DataTalks

ARENADATA
Hadoop

ARENADATA

Основы Hadoop. Современные подходы к обработке Big Data

В современном мире сложно представить компанию, которая бы не сталкивалась с проблемами сбора, обработки и анализа больших объемов данных. Объемы данных, которыми оперирует бизнес, растут экспоненциально с каждым годом, ценность данных тоже растёт, и для эффективного решения возникающих задач и проблем и поиска новых возможностей и идей необходимо владеть методами и технологиями Big Data.

Данный курс рассчитан на слушателей, которые хотят построить карьеру в сфере корпоративных систем обработки и хранения данных: аналитиков, разработчиков, архитекторов корпоративных систем, Data Scientists. Понимание ключевых подходов и инструментов Big Data позволит:

- Выбирать и эффективно использовать наиболее правильные и современные инструменты Big Data
- Понимать и использовать различные подходы к решению задач обработки больших данных
- Планировать, создавать и развивать комплексные системы хранения и обработки больших данных enterprise-уровня
- Знать и учитывать существующие преимущества и ограничения систем Big Data

Ростелеком

Ростелеком

ARENADATA

Виктор Бородаенко

Ведущий архитектор решений Big Data

Arenadata

v@arenadata.io

Основы Hadoop.

Современные подходы к обработке Big Data

ARENADATA

Модуль 1. Введение. Основные понятия Big Data

- Что такое Big Data?
- 3V, 4V, 5V, 10V Больших данных
- Что такое структурированные, полуструктурированные и неструктурированные данные?
- Что такое Data Lake?
- Что такое лямбда и каппа-архитектуры загрузки и обработки данных?

ARENADATA

Что такое Big Data?

Big Data – не сами данные, а концепция:

- Данные – огромных (для конкретной компании) объёмов и разнообразной структуры
- Технологии – горизонтально масштабируемые, ориентированные на параллельные вычисления
- Методы – как классические, так и современные узкоспециальные, в т.ч. алгоритмы машинного обучения и AI
- Источники – как классические, так и новые, в т.ч. открытые: социальные сети, Internet of Things, научные данные и т.д.

ARENADATA

3V Больших данных

Ростелеком

Volume

Объемы данных невозможно хранить и обрабатывать с помощью классических инструментов (СУБД, файловые системы и т.д.)

Variety

Данные могут быть в самых разнообразных форматах: классические реляционные, XML- или JSON-файлы, изображения, звук, видео, логи произвольных форматов и т.д.

Velocity

Данные могут поступать и должны быть обработаны на скоростях, близких к реальному времени; классический batch-подход более не приемлем по скорости работы

ARENADATA

4V, 5V, 10V, *V Больших данных

Ростелеком

Volume
Объемы данных невозможно хранить и обрабатывать с помощью классических инструментов (СУБД, файловые системы и т.д.)

Variety
Данные могут быть в самых разнообразных форматах: классические реляционные, XML- или JSON-файлы, изображения, звук, видео, логи произвольных форматов и т.д.

Value
Данные могут иметь различную ценность

Velocity
Данные могут поступать и должны быть обработаны на скоростях, близких к реальному времени; классический batch-подход более не приемлем по скорости работы

Veracity
Данные могут отражать разные аспекты «истины»

Validity
Данные могут существенно различаться по качеству, степени наполнения и соответствия «истине»

Variability
Данные могут существенно изменяться с течением времени

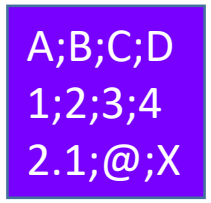
Venue
Данные могут приходить из множества разных источников, быть распределенными и иметь разных владельцев

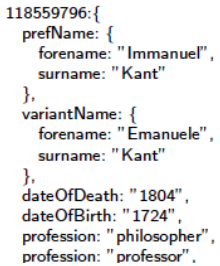
Vocabulary
Данные могут описывать другие данные, быть справочниками, онтологиями и т.д.

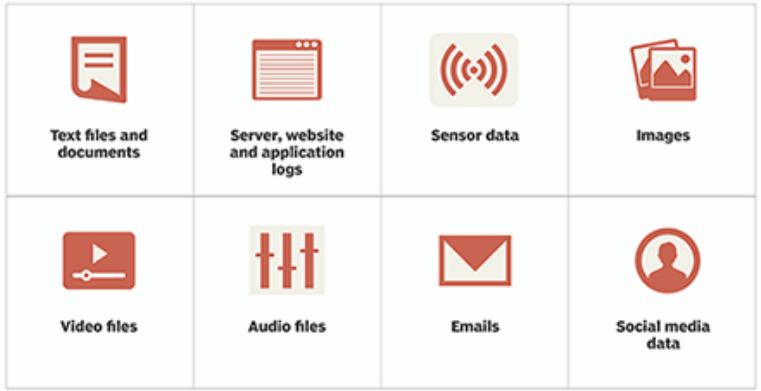
Vagueness
Данные могут казаться не несущими ценности или дающими лишь большую неопределенность

ARENADATA

Различные виды данных







Структурированные

- Реляционные данные
- Таблицы
- Документы фиксированного содержания

Полуструктурированные

- XML-файлы с известной схемой
- JSON-файлы с известной схемой

Неструктурированные

- Текстовые документы произвольного содержания
- Изображения, видео, аудиофайлы
- Информация из открытых источников
- Данные сенсоров

ARENADATA

Какие данные считаются «Большими»?

- «Большие» данные характеризуются не конкретным объёмом, а способностью конкретной компании хранить и обрабатывать свои данные и управлять ими
- Для кого-то «Большие данные» – это несколько терабайт таблиц, расположенных в реляционном корпоративном хранилище
- Для кого-то «Большие данные» – несколько геораспределенных гибридных кластеров, хранящих петабайты информации

ARENADATA

Загрузка и обработка данных в реальном времени

Ростелеком

Лямбда-архитектура – гибридный подход к загрузке и обработке данных

- Данные загружаются в два потока
- Данные, поступающие в реальном времени, загружаются и обрабатываются в реальном времени
- Согласно расписанию выполняется batch-обработка всех данных хранилища

Каппа-архитектура – загрузка и обработка данных в real-time или near-real-time

- Данные загружаются в реальном времени as is в исторический слой хранилища
- В параллельном потоке данные обрабатываются в реальном времени и через специальный слой (serving layer) поступают в соответствующие системы-приёмники (детальный слой хранилища, витрины, системы отчетности и т.д.)
- При необходимости пересчёта данные перегружаются из исторического слоя

ARENADATA

Архитектура Data Lake

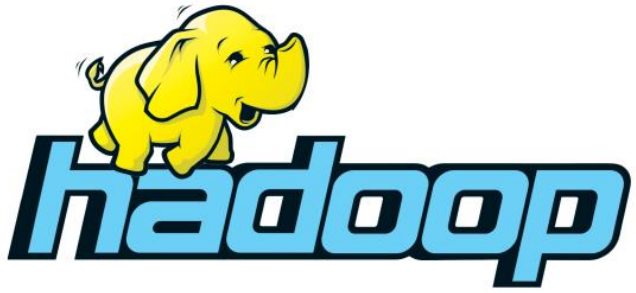
Ростелеком

Data Lake – концепция построения корпоративного хранилища данных

- Данные загружаются as is, без обработки и трансформаций (против ETL\ELT-процессов в классическом Data Warehouse); допускается как классическая batch-загрузка, так и лямбда- и каппа-архитектуры
- Обработка и трансформация производится по мере необходимости, структура данных определяется только на данном этапе (подход schema-on-read против schema-on-write классического хранилища)
- Данные могут загружаться и храниться в любом формате, так же, как в системах-источниках (против преобразования к реляционному виду в Data Warehouse)
- Основные потребители данных Data Lake – аналитики данных и Data Scientists (в Data Warehouse – преимущественно бизнес-пользователи)

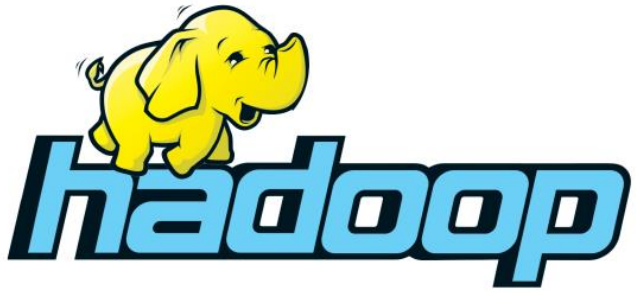
ARENADATA

Ростелеком



- Общий обзор платформы
- История Apache Hadoop
- Ядро Apache Hadoop: HDFS, MapReduce, YARN
- Другие компоненты платформы
- Основные вендоры Hadoop
- Будущее Hadoop

