизированы.
- Реляционные БД подходят не для всех видов данных
  - Измерения временных рядов, производимые DCS (Системой управления детектором), могут быть более просто представлены парами время:значение. Из-за их огромных размеров данные DCS должны быть сжаты перед записью в Oracle
- Схемы Oracle должны быть тщательно разработаны заранее, а затем их трудно расширить или изменить.
  - В ряде приложений данные не имеют фиксированной схемы
- Наличие только одной базовой технологии вынудило некоторые приложения, которым не нужна реляционная информация, использовать фиксированные схемы, которые могут быть не совсем оптимальными
- Эксперимент оказался критически зависим от лицензионной и коммерческой политики компании разработчика

# Не только Oracle

- Появились и развились Open-Source средства анализа данных, которые могут работать с огромными объемами менее структурированных данных
- Ближе к концу Run 1 в 2012 году и во время LS1 (2013-2014) был протестирован ряд новых решений для структурированного хранения данных ("Базы данных NoSQL") в качестве back-end для новых приложений
  -  и множество связанных с ним инструментов и форматов данных
  - Cassandra, MongoDB и т.д.
- В основном это системы, ориентированные на пары ключ-значение или столбцы
- По результатам проведенных исследований было решено создать и обеспечить поддержку кластера Hadoop для новых приложений со всеми связанными инструментами
- На Hadoop, была перенесена система учета управления данными (data management accounting) и учета для использования в вычислительных задачах

# ATLAS EventIndex

- Первой ИС ATLAS, разработанным непосредственно для кластера CERN Hadoop, стал EventIndex
- EventIndex – единый каталог всех событий, полученых, обработанных и смоделированных в эксперименте ATLAS
- На конец 2022 г. содержал 280 миллиардов записей событий с детектора, и 60 миллиардов MC событий
- Каждая запись включает краткую информацию о событии а также указатели на файлы содержащие полные данные о этом событии.
- Основные варианты использования:
  - Выборка событий по номеру, формату и версии
  - Подсчет и отбор событий на основе триггерных решений
  - Проверка полноты и согласованности данных, etc...

# Event Index

- EventIndex разрабатывался в 2012-2013 гг для нужд Run 2 с использованием самых передовых на тот момент технологий обработки и хранения данных.
- Запущен в 2015 году, в течении полугода тестировался
- Для увеличения производительности был добавлен дополнительный индекс на Oracle
- Успешно проработал до конца Run 2
- Из-за существенное увеличение потока и темпа поступления данных в Run 3 и Run 4 возникла необходимость в увеличении производительности
- В 2021 году модернизирован с использованием Apache HBase для хранения данных и Apache Phoenix для организации запросов в формате SQL

**На примере EventIndex мы видим, что информационные системы должны эволюционировать по ходу проведения эксперимента, подстраиваясь под новые задачи и изменения потоков и объемов данных**

