

[Общая архитектура системы](#)

[Ключевые возможности продукта](#)

[Единый доступ к данным](#)

[Обмен данными](#)

[Федерация данных](#)

[Различные представления данных](#)

[Происхождение данных](#)

[Интеллектуальная оптимизация запросов](#)

[Component Diagram](#)

[Users](#)

[Audit Log](#)

[Dataset](#)

[Pipeline \(UDP\)](#)

[Catalog](#)

[Maintenance](#)

[Profiling](#)

[Lineage](#)

[Deployment View](#)

[HDFS](#)

[Yarn Cluster](#)

[PostgreSQL](#)

Общая архитектура системы

Платформа виртуализации DAMS построена на следующих ключевых технологиях:

- Apache Hadoop
- Yarn
- Spark SQL

В качестве платформы для развертывания DAMS используется Docker Swarm

Пользовательский интерфейс DAMS — это одностраничное веб-приложение (SPA), основанное на следующем технологическом стеке:

- ReactJS + Material UI — уровень представления.
- React Router 4 — маршрутизация в браузере (навигация между страницами).
- Redux — уровень данных и однонаправленный поток данных.
- i18next + реакция-i18next — интернационализация
- fetch — связь с серверным REST API через протокол HTTPS.
- stomp — двусторонняя связь с бэкендом через протокол WebSocket.

Ключевые возможности продукта

Платформа DAMS предоставляет следующие функции виртуализации данных:

Единый доступ к данным

Платформа DAMS обеспечивает единый доступ к данным независимо от их местонахождения и исходного формата. Исходные данные могут храниться вне системы в реляционных (или нереляционных) базах данных, либо внутри системы в файлах разных форматов — json, XML, CSV. Пользователи могут получить доступ к этим данным единым способом, который не зависит от исходного местоположения или формата:

- 1) Запрос с помощью Spark SQL через соединение JDBC.
- 2) Загрузка целые наборы данных (или некоторую часть на основе пользовательских критериев) через REST API.
- 3) Преобразование данных с помощью внутриплатформенных конвейеров или пользовательских программ, которые выполняются на платформе.

Внутрисистемные наборы данных хранятся в файловой системе HDFS платформы. Все исходные наборы данных считываются с использованием соответствующих коннекторов данных Apache Spark. Платформа DAMS хранит метаданные, необходимые для доступа к исходному источнику, в собственной базе данных метаданных (PostgreSQL), а также контролирует, кто и к каким данным имеет доступ.

Доступ к JDBC основан на spark/hive thriftserver (встроенной платформе) и драйвере spark/hive jdbc.

Обмен данными

Платформа DAMS позволяет владельцам данных обмениваться данными с другими пользователями. Владельцы наборов данных могут делиться набором данных напрямую с конкретными пользователями или публиковать его в каталоге и делать его доступным для определенных категорий пользователей.

Платформа позволяет контролировать, какие данные передаются, а какие нет. Некоторые поля можно замаскировать или полностью удалить из опубликованной версии набора данных для всех пользователей или для некоторых категорий пользователей. Также можно ограничить, какие записи являются общими, а какие нет, применив подходящий фильтр.

Пользователи могут получить доступ к этим общим наборам данных так же, как и к своим обычным наборам данных.

Федерация данных

Платформа DAMS позволяет объединять данные из разных источников и

просматривать их или манипулировать ими. Доступ к наборам данных из разных источников можно получить из одного и того же сеанса JDBC, или можно объединить один и тот же конвейер, или одну и ту же пользовательскую программу и т. д. Он основан на обработке набора данных Spark.

Различные представления данных

Пользователи данных могут создавать различные способы просмотра данных, используя виртуальные наборы данных. Они могут выбрать один или несколько исходных наборов данных, определить, какие операции над ними необходимо выполнить, чтобы получить желаемое представление (объединение, фильтрация, объединение и т. д.). Платформа DAMS хранит всю эту информацию в виде «виртуального набора данных». При доступе к этому виртуальному набору данных DAMS выполняет сохраненные операции с исходными наборами данных и предоставляет результаты в виде набора данных.