Общий обзор платформы Hadoop



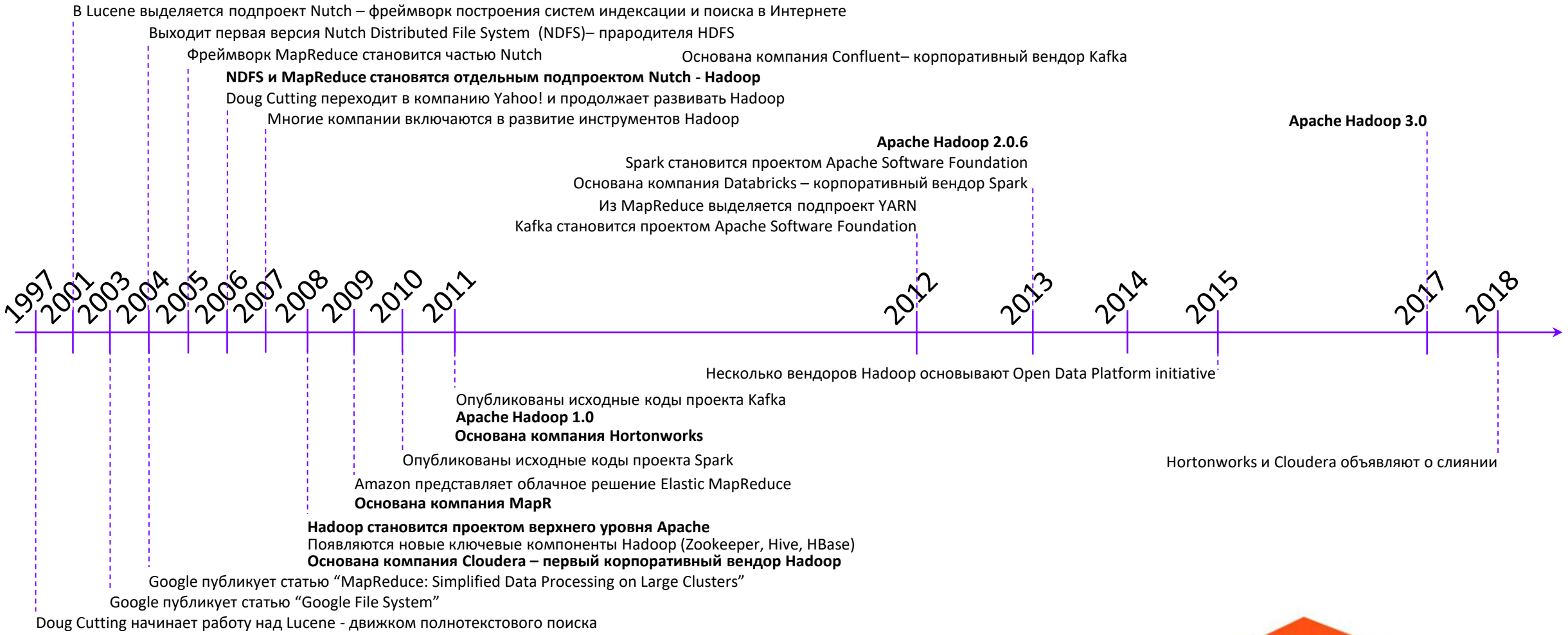
- Apache Hadoop – open source технология, набор библиотек и утилит для распределенного, отказоустойчивого, горизонтально масштабируемого хранения больших объемов данных и их массивно-параллельной обработки

  Основные компоненты Hadoop – распределенная файловая система Hadoop Distributed File System (HDFS), система управления кластером и оркестрации параллельных вычислений YARN, фреймворк распределенных вычислений Apache MapReduce

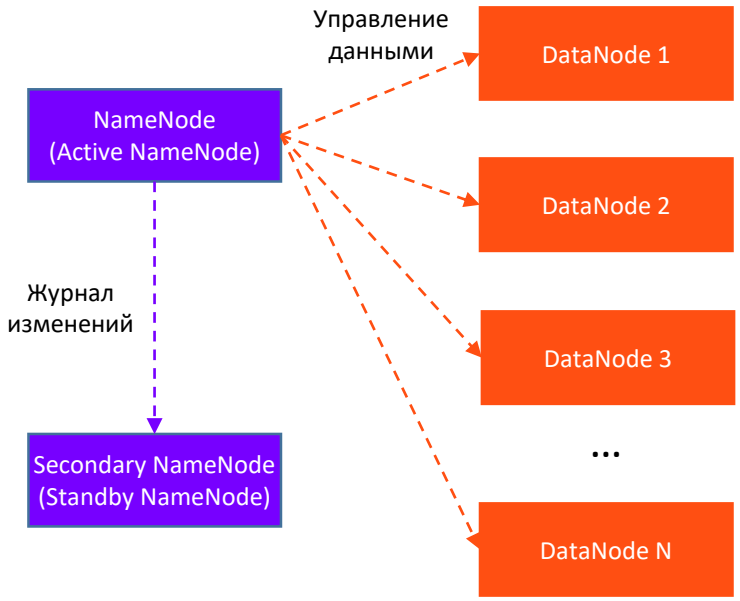
  Помимо основных, в состав Hadoop включается множество других компонентов, утилит и технологий, использующих основные компоненты Hadoop как связующие элементы

- Apache Hadoop – де-факто отраслевой стандарт для построения платформ обработки и хранения Больших Данных, определивший вектор развития множества систем и технологий

История Hadoop

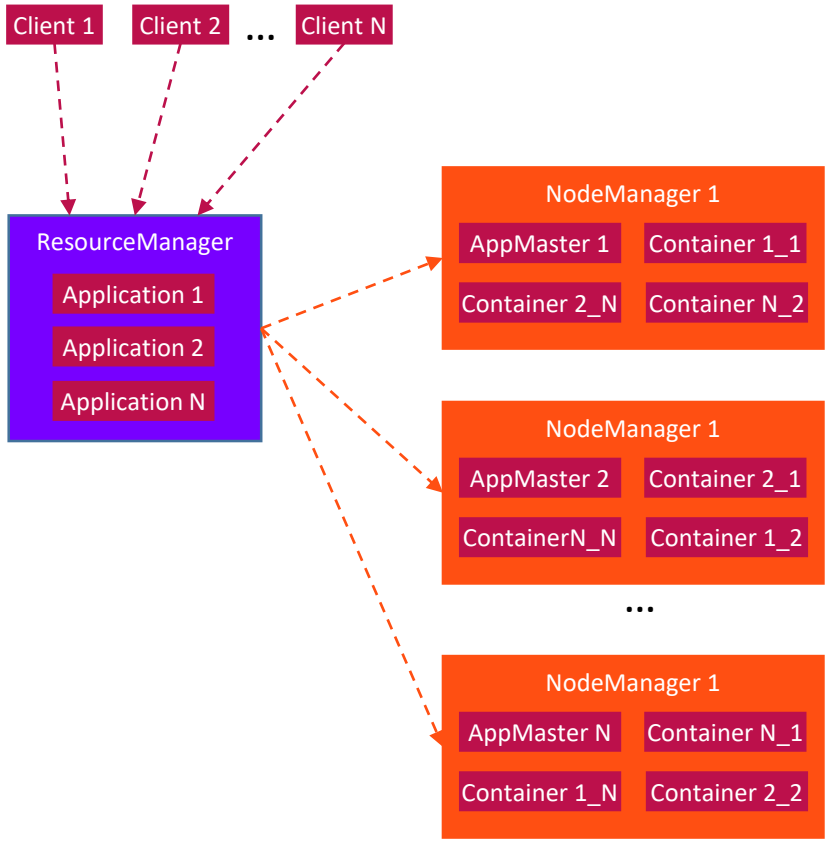


Ядро Hadoop: HDFS



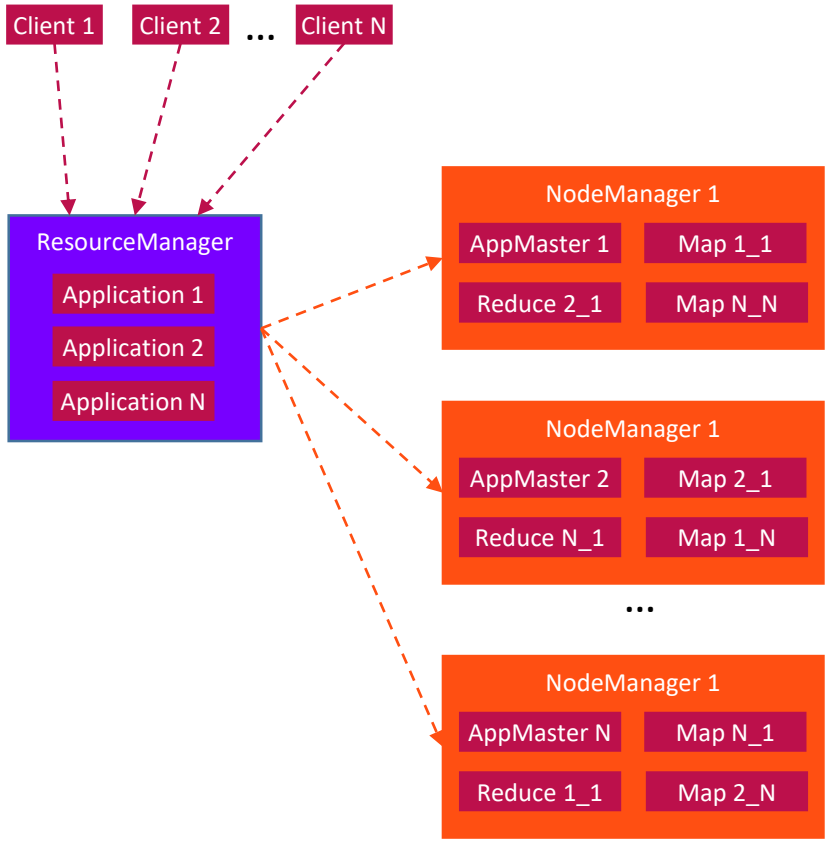
- Два основных компонента – NameNode и DataNode
- NameNode – аналог таблицы разметки традиционной ФС, содержит информацию о соответствии блоков данных файлам в HDFS, о фактическом размещении и состоянии блоков данных; данные в NameNode резервируются в Secondary NameNode (warm standby) или в Standby NameNode (hot standby)
- DataNode – содержит блоки данных; каждый блок реплицирован на несколько NameNode (на 3 по умолчанию); блоки имеют гораздо больший по сравнению с традиционными ФС размер (по умолчанию 64-128 МБ)
- За счет избыточного хранения блоков данных и репликации NameNode достигается высокая степень отказоустойчивости и доступности