# Информационные системы

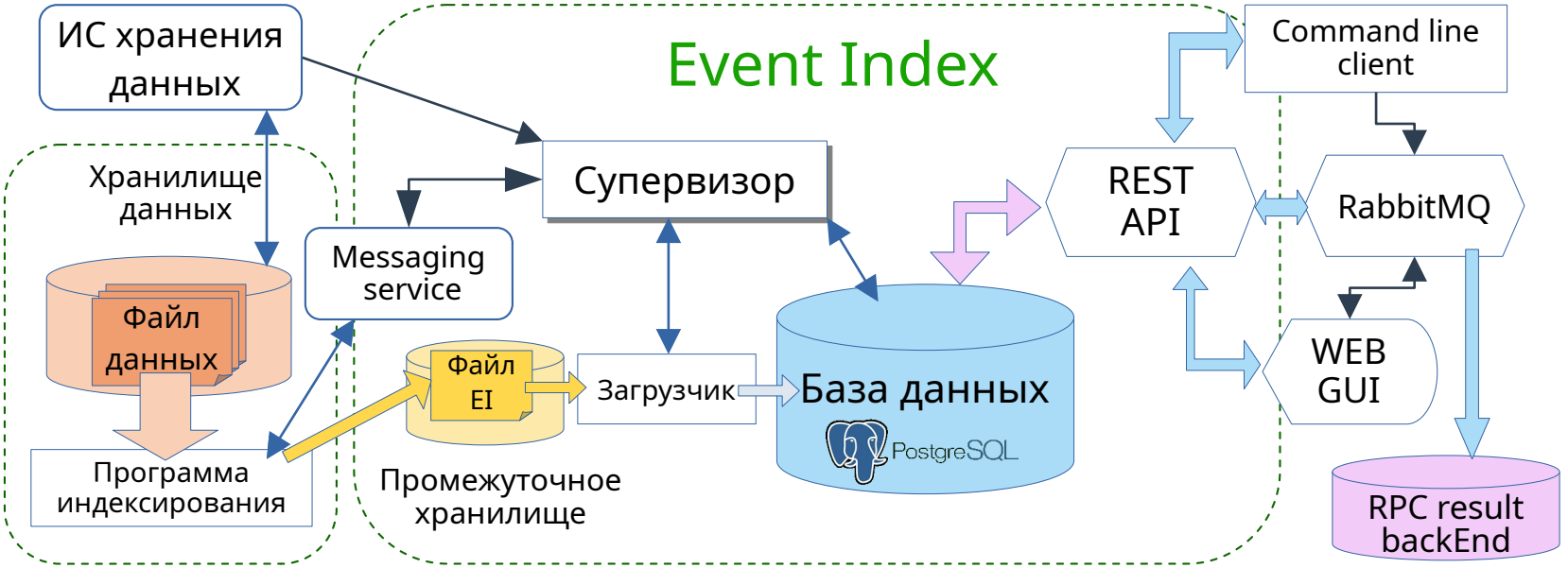
- Базы данных не существуют сами по себе, как правило являются ядром информационной системы
- Помимо БД и СУБД ИС могут содержать
  - Средства сбора информации
  - Средства транспортировки информации
    - Брокеры и обработчики сообщений
  - Интерфейсы пользователей
  - Интерфейсы приложений (API)
  - Сервера приложений (например кэширующие прокси)
  - Оркестровщики (supervisor) и мониторинг
  - Интерфейсы внешних информационных систем
- ИС экспериментальной установки как правило тесно взаимодействуют между собой
- ИС могут взаимодействовать с внешними системами (ИС ускорителя, технические службы, сетевые службы)

# SPD Event Index

- SPD Event Index разрабатывается как комплексная информационная система, которая должна обеспечить
  - Получение информации о событиях эксперимента и моделированных данных путем индексирование файлов данных содержащих информацию о этих событиях
  - Передачу этой информации и запись в базы данных
  - Доступ к информации пользователей через интерактивные и асинхронные интерфейсы
    - Клиент командной строки, Web GUI, сервис запросов
  - Доступ к информации для программ обработки и анализа данных через API
- Запись события будет содержать:
  - Идентификаторы события (номер Runa и события)
  - Информацию о результатах онлайн фильтра
  - Указатели на постоянные файлы, содержащие информация о событии, в различных форматах и версиях обработки
    - Информация о файлах добавляется по мере накопления данных
  - Важные параметры события по которым проводится отбор
    - Добавляются в результате обработки данных

# SPD Event Index

- Ожидаемый размер  записи событий будет существенно меньше чем для ATLAS
  - Отсутствие триггерной информации - вместо нее закодированы результат ОФ
  - Отсутствие производных файлов DAOD — меньшее количество ссылок на реплики
- Меньший объем данных позволит использовать обычную реляционную СУБД вместо гибридного решения на основе HBase
- Общая архитектура Event Index аналогична ATLAS EI