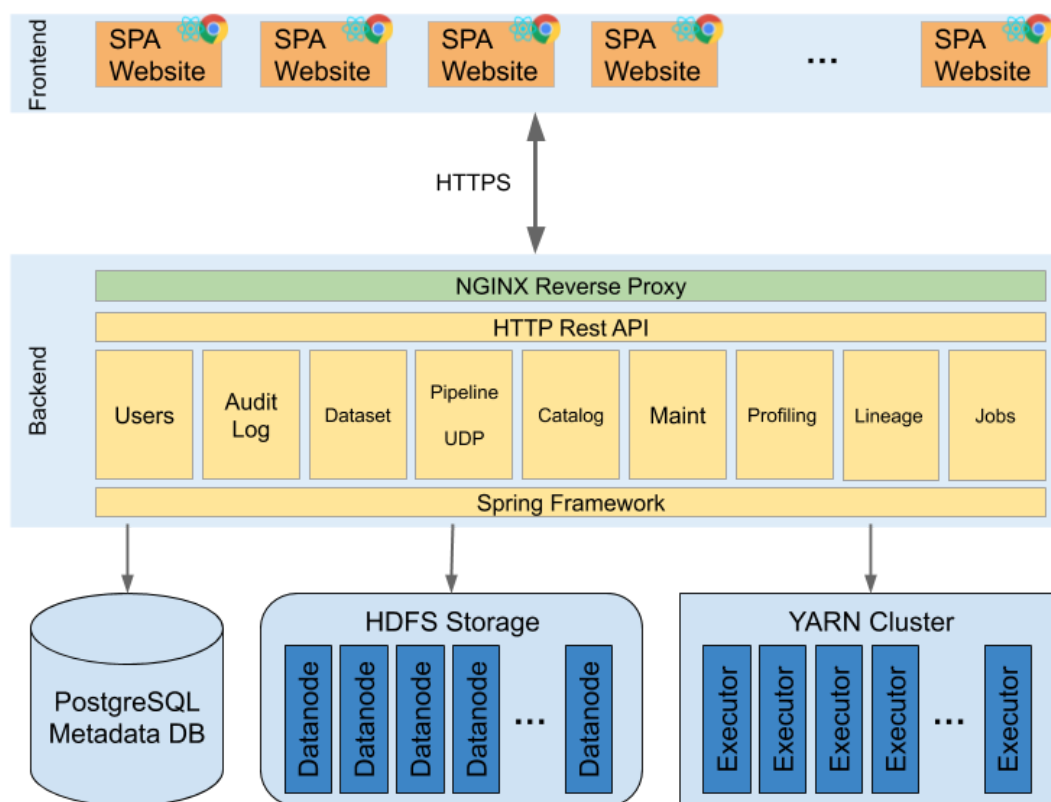
Происхождение данных

Платформа DAMS отслеживает происхождение данных. Таким образом, можно узнать, когда одни данные создаются из других, либо путем публикации, либо путем преобразования через конвейер или пользовательскую программу. Эту информацию могут просматривать авторизованные пользователи, а также она используется для поддержания ограничений на обмен данными. Например, если владелец набора данных отключит доступ к опубликованному набору данных через JDBC, а вы сделаете копию этого набора данных через конвейер, доступ JDBC к копии также будет запрещен.

Интеллектуальная оптимизация запросов

Платформа DAMS предоставляет возможность для реализации механизма высокопроизводительного доступа к данным, оптимизации запросов с несколькими источниками и механизма параллелизации исполнения запросов.

Component Diagram



Бэкэнд DAMS — это приложение Spring, состоящее из набора сервисов. В этом разделе перечислены и описаны наиболее важные сервисы.

Users

- AccountService: управление пользователями и организациями (создание/обновление/архивирование/и т. д.).
- SessionService: сеансы пользователей
- PasswordPolicyService: настройки политики паролей для пользователей

Audit Log

Служба журнала аудита, отвечающая за запись и чтение журнала аудита DAMS. Служба AuditLogValidation отвечает за контроль целостности журналов аудита (чтобы они не были изменены/удалены злоумышленниками).