Ядро Hadoop: YARN



- Два основных компонента – ResourceManager и NodeManager
- ResourceManager принимает запросы от клиентов, оркеструет выполнение приложений на кластере, следит за состоянием выполняемых приложений и доступными ресурсами кластера, перезапускает приложения в случае нештатных ситуаций и т.д.
- NodeManager – узел кластера, на котором фактически выполняются приложения; размещается вместе с HDFS DataNode для обеспечения локальности выполнения вычислений

Ядро Hadoop: MapReduce



- MapReduce – основной фреймворк распределенных вычислений в составе Hadoop
- Состоит из двух основных фаз:
  - Map – распределенная обработка входных данных: к каждой порции данных применяются одни и те же вычисления
  - Reduce – «свёртка» рассчитанных на шаге Map промежуточных данных: все промежуточные данные агрегируются согласно требуемому алгоритму
- По состоянию на 2018 год фреймворк в чистом виде практически не используется ввиду существенных ограничений алгоритма, сложности разработки приложений и скорости выполнения работы

Другие компоненты Hadoop

- Apache Hive – SQL-движок, позволяющий работать с кластером Hadoop как с реляционной СУБД (с учетом некоторых особенностей синтаксиса SQL); фактически преобразовывает SQL-запрос к данным в HDFS в цепочку MapReduce-приложений, выполняемых в YARN
- Apache HBase – NoSQL СУБД (key-value), работающая поверх HDFS; использует собственный эффективный механизм доступа к данным
- Apache Spark – фреймворк распределенных вычислений с отличной от MapReduce моделью исполнения; приложения Spark могут исполняться с помощью YARN над данными в HDFS
- Solr\ElasticSearch – основанные на Lucene движки полнотекстового поиска, могут обрабатывать и хранить данные в HDFS
- Oozie – инструмент импорта\экспорта данных в HDFS из реляционных СУБД

… и многие другие

Основные вендоры платформы Hadoop

Ростелеком

Hortonworks

- Одни из основателей и визионеров инициативы ODPi – попытки стандартизировать существующие дистрибутивы Hadoop
- Дистрибутив Hortonworks Data Platform ближе всего (среди двух других) к open-source версии Apache Hadoop
- Позиционирование – вендор платформы для IoT и потоковой загрузки и обработки данных

Cloudera

- Дистрибутив Cloudera Distribution of Hadoop (CDH) содержал множество проприетарных компонентов, что осложняло миграцию между дистрибутивами
- По состоянию на начало 2019 года дистрибутив Hadoop в чистом виде исчез, компания сменила вектор развития
- Позиционирование – вендор платформы для облачных вычислений, аналитики и машинного обучения
- Осенью 2018 года Cloudera и Hortonworks объявили о слиянии

MapR

- В дистрибутиве Hadoop вместо HDFS присутствовала собственная проприетарная файловая система, полностью совместимая с HDFS
- Ряд других компонентов платформы Hadoop также был заменён на проприетарные разработки компании
- Компания практически не представлена в России
- Позиционирование – вендор конвергентной платформы данных для аналитики и AI

Arenadata

- Единственная российская компания – вендор корпоративного дистрибутива Hadoop
- Дистрибутив Arenadata Hadoop соответствует спецификациям ODPi и полностью основан на open-source компонентах
- Тесная интеграция с другими продуктами: аналитической СУБД, in-memory СУБД и платформой потоковой загрузки и обработки данных
- Позиционирование – вендор единой корпоративной платформы данных

ARENADATA

Перспективы и будущее Hadoop

Ростелеком

- В мире ажиотаж вокруг платформы утихает
- Фреймворк MapReduce в чистом виде практически не используется, разработан ряд оптимизаций и альтернатив (Tez, Spark и т.д.)
- Большую популярность набирает фреймворк Apache Spark
- Основные концепции, заложенные в платформе, продолжают использоваться (распределенные вычисления и хранение, перемещение вычислений ближе к данным и т.д.), меняются лишь используемые инструменты
- Несмотря на падение популярности, платформа активно развивается:
    - HDFS 3.0 – новые эффективные механизмы репликации
    - YARN 3.0 – возможность работы сразу на нескольких кластерах
    - Развиваются инструменты обеспечения безопасности и управления данными
    - И т.д.

ARENADATA

Модуль 3. Hadoop ≠ Big Data. Другие технологии и инструменты обработки Больших Данных

- Какие задачи Big Data решают крупные компании и как они их решают?
- Можно ли по-прежнему обойтись одним SQL?
- Когда SQL уже недостаточно? Что такое NoSQL и NewSQL системы?
- Как работают платформы полнотекстового поиска?
- Можно ли хранить и обрабатывать Большие Данные в оперативной памяти?

ARENADATA

Какие задачи Big Data решают крупные компании?

Ростелеком

Хранение огромных массивов данных с real-time доступом

Транзакции и операции по счетам, обработка активных клиентских сессий

Классические задачи Data Warehouse, но на огромных объемах данных

Управленческая и прочая отчетность, объединение всей информации компании, аналитика

Анализ активности в Интернете

Аналитика над данными из соцсетей, анализ активности посетителей сайта, таргетинг рекламы, анализ настроений и трендов

Обработка данных в реальном времени

Оценки кредитоспособности потенциальных заёмщиков, биллинг абонента сотовой сети, формирование маркетинговых предложений в реальном времени

Поиск информации в массиве текста

Анализ документов, поиск ключевых слов, зависимостей и заимствований

IoT и четвертая индустриальная революция

Сбор и анализ сырых показаний сенсоров, предсказание поломок оборудования и брака продукции

Машинное обучение и AI

Распознавание изображений, видео и аудиопотоков, беспилотные аппараты, автоматический перевод текста, рекомендательные модели

ARENADATA

Можно ли по-прежнему обойтись SQL?

Ростелеком

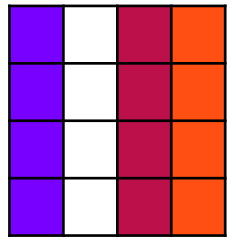
- Возможности современных реляционных СУБД позволяют решить часть задач Big Data:
  - Кластерные СУБД могут хранить и обрабатывать гигантские объемы данных, стало возможно горизонтальное масштабирование
  - Стала возможна обработка полуструктурированных данных, геоданных, возможен полнотекстовый поиск
  - В некоторых СУБД возможно запускать модели машинного обучения
  - Почти все современные СУБД enterprise-уровня интегрируются с Hadoop
- Однако не все проблемы можно решить одним лишь SQL:
  - Проблемы с хранением и обработкой бинарных объектов: аудио, видео, изображений
  - Сложно реализовывать потоковую загрузку и обработку: SQL – язык операций над множествами объектов, рассчитанный на batch-операции
- Вывод: SQL в эпоху Big Data всё ещё необходим, но для своего набора задач:
  - Массивно-параллельная обработка больших объемов структурированных данных
  - Интеграция с классическими BI-решениями и реляционными СУБД

ARENADATA

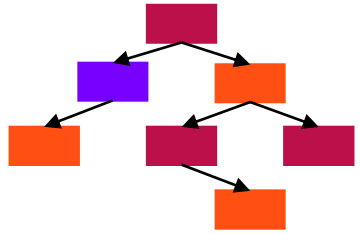
Что такое NoSQL и NewSQL?

- NoSQL (Not only SQL) – набор подходов к организации систем хранения данных, в которых за счет отказа от принятых в реляционных СУБД требований к атомарности и консистентности данных достигается масштабируемость и доступность данных
- Вместо требований ACID (Atomicity, Consistency, Isolation, Durability) традиционной транзакционной СУБД приходят свойства BASE:
  - Basic Availability – каждый запрос гарантированно завершается
  - Soft state – состояние системы может изменяться со временем, для достижения согласованности
  - Eventual Consistency – данные в системе становятся консистентными по прошествии определенного времени
- За счет отказа от реляционной модели хранения возможно множество типов организации информации:
  - Key-value: информация хранится в виде ассоциативных массивов. Примеры: Ignite, Redis, Aerospike
  - Документы: информация хранится в виде иерархических структур. Примеры: Couchbase, MongoDB
  - Графы: вместе с данными хранятся материализованные связи между ними. Примеры: Neo4j, Giraph
  - Семейства колонок: данные хранятся в виде разреженной матрицы. Примеры: HBase, Cassandra
- NewSQL – класс реляционных СУБД, в которых традиционные требования к транзакционности и атомарности (и поддержка SQL) совмещены с идеями организации хранения данных и масштабируемости NoSQL. Примеры: SAP HANA, VoltDB, OrientDB

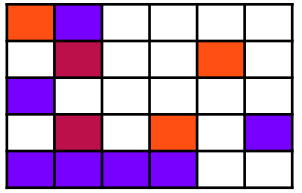
РСУБД



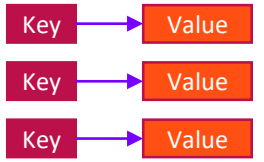
Документ

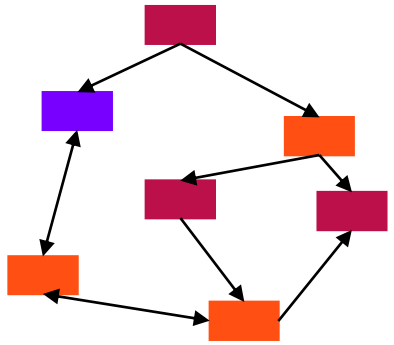


Семейство колонок



Ключ-значение





Граф

Как работают платформы полнотекстового поиска?