# SPD Event Index: разработка

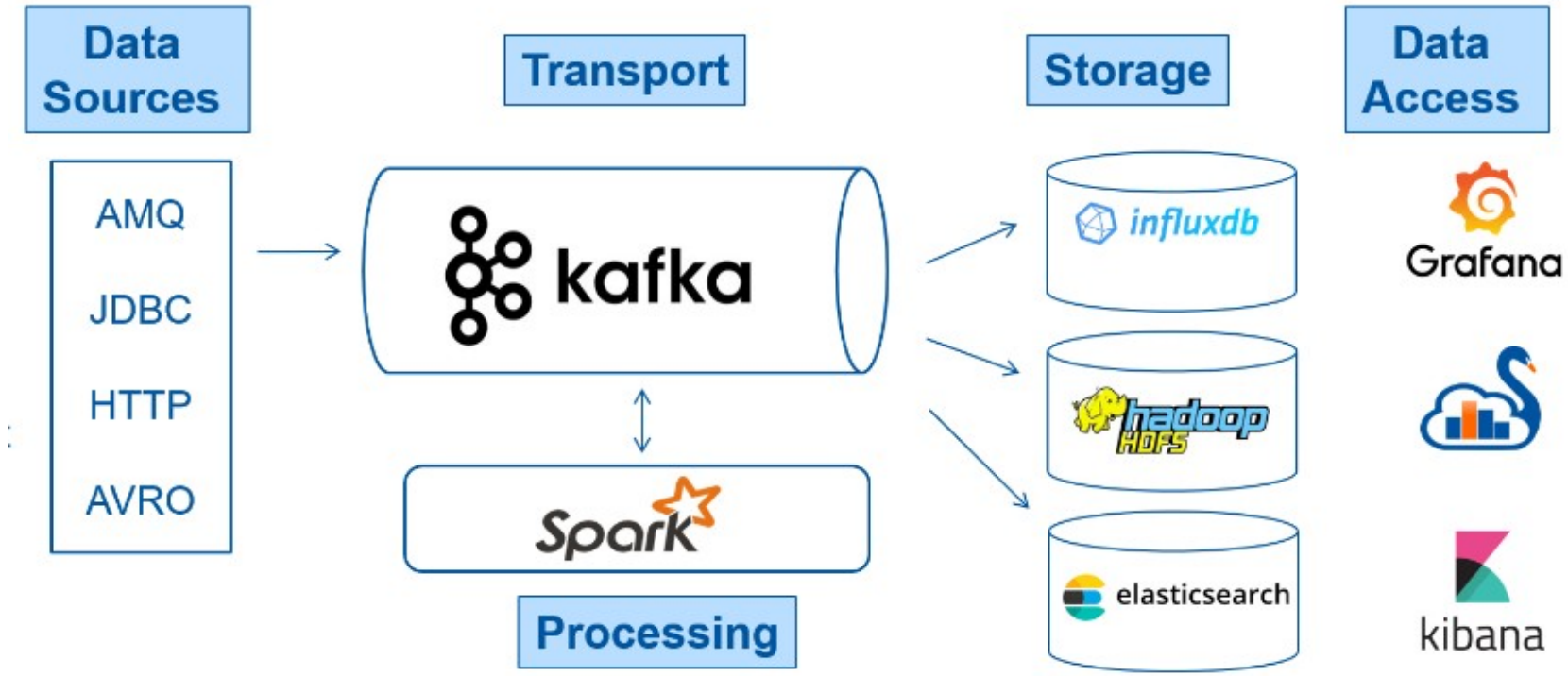
- ИС эксперимента SPD только создаются
- Создание Event Index естественно начать с частей не зависящих от других подсистем установки
- В настоящее время идет разработка системы хранения и управления данными и служб имеющих дело с ней
  - Загрузчики данных
  - Клиентское API
  - Клиенты командной строки
  - Веб-интерфейс
  - Асинхронная служба ,запросов
- В качестве платформы выбрана СУБД PostgreSQL
- Ожидаемая скорость поступления данных в EI составляет ~20-30 тысяч событий в секунду.
- Проводится исследование способов повышения скорости записи больших массивов данных
  - “bulk loading” с использованием специализированных интерфейсов (asyncpg)
  - Оптимизация параметров записи
  - Параллелизация, оптимизация IO носителей, etc.