Dataset

Сервисы для управления наборами данных:

- DatasetAccessService: чтение/запись различных типов наборов данных.
- DatasetColumnService: обработка информации о столбцах табличного набора данных.
- TableDatasetService: функции, специфичные для наборов табличных данных.
- File(Group)DatasetService: функции, специфичные для наборов данных файлов и каталогов.
- DatasetPreviewService: создание предварительного просмотра для различных типов наборов данных.
- MetadataService: общие процедуры для метаданных наборов данных и источников данных.
- DatasetRelationshipService: отслеживает взаимоотношения между наборами данных, например, если один набор данных создается из другого с помощью конвейера или пользовательской программы. Используется для отображения происхождения данных.
- DatabaseSchemaService: процедуры для создания/обновления реляционных таблиц с заданной схемой.
- VirtualDatasetService: управление виртуальными наборами данных.

Pipeline (UDP)

- PipelineExecutionService: выполнение конвейера.
- Несколько служб для каждого типа оператора конвейера, например JoinPipelineStageService.
- PythonService: выполнение кода Python в конвейерах (оболочка интерпретатора Jython) для оператора Jython.
- JavaCompileService: управление кастомными классами Java для выполнения Java UDP

Catalog

Управление каталогами наборов данных:

- PublicationService отвечает за публикацию наборов данных в каталоге.
- SubscriptionService отвечает за подписки на опубликованные наборы данных.
- KeywordService отвечает за управление ключевыми словами каталога.
- ApprovalProcessingService: управляет процессами разрешения публикаций и подписок наборов данных.

Maintenance

Сервисы по обслуживанию системы:

- UpgradeService: запуск обновления через пользовательский интерфейс, ведение истории версий.
- DockerAccessService: общие процедуры для доступа к Docker Swarm.
- LogSettings: управление и ротация различных системных журналов.