ID документа	Текст документа
1	To be is to be. To do is to do.
2	To be or not to be. I am what I am.
3	I think therefore I am. Do be do be do.
4	Do do do, da da da, let it be, let it be

Слово	Документ 1	Документ 2	Документ 3	Документ 4
To	X	X		
Do	X		X	X
Is	X			
Be	X	X	X	X
Or		X		
Not		X		
I		X	X	
Am		X	X	
What		X		
Think			X	
Therefore			X	
Da				X
Let				X
it				X

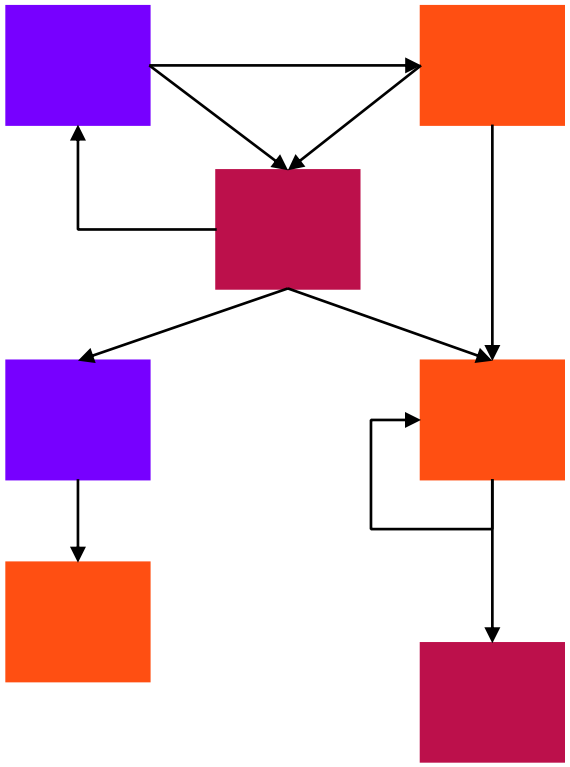
- Полнотекстовый поиск – процесс поиска документов по их содержимому
- Большинство современных алгоритмов формируют и используют для поиска полнотекстовый индекс – массив всех значимых слов и документов, в которых эти слова встречаются
- В общем случае полнотекстовый индекс – разреженная матрица слов и документов, в которых эти слова встречаются
- Для оптимизации задач поиска применяется множество различных техник и инструментов, например нечеткий поиск и ранжирование документов
- Примеры популярных поисковых систем:
  - Apache Lucene
  - Apache Solr
  - ElasticSearch
  - Sphinx

Обработка и хранение данных в оперативной памяти

С появлением 64-битных процессоров и снижением стоимости стало возможным использовать оперативную память как основное хранилище данных. Технологии, использующие RAM как основное хранилище, разделяются на следующие виды:

- In-memory надстройки над классическими реляционными СУБД – позволяют повысить производительность СУБД без её миграции за счет переноса части вычислений в RAM. Данные решения наследуют всё недостатки СУБД, в которых они реализованы, в т.ч. например, отсутствие масштабируемости и поддержка только возможностей классического SQL
- SQL/NewSQL СУБД – полностью размещают данные и вычисления в RAM и предоставляют поддержку SQL. При этом не всегда есть возможность горизонтального масштабирования или коллокации данных и вычислений. Примеры: VoltDB, SAP HANA
- In-memory grid – платформы для развертывания кластеров, обрабатывающих и хранящих данные больших объемов. Предоставляют наиболее полный набор возможностей для реализации полноценных in-memory платформ, но сложны в обслуживании и в разработке прикладного ПО, использующего платформу. Примеры: Apache Ignite, Hazelcast, Infinispan.

Модуль 4. Методики и алгоритмы обработки и хранения больших данных

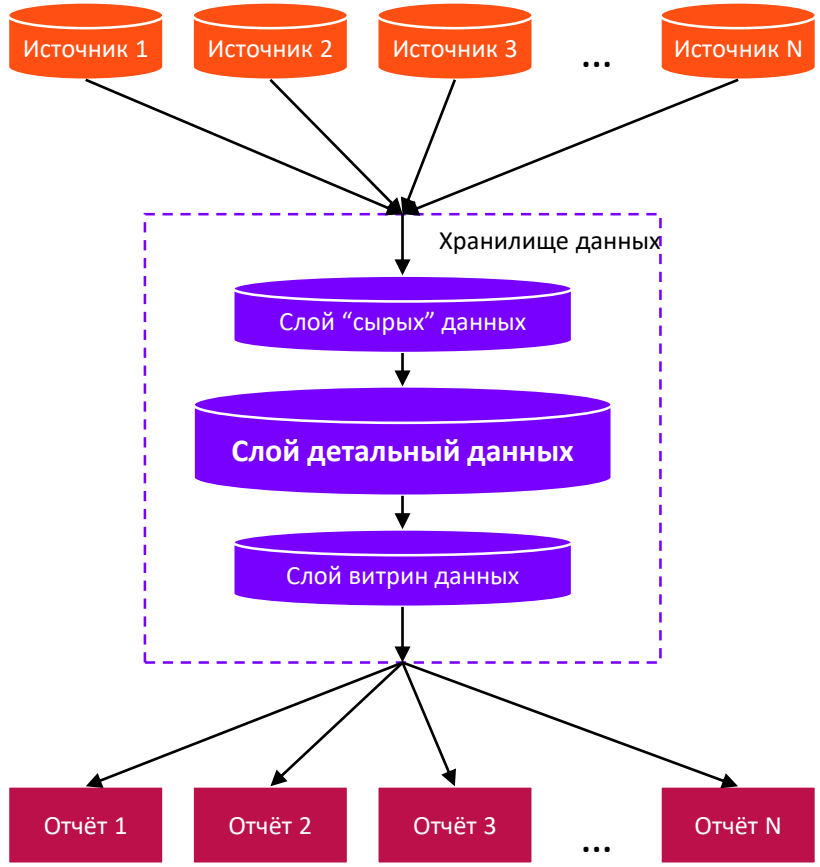


- Чем похожи и чем отличаются методологии построения корпоративного хранилища Data Lake и Data Warehouse? Каковы преимущества и недостатки подходов Schema-on-write и Schema-on-read, и когда нужно выбрать тот или иной? Что такое Data Virtualization и можно ли использовать этот метод в корпоративном хранилище?
- Какие методологии построения моделей данных корпоративного хранилища используются в современных системах? Чем похожи и чем отличаются “классические” подходы Б. Инмона и Р. Кимбалла и современные Data Vault и Anchor Modeling? Когда выбирать тот или иной подход, можно ли использовать “гибридные” методологии и до какой степени нужно нормализовывать данные?
- Что такое batch-, лямбда- и каппа-архитектуры обработки данных? В чем преимущества и недостатки каждой из них? Как определить, какая архитектура обработки будет оптимальной?
- Какие основные задачи решает Data Science? как построить платформу для обработки больших объемов данных с помощью алгоритмов машинного обучения и искусственного интеллекта? Что такое Deep Learning, и не является ли это исключительно маркетинговым термином?

ARENADATA

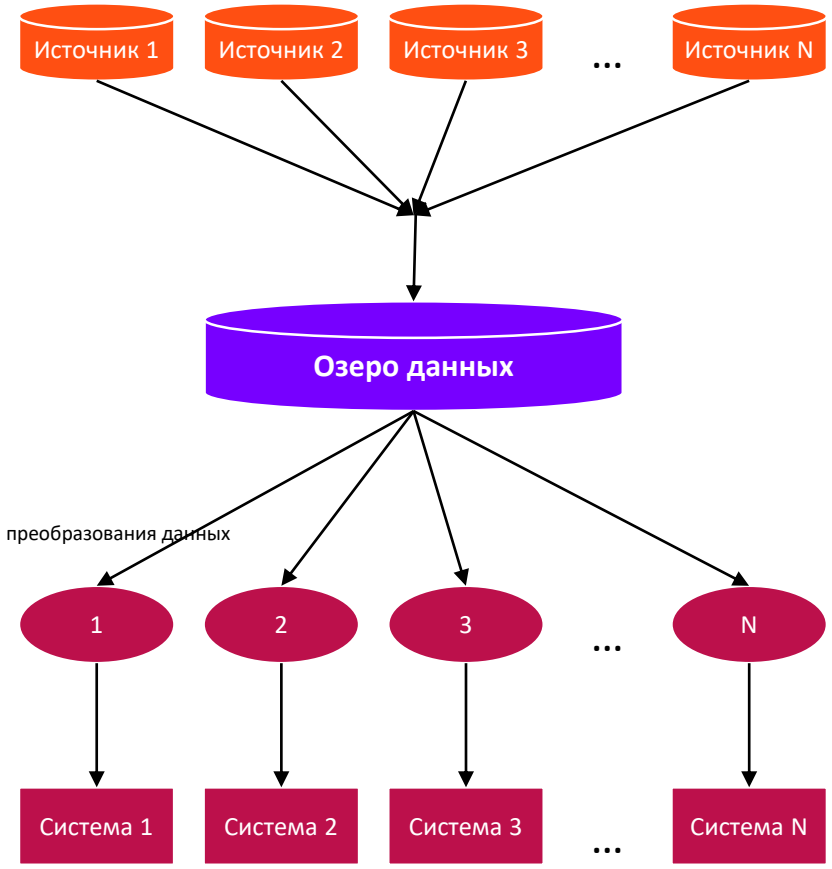
Data Warehouse

- Data Warehouse - **предметно-ориентированная** база данных, предназначенная для консолидации всей информации компании и служащая **единым источником правды** для бизнес-отчетности, систем поддержки принятия решений и аналитики
- Традиционно строится на основе реляционной СУБД
- Основные источники данных - реляционные СУБД с транзакционной информацией и прочие структурированные источники
- Основные потребители - бизнес-аналитики и системы бизнес-отчетности