# SPD Event Index: API

- Разработан простой программный интерфейс для обмена данными в формате  REST (Rest API)
- Используется фреймворк Flask для языка python
  - RestAPI осуществляет preprocessing данных(пар), введенных пользователем со стороны графического интерфейса
  - подключается к базе данных “eventindex”,
  - осуществляет SQL- запрос к таблице “events” через интерфейс psycopg2
  - возвращает список FUID_RAW - указателей на файлы
- Разработан простой графический интерфейс пользователя
  - Для разработки графического клиента был выбран современный фреймворк Angular, который даёт возможность легко, быстро и эффективно строить графический интерфейс с широким функционалом.
- Веб-интерфес позволяет запрашивать данные по идентификатору события, вводимому вручную либо из файла

# Monitoring

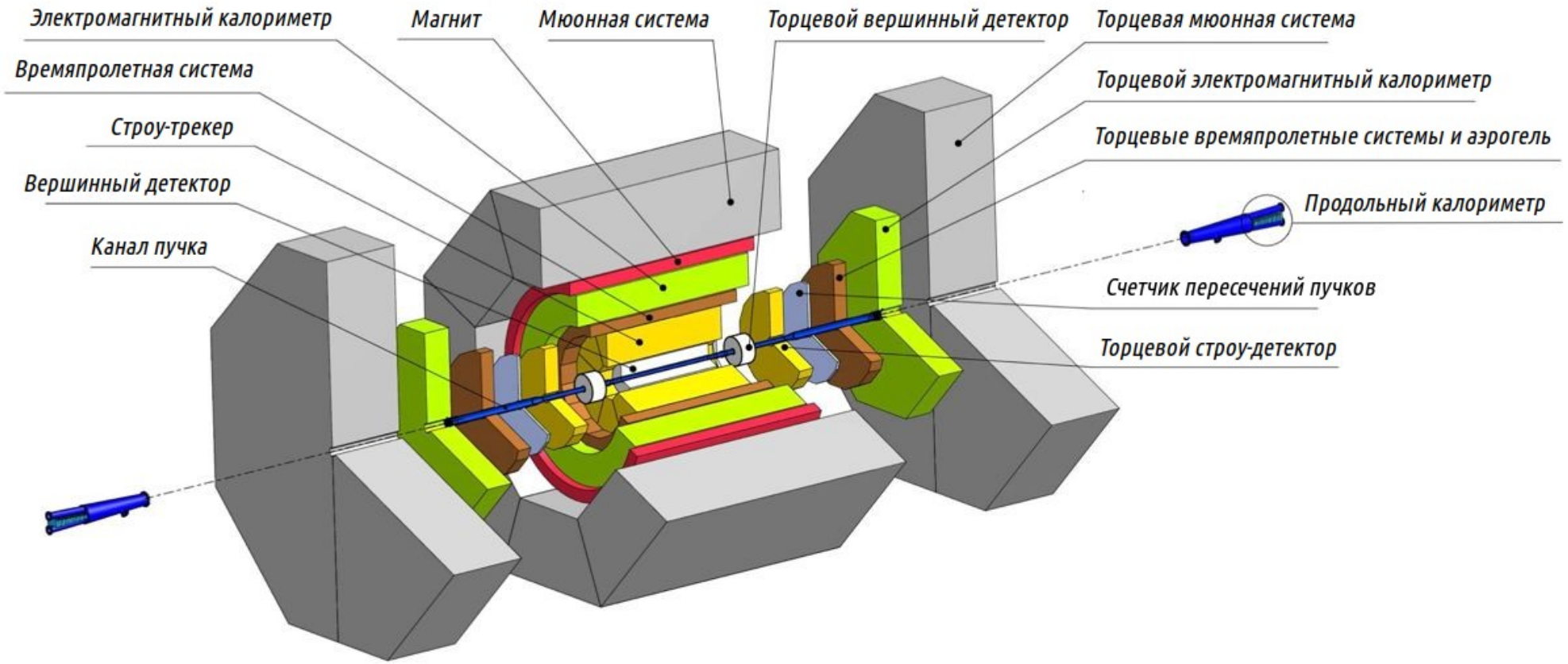
- Система мониторинга ATLAS использует общую для CERN инфраструктуру MonIT
  - Основана на публичных и открытых технологиях Collectd, Kafka, Spark, Elasticsearch, InfluxDB, Grafana
  - Использует модульный подход с конкретными задачами в отношении потока данных мониторинга