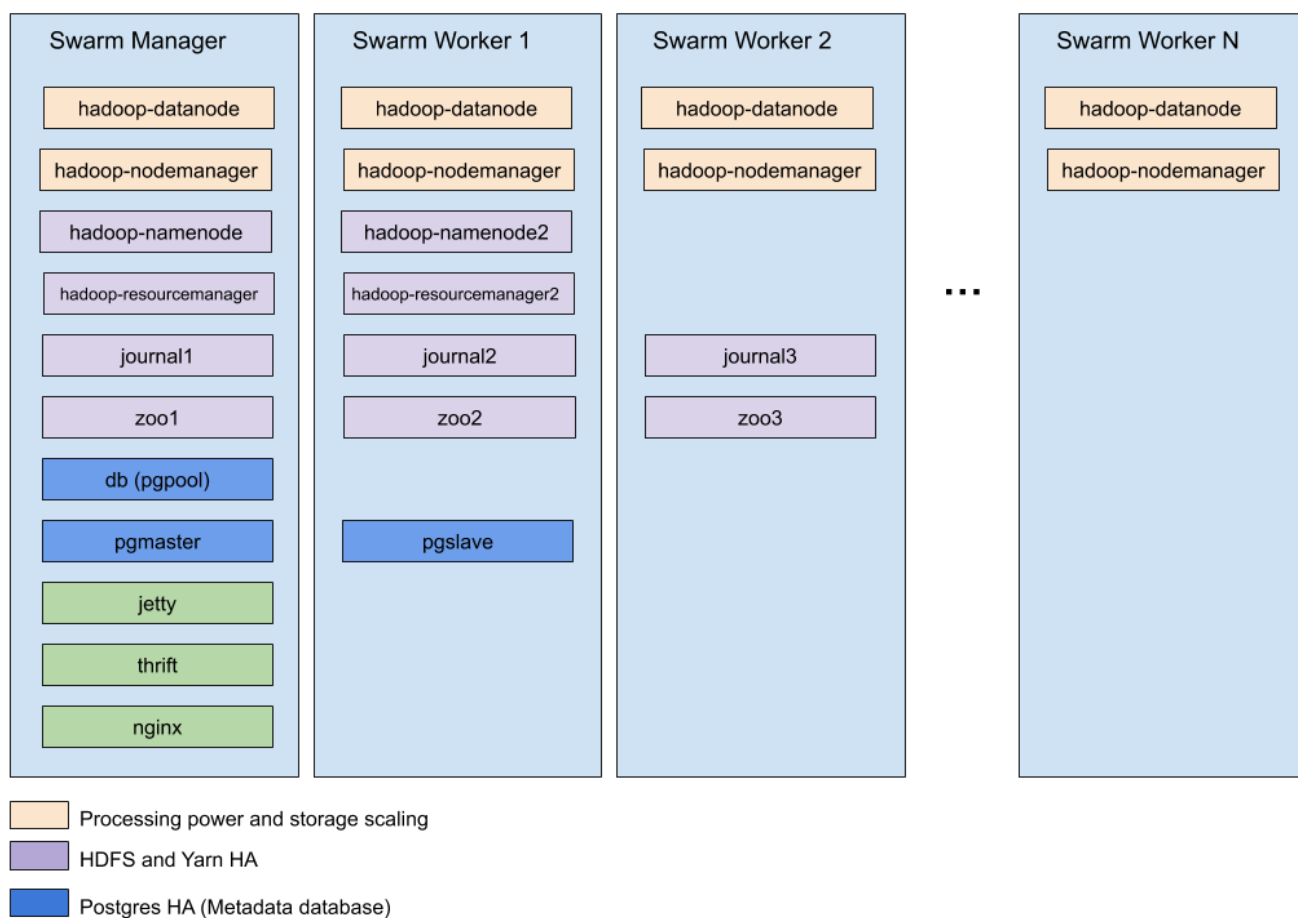
Profiling

Сервисы для профилирования (изучения) данных.

Lineage

Сервисы по управлению отслеживанию происхождения данных.

Deployment View



С точки зрения развертывания DAMS представляет собой набор сервисов, развернутых в Docker Swarm. Есть следующие услуги:

jetty	Серверный сервер DAMS. Предоставляет REST API для пользовательского интерфейса DAMS, общедоступного API и доступа к данным JDBC.
db	Собственная база метаданных DAMS или Интерфейс базы данных (балансировщик нагрузки) в случае развертывания высокой доступности

hadoop-datanode	Узлы данных для HDFS. Используется для хранения/доступа к содержимому данных в HDFS.
hadoop-namenode[1 2]	Узел имени для HDFS. Используется для хранения/доступа к метаданным HDFS (имена файлов/каталоги/т. д.)
hadoop-resourcemanager[1 2]	Менеджер ресурсов YARN, отвечающий за предоставление ресурсов для задач DAMS.
hadoop-nodemanager	Менеджеры узлов YARN. Контейнеры YARN, выполняющие задачи DAMS Spark, выполняются в этих докер-контейнерах.
journal[1 2 3]	Журнал Hadoop
zoo[1 2 3]	Смотритель зоопарка Hadoop
pgmaster	Главный узел сервера базы данных метаданных плотин
pgslave1	Подчиненный узел сервера базы данных метаданных плотин
nginx	Интерфейсный http-сервер DAMS (NGINX) обеспечивает доступ к статическим файлам, завершение SSL.
nginx-gen	Вспомогательный сервис для настройки dams_nginx
adapter-hbase2	Вспомогательный сервис для доступа к наборам данных hbase2.

Платформу DAMS можно развернуть в режиме высокой доступности (HA).

HDFS

Имеется два namenode (namenode и namenode2), в любой момент времени один из них активен, другой находится в режиме **горячего резерва**. Все метаданные HDFS распределяются между узлами имен с помощью кворума узлов журнала (journal1, journal2, journal3). Кворум Zookeeper используется для выбора текущего **активного** узла имени в случае сбоя (zoo1, zoo2, zoo3).

Yarn Cluster

Аналогично с HDFS HA есть два узла менеджера ресурсов (resourcemanager и resourcemanager2), в любой момент времени один из них активен, другой — в **режиме ожидания**. В случае сбоя nodemanager-ы (текущие рабочие) перерегистрируются на следующем доступном менеджере ресурсов. Выбор активного менеджера ресурсов осуществляется с помощью кворума Zookeeper.

PostgreSQL

PostgreSQL развертывается в конфигурации «master-slave», каждый запрос WRITE обрабатывается master и реплицируется на slave с помощью диспетчера репликации, запросы READ распределяются между master и slave с помощью

службы pgpool. В случае сбоя master, slave становится новым master. Предыдущий master будет автоматически переключен в slave режим после повторного появления в swarm.