ARENADATA

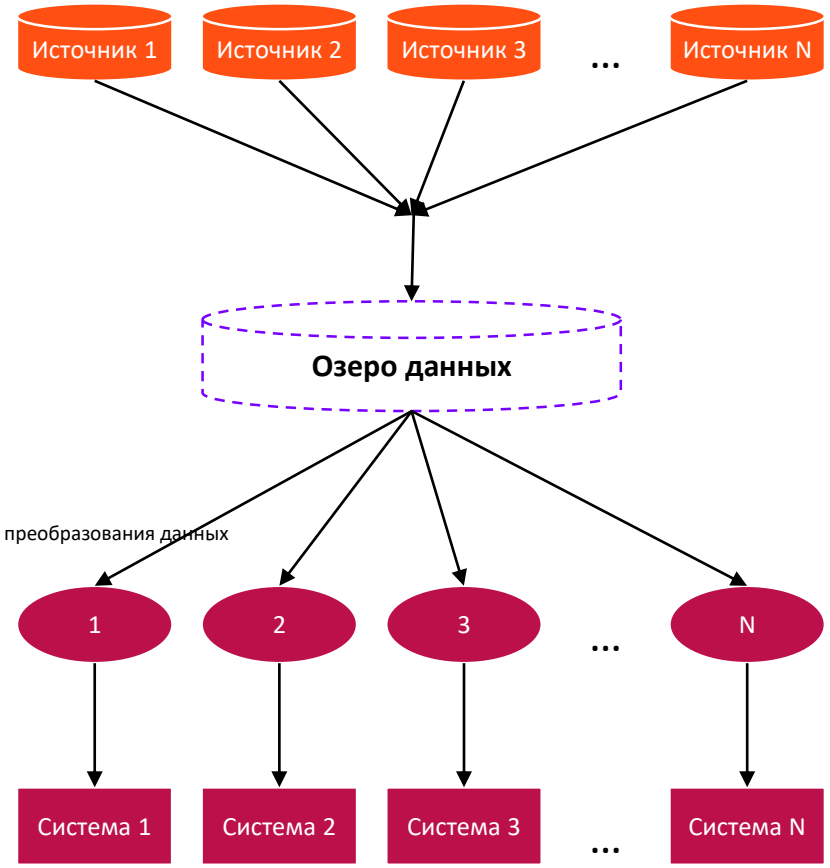
Data Lake



- Data Lake – особая модель и архитектура построения хранилища данных
- Строится на основе систем, позволяющих хранить и обрабатывать как структурированные, так и полуструктурированные и неструктурированные данные
- Источники данных – в идеале **все** источники данных предприятия
- Данные извлекаются и сохраняются в Data Lake **как есть**, без преобразований и фильтрации
- Данные преобразовываются только на этапе их чтения их Data Lake непосредственно перед использованием потребителями
- Основные потребители – аналитики и системы анализа данных

Data Virtualization

- Data Virtualization - дальнейшее развитие концепции Data lake
- Основное отличие - данные из систем-источников **не копируются** в отдельную область (собственно Data Lake), а **остаются в источниках**
- Фактическое извлечение и трансформация данных происходят **только в момент запроса** от системы-потребителя
- **Важно**: создает большую нагрузку на системы-источники. Не рекомендуется использовать данный подход в предприятиях, для которых такая нагрузка недопустима (пример: системы, обрабатывающие транзакции по картам и операции по счетам в банках), или если системы-источники не сохраняют исторические данные



Data Warehouse vs Data Lake vs Data Virtualization

Ростелеком

Data Warehouse

- Содержит отфильтрованные, очищенные и преобразованные данные
- Содержит только те данные, которые используются в настоящий момент для удовлетворения текущих потребностей предприятия
- Основные потребители: бизнес-пользователи, системы отчетности и поддержки принятия решений
- Доработки и модификации компонентов хранилища сложны, долги и дороги

Data Lake

- Содержит данные в том виде, в котором они были получены из источников
- Содержит все данные, которые можно извлечь из всех доступных источников
- Основные потребители: Data Scientists и аналитики данных
- Развитие и модификация хранилища происходит быстро и с небольшими затратами
- Каждый потребитель данных должен самостоятельно определять механизмы преобразования и очистки данных хранилища под свои нужды

Data Virtualization

- Не содержит собственно данных, а только информацию об источниках, содержащихся в них данных и механизмах их извлечения
- Создает большую нерегулярную нагрузку на системы-источники
- Системы-источники могут не хранить необходимые исторические данные
- Прямой доступ к источникам может быть нарушением правил безопасности

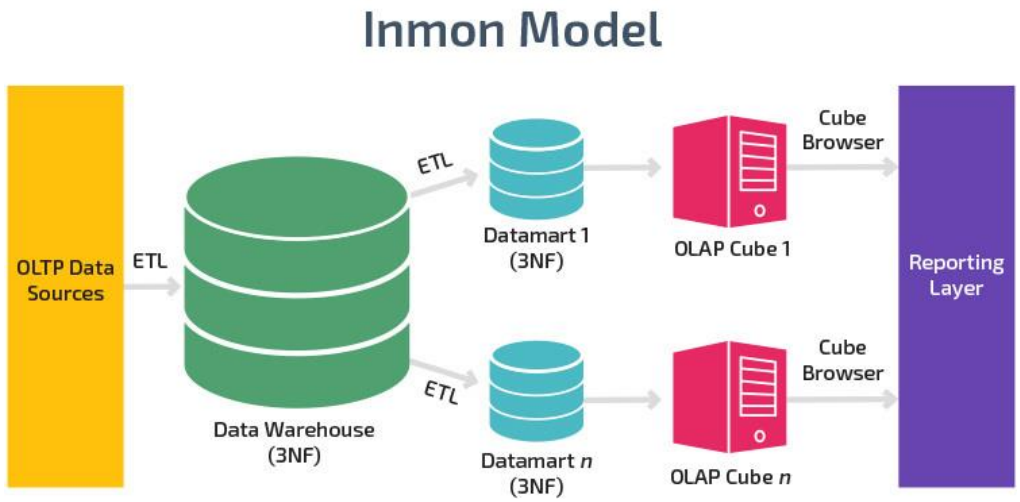
ARENADATA

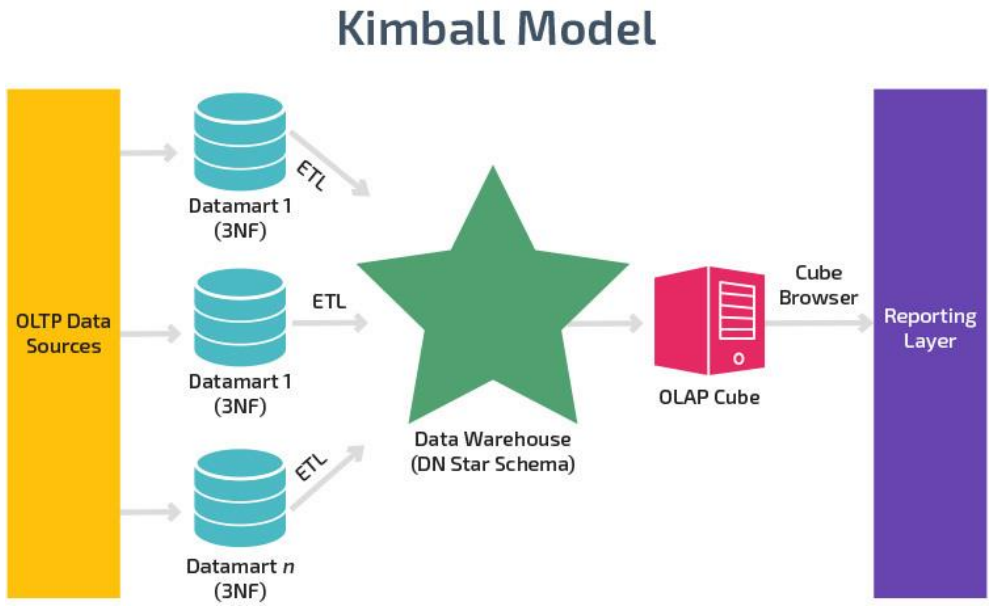
Классические подходы к построению модели данных