# Monitoring: компоненты

- Data Sources — сбор и проверка информации из различных источников
  - Данные передаются в формате JSON без жесткой схемы
- Transport - транспортировка и переработка данных
  - Все данные направляются в кластер Apache Kafka, который обеспечивает единую транспортную платформу с высокой пропускной способностью и низкой задержкой.
- Потоковая обработка при помощи Apache Spark
  - используется для проведения различных операций с данными мониторинга
    - обогащение, агрегирование и корреляция данных
- Данные (3 ТБАЙТ  в день) хранятся
  - в различных конечных точках хранения
  - в различных форматах и степенх детализации для различных типов отображения и доступа
- Доступ к данным мониторинга в MonIT можно получить с помощью хорошо известных технологий визуализации и анализа данных
  - Kibana, Grafana и Swan (проект CERN, основанный на Jupyter)

# SPD


Электромагнитный калориметр
Магнит
Мюонная система
Торцевой вершинный детектор
Торцевая мюонная система
Времяпролетная система
Торцевой электромагнитный калориметр
Строу-трекер
Торцевые времяпролетные системы и аэрогель
Вершинный детектор
Продольный калориметр
Канал пучка
Счетчик пересечений пучков
Торцевой строу-детектор

# Информационные системы SPD

- На установке SPD будут использоватся различные БД и информационные системы:
  - Базы данных оборудования и подключений
  - Базы данных условий и калибровок
  - Системы управления распределенными вычислениями и хранением данных
  - БД физических метаданных
  - Event Index
  - Системы мониторинга
  - Системы ведения журналов и учета
  - База данных персонала и публикаций
- При разработке этих систем будет использоваться опыт полученный на ATLAS и других экспериментах
  - Некоторые системы, такие как Rucio, PanDA и, возможно, CREST, будут адаптированы для использования на SPD
  - Для других, таких как Event Index, берется общая структура при существенно отличной реализации компонент
- Следует предусмотреть адаптацию к изменяющему программно-аппаратному окружению

# Информационные системы SPD

- По возможности будут использоваться решения общие с другими экспериментами на ускорительном комплексе NICA
- Предполагается использовать общую службу ведения электронных журналов и документации
- В качестве основной платформы будет использована реляционная СУБД PostgreSQL
- Для снижения стоимости и затрат на техническое обслуживание следует создать кластер баз данных, построенный на общей базе серверов и хранилищ
- Такой PostgreSQL-сервис будет централизованно поддерживаться квалифицированными специалистами, что критически важно для стабильности и производительности расположенных на нем БД
- В зависимости от задач возможно использование других платформ, таких как NoSQL БД

# Заключение

- Любой большой эксперимент нуждается в сложной системе баз данных
  - Сложное устройство, наличие различных подсистем
  - Большое количество каналов сбора гетерогенной информации
  - Большие объемы данных
  - Сложная система обработки этих данных
- Эксперименты меньшего масштаба также требуют наличие баз данных, меньшего масштаба и сложности
  - Нейтринные эксперименты Байкал, Minerva, с существенно меньшими частотами событий и объемами данных
  - Решения от больших эксперементов для них могут не подойти
    - Для эксперимента Minerva изначально предполагалось использование Condition Database сходной с ATLAS.
    - Получившаяся в результате система оказалась громоздкой, медленной и сложной в использовании
    - Ту же задачу удалось решить используя MySQL с существенно меньшими затратами