Ростелеком

Подход Билла Инмона (top-down)

- Модель данных строится на основе требований к конечным витринам, на их основе строится модель ядра хранилища
- Ядро хранилища - слой детальных данных - таблицы преимущественно в третьей нормальной форме





Подход Ральфа Кимбалла (bottom-up)

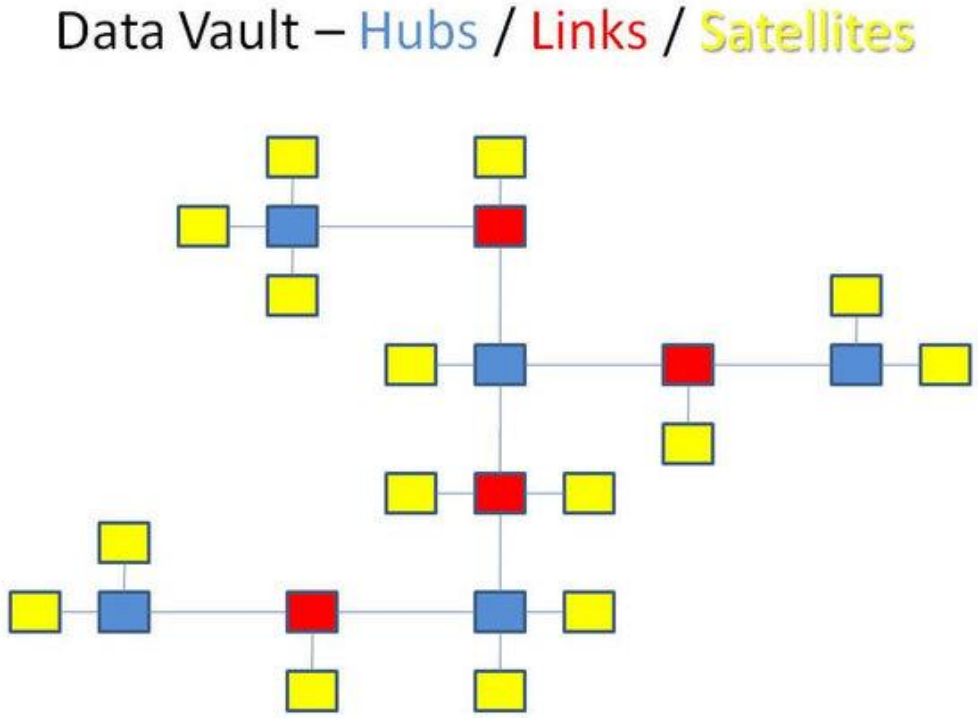
- Модель данных ядра хранилища строится на основе объединения моделей данных всех необходимых отчётов и витрин
- Итоговая модель ядра хранилища - star- или snowflake-схема

ARENADATA

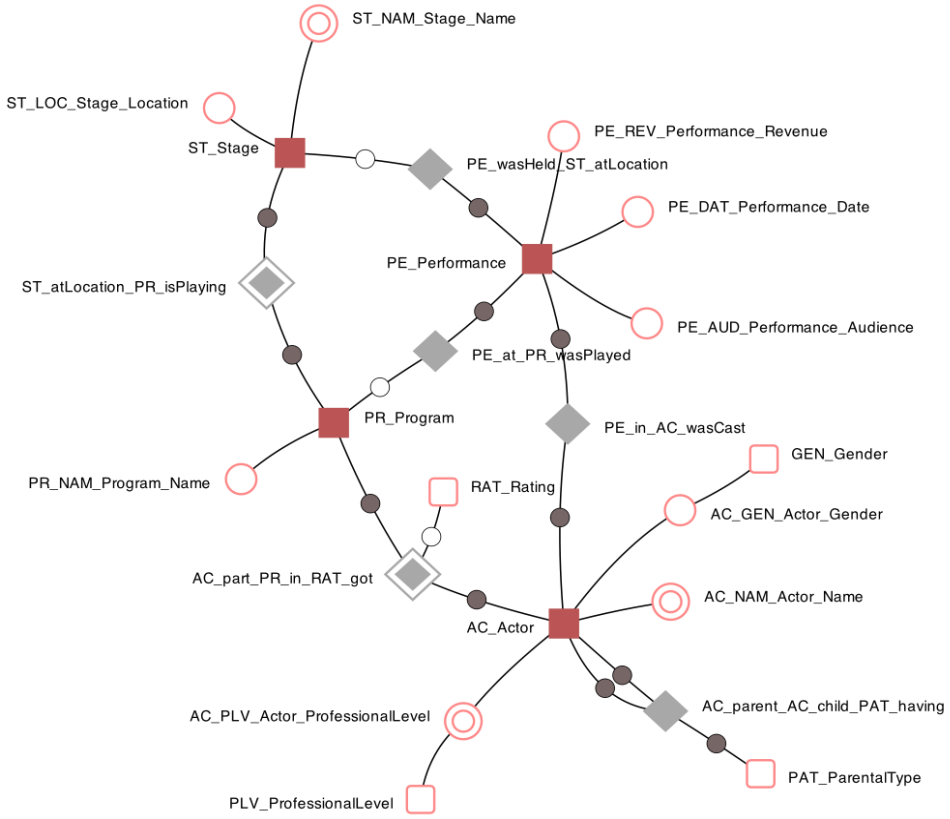
Data Vault

Data Vault - дальнейшее развитие идей, заложенных в обоих подходах:

- Всего три сущности модели данных: Hub, Link, и Satellite
- Hub - основное представление бизнес-сущности (Клиент, Счёт, Договор и т.д.)в модели данных. Содержит только первичный ключ сущности и технические поля
- Link - таблица связи между двумя или более Hubs, содержит только их первичные ключи и технические поля
- Satellite - атрибуты бизнес-сущности, содержит первичный ключ бизнес-сущности, один или несколько атрибутов и технические поля



Anchor Modeling



Anchor Modeling - самый новый подход, в котором идеи Ральфа Кимбалла и Data Vault доведены до предела:

- Все таблицы хранилища находятся в 6 нормальной форме (т.е. дальнейшая нормализация невозможна)
- Всего 4 типа таблиц:
  - Якорь (Anchor) - таблица с единственным полем - ключом бизнес-сущности
  - Атрибут (Attribute) - таблица с ключом бизнес-сущности и единственным полем со значением атрибута. Если допускается изменение значения атрибута со временем, содержит также поле с датой начала действия данной версии значения
  - Связь (Tie) - таблица с двумя или более ключами бизнес-сущностей. Если допускается изменение связи со временем, содержит также поле с датой начала действия данной версии связи
  - Узел (Knot) - аналог справочника, содержит ключ сущности справочника и значение атрибута справочника.

ARENADATA

3NF vs Dimensional Modeling vs Data Vault vs Anchor Modeling

Ростелеком

3NF (Inmon)

+Низкая избыточность данных, более простые ETL-процессы загрузки хранилища

+Модель данных поначалу легко сопровождать и менять

-Со временем сопровождение выросшей модели становится сложным

-Долгий процесс первичного развертывания хранилища

-Более сложные ETL-процессы наполнения витрин

Dimensional Modeling (Kimball)

+Быстрое первичное развертывание хранилища

+Более простая в сопровождении модель данных

-Некоторые данные хранятся избыточно

-Сложные процессы изменения структуры таблиц фактов

-Сложные процессы загрузки legacy-данных в хранилище

Data Vault

+Высокая степень параллелизма загрузки данных в хранилище

+Простая концепция построения модели данных, которую просто развивать

-Несмотря на простую концепцию, само построение модели может быть сложной задачей с несколькими решениями

-Низкая производительность построения витрин данных из-за сильно нормализованной структуры данных

Anchor Modeling

+Модель данных легко развивать: практически любое изменение сводится к добавлению или удалению таблиц

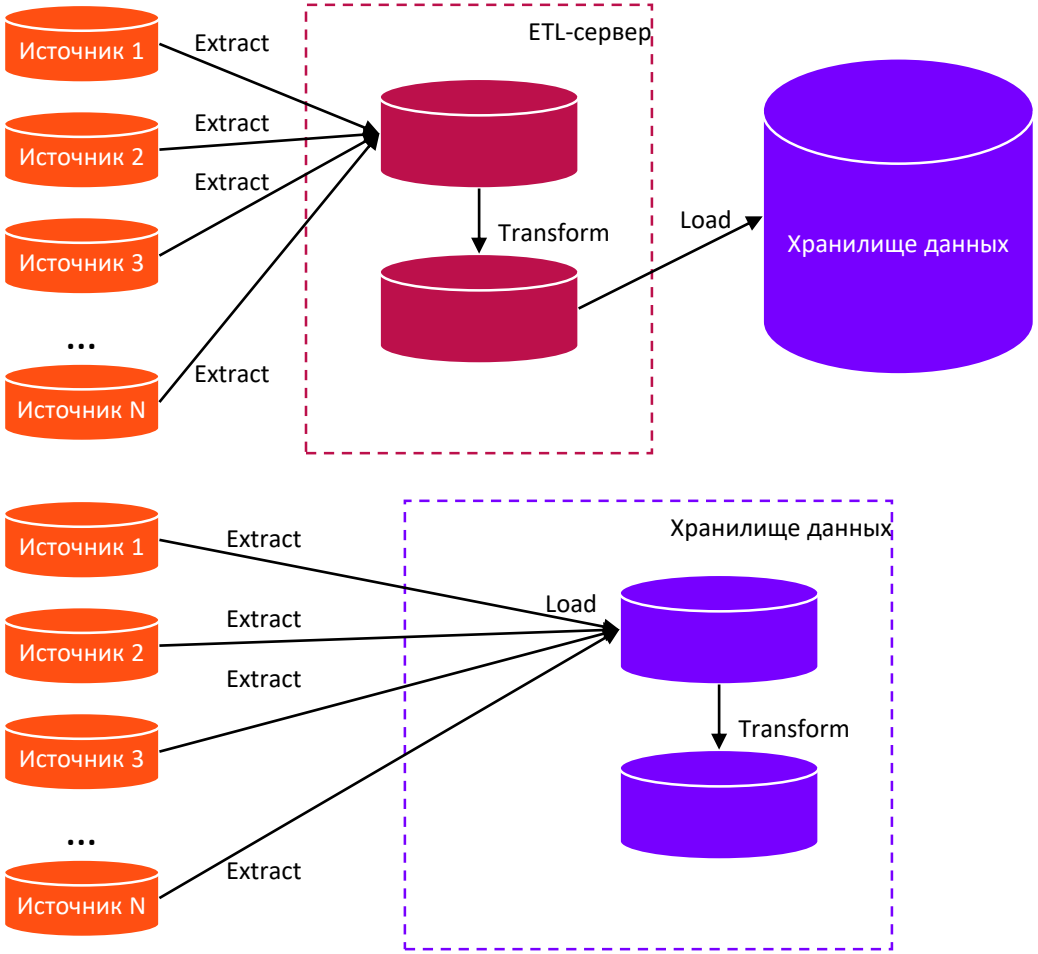
+Высокая степень параллелизма загрузки

-Ещё более сложное, чем в Data Vault, первичное создание и сопровождение модели данных из-за огромного числа таблиц

-Очень низкая производительность построения витрин данных из-за большого числа JOIN

ARENADATA

Классические подходы к загрузке данных: ETL и ELT

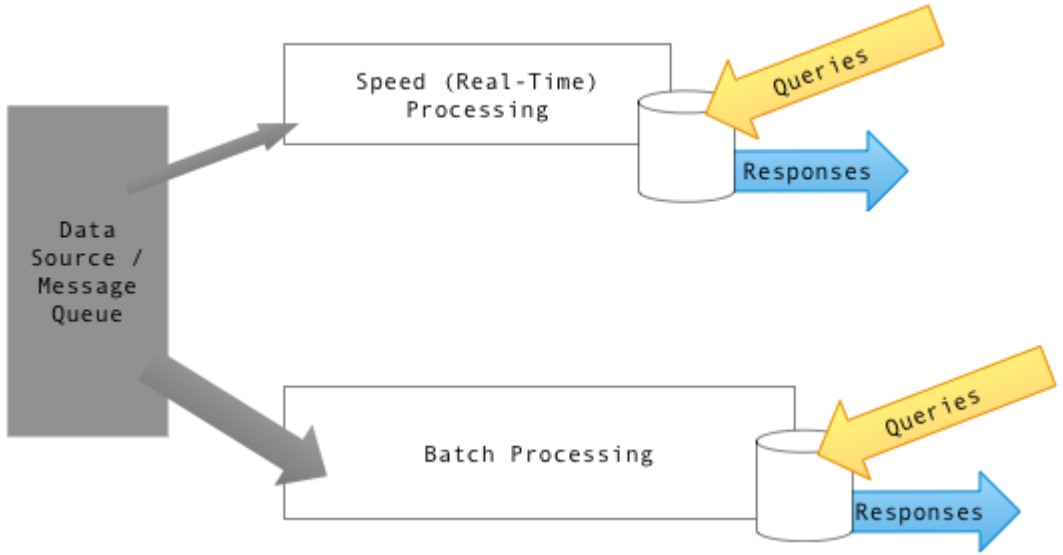


ETL (Extract, Transform, Load) и ELT (Extract, Load, Transform) – классические подходы к загрузке данных в хранилище, состоящие из этапов извлечения (Extract) данных из источников, преобразования (Transform) их согласно правилам и модели данных хранилища, и загрузки (Load) из в хранилище.

Отличие ETL и ELT – в порядке этапов. в ETL требуется промежуточная область вне хранилища, в которой данные будут преобразованы перед загрузкой, в ELT данные сначала загружаются в промежуточный слой хранилища, после чего преобразуются уже внутри хранилища.

ARENADATA

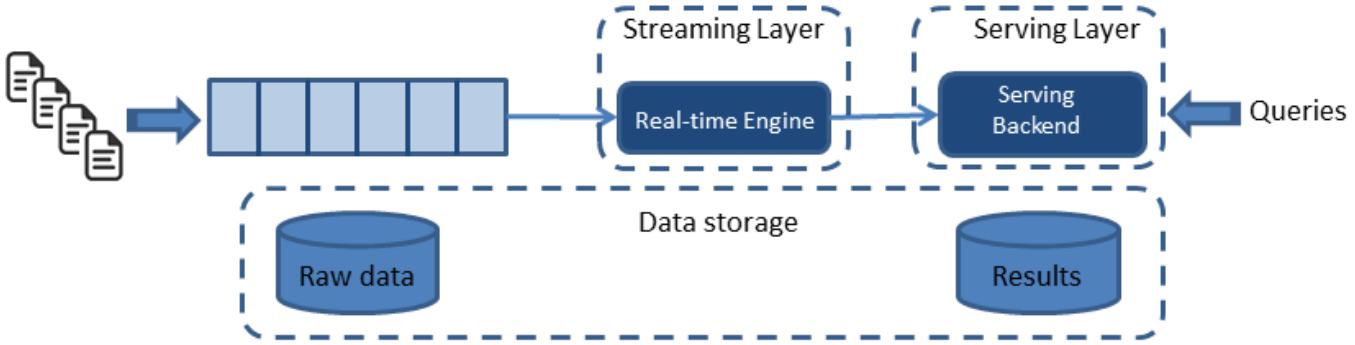
Лямбда-архитектура загрузки данных



Лямбда-архитектура – гибридный подход к загрузке и обработке данных

- Данные загружаются в два потока
- Данные, поступающие в реальном времени, загружаются и обрабатываются в реальном времени
- Согласно расписанию выполняется batch-обработка всех данных хранилища

Каппа-архитектура загрузки данных



Каппа-архитектура – загрузка и обработка данных в real-time или near-real-time

- Данные загружаются в реальном времени as is в исторический слой хранилища
- В параллельном потоке данные обрабатываются в реальном времени и через специальный слой (serving layer) поступают в соответствующие системы-приёмники (детальный слой хранилища, витрины, системы отчетности и т.д.)
- При необходимости пересчёта данные перегружаются из исторического слоя

ARENADATA

ETL vs ELT vs Лямбда-архитектура vs Каппа-архитектура

Ростелеком

ETL

+Самый простой подход к загрузке данных, не требует дополнительных слоёв хранилища и дополнительной нагрузки на хранилище

-Требует выделенного ETL-сервера, на котором будет происходить преобразование данных, и который может стать узким местом при росте объема данных

-Только batch-загрузка

ELT

+Более совершенный по сравнению с ETL подход

+Не требует выделенного ELT-сервера, все преобразования производятся внутри хранилища

-Дополнительная высокая нагрузка на хранилище в ходе преобразования загруженных данных

-Только batch-загрузка

Лямбда-архитектура

+Гибрид batch- и real-time-подхода

+Доступны как исторические данные хранилища, точно сформированные в ходе batch-загрузки, так и свежие данные, загруженные real-time-потоком

-Необходимость сопровождать и разрабатывать двойной объём потоков загрузки и обработки

-Данные, загруженные real-time-процессами, могут быть обработаны не с такой же точностью, как в batch

Каппа-архитектура

+real-time загрузка и обработка данных

-При изменении процесса загрузки все рассчитанные данные в хранилище необходимо пересчитывать заново

-Ограничения на процессы загрузки, связанные с требованием real-time

ARENADATA

Data Science и AI в корпоративном ландшафте

Ростелеком

- Все алгоритмы машинного обучения, Data Science и AI используют данные - следовательно, чем лучше данные будут доступны, тем точнее будут работать алгоритмы. Вот почему так важно правильно выстроить процессы загрузки и обработки данных.
- Алгоритмы машинного обучения и AI могут выступать не только как потребители ETL-процессов, но и как их часть, причем не только как инструмент преобразования данных (например, рассчитать значение показателя по известной модели), но и как инструмент извлечения данных (например, на основе модели определить, какие данные необходимо извлечь из источника, а какие можно отфильтровать)
- Искусственный интеллект может использоваться для оптимизаций модели данных и параметров хранилищ данных

ARENADATA

Модуль 5. Безопасность больших данных и управление большими данными

Ростелеком

- Что такое безопасность данных, как и на каких уровнях она обеспечивается?
- Что такое управление данными (Data Governance)?

ARENADATA

Безопасность Больших данных

Ростелеком

Перемещение данных

Шифрование каналов и протоколов связи:

- HTTPS
- SSL/TLS
- SSH
- SFTP и др.

Хранение данных

Шифрование хранимых данных:

- На уровне приложения
- На уровне БД/системы хранения
- На уровне файловой системы
- На уровне дисковой подсистемы и др.

Доступ к данным

Обязательная аутентификация конкретного пользователя или приложения:

- LDAP
- Kerberos
- IDM-системы и др.