# Заключение

- Базы данных и приложения должны разрабатываться с целью обеспечения масштабируемости, имея в виду долгосрочную работу в изменяющихся условиях, с различными потоками данных и частотой запросов.
  - Должна быть обеспечена адаптация к различным форматам данных и их содержанию.
  - Также к различными версиями программного обеспечения
- Следует отдавать предпочтение решениям с открытым исходным кодом

- Разработка и внедрение обычно занимает довольно длительный период, некоторые технологии могут устареть или стать недоступными, и наоборот могут появиться другие.
  - Необходимо иметь в виду возможность перехода на новую платформу
  - Следует поощрять использование модульной архитектуры контейнеров и похожих решений

# Заключение

- Для снижения стоимости и затрат на техническое обслуживание используют кластеры баз данных, построенные на общей базе серверов и хранилищ
  - Такие кластеры обслуживают различные типы рабочей нагрузки приложений
- Использование одной технологии для обработки и хранения всех данных эксперимента
  - С одной стороны такой подход позволяет упростить поддержку и разработку приложений
  - С другой стороны для разных данных сложность, объемы и потоки могут существенно различаться, что может требовать использование различных технологий.
- С течением времени растет объем и сложность данных, возникает необходимости в изменении имеющих и создании новых БД
  - Появление новых информационных технологий, устаревание имеющихся, вопросы лицензирования и финансирования могут потребовать изменения используемых систем БД

# Заключение

- Должна быть проведена тщательная оценка объемов и потоков данных а также способов доступа к ним
  - Схемы базы данных, объекты и запросы должны быть разработаны в соответствии с этими оценками, чтобы избежать проблем с производительностью и стабильности
- При проектировании системы необходимо уделить внимание  интерфейсам для доступа к БД пользователей и программ.
  - Должны обеспечивать требуемые потоки данных, частоту запросов и время реакции
- Распределенных вычисления порождают динамическую нагрузку на базы данных с высокими пиковыми значениями. Следует применять трехуровневую модель организации ИС
  - Использование промежуточного уровня между клиентом и БД помогает избегать перегрузки серверов БД при большом потоке запросов
- Перед запуском в обработки реальных данных следует провести тщательное тестирование с соответствующими объемами псевдоданных в отдельном сервисе базы данных разработки

Разработка БД

Интерфейсы

# SPD


TO JOIN DATABASE DEVELOPMENT

приложения

БД admin

# BACKUP

# PostgreSQL

- PostgreSQL это а объектно-реляционная СУБД с открытым исходным кодом.
  - Поддержка многочисленных типов данных
    - Численные, булевые, символьные, составные, сетевые типы данных, перечисление, типы «дата/время», массивы, etc.
  - Поддержка JSON, что позволяет использовать schema-less данные
    - Встроенные  специализированные JSON операторы и функции
  - Поддержка пользовательских объектов и их поведения, включая типы данных, функции, операции и индексы.
  - Индексирование: частичное, функциональное, GiST, GIN, BRIN
  - Функции виртуальных таблиц, материализованные представления
  - Возможность добавления/изменения столбцов

# PostgreSQL

- Обеспечивает высокую надежность и производительность:
  - Соответствие принципам ACID (атомарность, изолированность, непротиворечивость, сохранность данных)
  - Многоверсионный контроль конкурентных транзакций и изоляция транзакций
    - возможность изменения баз данных одновременно разными пользователями.
    - минимизирует блокировки данных и позволяет увеличить производительность
  - Журналы  опережающей записи (Write Ahead Logging)
    - фиксирующая все изменения до их фактического применения.
  - Резервное копирование и восстановление
  - Возможность восстановления базы данных Point in Time Recovery
  - Откат состояние базы данных к предыдущему стоянию используя WAL

- **Основные ограничения**

| | |
|---|---|
| Максимальный размер БД | Неограничен |
| Максимальный размер таблицы | 32 TB |
| Максимальный размер строки | 1.6 TB |
| Максимальный размер поля | 1 GB |
| Максимальное количество строк в таблице | Неограниченно |
| Максимальное количество столбцов в таблице | 250-1600 |
| Максимальное количество индексов в таблице | Неограничено |

# Hadoop: Приложения учета

- Первым приложением, которое было перенесено на Hadoop, была система учета управления данными (data management accounting) и учета из использования в вычислительных задачах
- Основные приложения для распределенных вычислений (Rucio и ProdSys/PanDA) имеют очень высокую скорость транзакций
  - База данных Oracle очень эффективно справляется с этим большим потоком информации
  - Такие приложения, как мониторинг и учет, которые только считывают данные из базы данных, вместо этого лучше подходят для других систем хранения, при этом необходимые данные извлекаются из Oracle и форматируются соответствующим образом для ожидаемых запросов
- Задачи по извлечению соответствующей информации из Oracle и хранению ее в Hadoop выполняются непрерывно и предоставляют входные данные нескольким другим инструментам:
  - Популярность датасетов
  - Мониторинг задач
  - Учет управления данными
  - И т.д. и т.п.

# Apache HBase

- Apache HBase принадлежит к семейству HADOOP.
  - Нереляционная (NoSQL) база данных с открытым кодом
  - Распределенное и масштабируемое хранилище данных
- Данные организованы в таблицы, проиндексированные первичным ключом, который в HBase называется RowKey.
- Для каждого RowKey ключа может храниться неограниченны набор атрибутов (или колонок).
  - Каждый ряд может иметь свою схему
  - Любой атрибут может отсутствовать или присутствовать для каждого ключа, при этом если атрибут отсутствует — это не вызывает накладных расходов на хранение пустых значений.
  - Для каждого атрибута может храниться несколько различных версий. Разные версии имеют разный timestamp.
- Доступ к значениям возможен по ключу и имени колонки

# Apache HBase

- Колонки организованны в группы (Column Family)
  - Список и названия групп колонок фиксирован и имеет четкую схему.
  - Как правило в одну Column Family объединяют колонки, для которых одинаковы паттерн использования и хранения.
  - Данные соответствующие разным Column Family хранятся отдельно, что позволяет при необходимости читать данные только из нужного семейства колонок.
  - Атрибуты, принадлежащие одной группе колонок и соответствующие одному ключу физически хранятся как отсортированный список.
- Свойства такой организации:
  - Относительно небольшой размер
  - Позволяет уникально идентифицировать события
  - Позволяет производить поиск по  интервалам
  - При росте таблиц используется гомогенное пространство RowKey
  - Использование нестандартных паттернов запросов может существенно снизить производительность

# Apache Phoenix

- Apache Phoenix это транзакционная система и операционная аналитика для HBase
  - Принимает SQL запросы
  - Преобразует их в серии сканирований в HBase
  - Напрямую использует API HBase, а также сопроцессоры и специальные фильтры
  - Выдает результаты запросов в стандарте JDBC
- Дизайн RowKey адаптируется к типам и размерам Phoenix за счет небольшого уменьшения производительности
- Phoenix позволяет использовать поля RowKey в запросах, при этом они сохраняются как один объект в HBase
- Добавление функциональности SQL к HBase дает дополнительные преимущества:
  - Структурированные данные проще понимать и поддерживать
  - Стандартная логика запросов, пригодная для сложных запросов

# Monitoring: сбор информации

- Data Sources — сбор и проверка информации из различных источников
- Данные передаются в формате JSON без жесткой схемы
  - Должны присутствовать несколько обязательных полей (ID источника, отметка времени)