Обязательная авторизация конкретного пользователя или приложения:

- ACL
- Политики доступа и др.

ARENADATA

Управление данными (Data Governance)

Data Governance - ряд методологий и рекомендаций, направленных на создание на предприятии бизнес-процесса управления данными как активом предприятия.

Методы управления данными включают в себя:

- Управление жизненным циклом данных
- Управление архитектурой данных
- Управление качеством данных
- Управление безопасностью данных
- Управление нормативно-справочной информацией
- Управление мастер-данными

И многое другое

ARENADATA

Модуль 6. Прочие вопросы построения платформ обработки и хранения больших данных

Ростелеком

- Что такое on-premise, облачные и гибридные решения развертывания платформы, в чём их преимущества и недостатки?
- Что такое модели облачных сервисов, чем отличаются IaaS, PaaS, SaaS, DaaS?
- Как выбирать и использовать open source технологии и решения, когда и в каких случаях не стоит использовать open source; преимущества и недостатки in-house разработки
- CAP- и PACELC-теоремы и другие теоретические проблемы распределенных вычислений

ARENADATA

Размещение платформы: on-premise, облако, гибрид

Ростелеком

On-premise

+Платформа данных разворачивается полностью на мощностях, принадлежащих предприятию

+Безопасность ценных данных и соответствие регламентам безопасности предприятия

-Высокая стоимость закупки и поддержки оборудования

-Необходимо наличие штата квалифицированных инфраструктурных специалистов

-Может быть достигнут предел производительности оборудования без возможности быстрого наращивания мощностей

Облако

+Нет необходимости закупать дорогостоящее оборудование и нанимать дополнительный персонал

+Масштабируемость мощностей оборудования по запросу с минимальной задержкой

-Итоговая стоимость использования оборудования может оказаться существенно выше, чем полная стоимость владения собственным оборудованием

-Возможны риски утечки данных

Гибридная модель

+Возможность разграничить хранение и обработку ценных данных и общедоступных

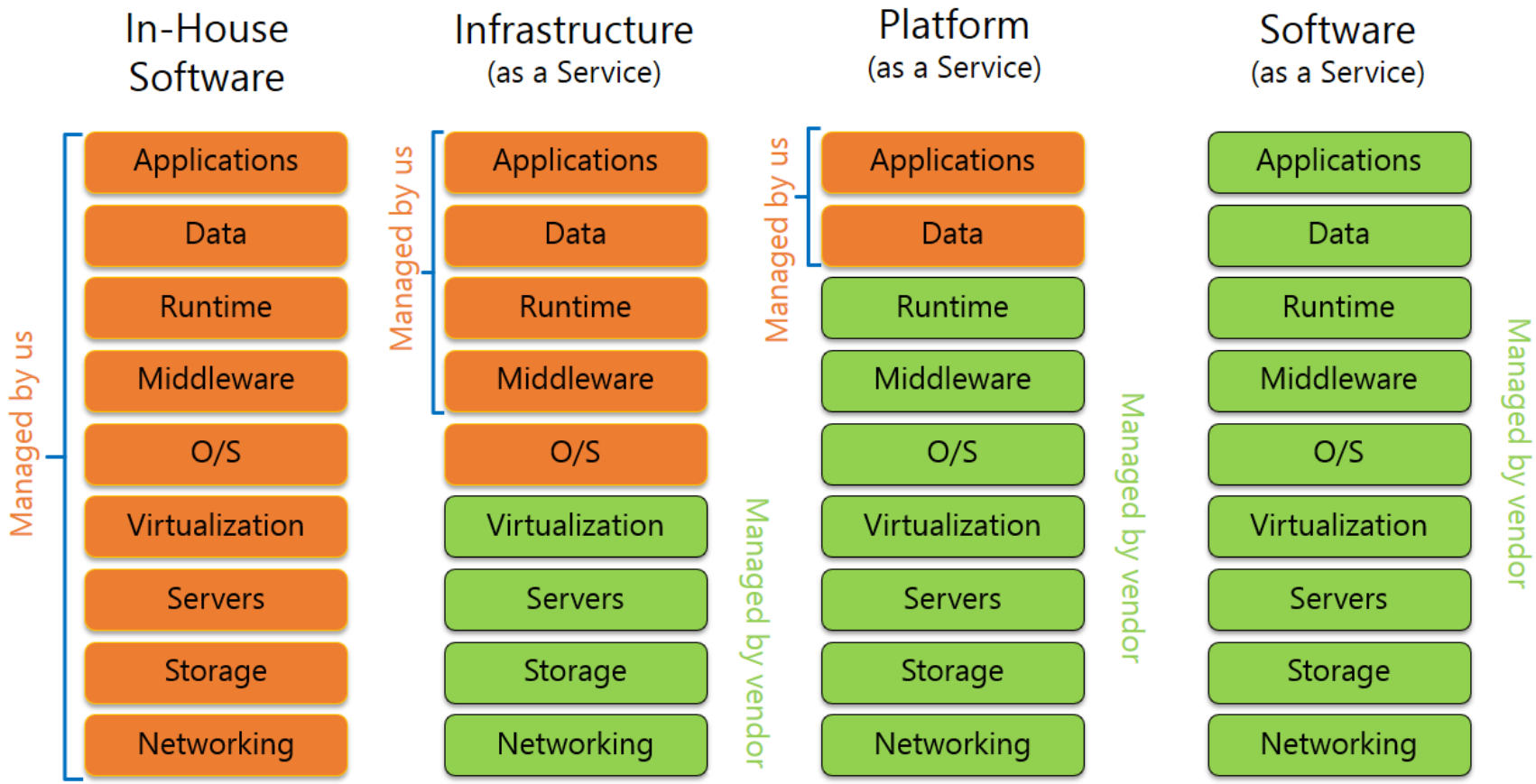
+Гарантированные мощности оборудования on-premise и масштабируемость облачного решения

-Высокая суммарная стоимость владения собственным оборудованием и использования облачного

-Высокая сложность интеграции и сопровождения гибридного решения

ARENADATA

Модели предоставления облачных услуг



ARENADATA

Выбор ПО: open source, enterprise, in-house

Ростелеком

Open Source

+Полностью открытое ПО, которое возможно дорабатывать под свои нужды

+Большое сообщество разработчиков и специалистов, использующих данное ПО

+У многих open source проектов есть компания вендор, предоставляющая техническую поддержку

-Open Source ≠ бесплатное ПО

-Не гарантированы стабильность работы и отсутствие багов

Enterprise

+Гарантированно работающие продукты, прошедшие проверку временем

+Техническая поддержка вендора

-Очень высокая стоимость лицензий и технической поддержки

-Vendor lock-in

In-house

+Возможность разработать продукт, полностью отвечающий требованиям предприятия

-Необходимость нанимать штат квалифицированных разработчиков

-Может уже существовать open source или enterprise ПО, полностью соответствующее требованиям или требующее минимальных доработок

ARENADATA

Заблуждения о распределенных вычислениях

Ростелеком

1. Сеть надёжна
2. Задержка передачи данных нулевая
3. Пропускная способность сети бесконечна
4. Сеть безопасна
5. Топология сети никогда не меняется
6. Администратор всегда только один
7. Цена передачи данных нулевая
8. Сеть однородна

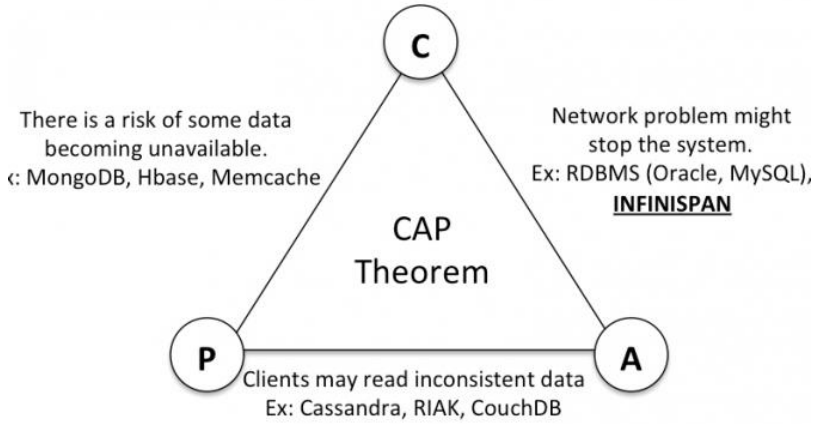
ARENADATA

CAP-теорема

Ростелеком

В любой реализации распределенных вычислений можно обеспечить не более двух из трех свойств:

- Consistency (согласованность данных) - во всех вычислительных узлах в один момент времени данные не противоречат друг другу
- Availability (доступность) - на любой запрос к распределенной системе гарантированно возвращается корректный отклик, без гарантии совпадения ответов от всех узлов системы
- Partition tolerance (устойчивость к разделению) - расщепление распределённой системы на несколько изолированных секций не приводит к некорректности отклика от каждой из секций



ARENADATA

PACELC-теорема

PACELC-теорема - расширение CAP-теоремы:

- В случае разделения (**P**artition) сети нужно выбирать между
  - Доступностью (**A**vailability) и
  - Консистентностью (**C**onsistency)
- Иначе (**E**lse), если разделения сети нет, нужно выбирать между
  - Задержками (**L**atency) и
  - Консистентностью (**C**onsistency)