- Типы источников:
  - Collectd:
    - собирает, передает и хранит данные о производительности компьютеров и сетевого оборудования
  - HTTP-канал и система обмена сообщениями ActiveMQ:
    - Для передачи информации от внешних производителей данных
    - Производитель информации инициирует передачу
  - Запрос JDBC:
    - Для извлечения  информации из внешних БД

# Monitoring: транспортировка

- Transport - транспортировка и переработка данных
- Все данные, полученные через источники данных MONIT, направляются в кластер Apache Kafka, который обеспечивает единую транспортную платформу с высокой пропускной способностью и низкой задержкой.
- Данные в Kafka передаются в компонент хранения в режиме реального времени (менее 10 секунд задержки)
  - Данные буферизуются на срок хранения 72 часа
  - Позволяет предотвратить потерю данных при любом простое потребителей информации.
- Подход, основанный на Kafka как транспортном уровне
  - разделяет производителей и потребителей данных мониторинга
  - обеспечивает полную изоляцию рабочих процессов
  - защищает пользователей от влияния действий друг друга и проблем с инфраструктурой.

---

Kafka — распределённый программный брокер сообщений для обработки потоковых данных в реальном времени с высокой пропускной способностью и низкой задержкой.

# Monitoring: обработка данных

- **Потоковая обработка при помощи Apache Spark используется для проведения различных операций с данными мониторинга**
  - **“обогащение” данных — добавляется информация из нескольких других источников**
  - **агрегирование данных**
    - **во времени, например, создание сводной статистики по временному интервалу**
    - **по другим измерениям, например вычисление совокупной метрики для набора компьютеров**
  - **корреляция данных**
    - **для обнаружения одновременных аномалий**
    - **для обнаружения сбоев, исходящих из нескольких источников**
- **Пакетная обработка используется для**
  - **повторной обработки и пересчета исторических данных,**
  - **сжатия старых исторических данных и**
  - **извлечения статистических данных и отчетов**

Apache Spark — фреймворк для реализации распределённой обработки неструктурированных и слабоструктурированных данных

# Monitoring: хранение данных

- Все данные, полученные и переданные MonIT (3 ТБАЙТ в день), хранятся в различных конечных точках хранения в различных форматах и степенх детализации для различных типов отображения и доступа
  - HDFS используется для долгосрочного архивирования и автономной аналитики всех данных
    - Сжатый JSON или Parquet для метрик и журналов
  - Elasticsearch (ES) - для кратковременного хранения и индексации данных
    - Необработанные данные, метрики и журналы, хранятся в течение одного месяца в 3-х экземплярах
  - InfluxDB - для средне- и долгосрочного хранения данных временных рядов в необработанном либо в агрегированном формате.
    - Данные хранятся в течение 5 лет и подвергаются выборке с понижением и агрегированию
      - необработанные данные - за 1 неделю, 5-минутные бины за 1 месяц, 1 ч бины за 5 лет.

---

Parquet — это бинарный, колоночно-ориентированный формат хранения данных, изначально созданный для экосистемы hadoop.

# Monitoring: представление данных

- Доступ к данным мониторинга в MonIT можно получить с помощью хорошо известных технологий визуализации и анализа данных
  - **Kibana** предназначена для визуализации метрик и журналов, подключенных к Elasticsearch
    - обеспечивает возможности поиска / фильтрации, а также интерактивное обнаружение данных мониторинга и журналов
  - **Grafana** предназначена для визуализации данных временных рядов, обслуживающих данные как для Elasticsearch, так и для хранилища InfluxDB.
    - предоставляет шаблоны, специальные фильтры, расширенные запросы, списки управления доступом и пердупреждения
  - **Swan** - проект CERN, основанный на Jupyter
    - предоставляет пользователям доступ к данным, хранящимся в HDFS, Elasticsearch и интеграцию с несколькими широко используемыми инструментами физики высоких энергий (HEP) для создания интерактивных ноутбуков.