Hadoop

HDFS, MapReduce, Hive

Обо мне

Егор Матешук, Data Engineer

@ Ostrovok.ru

@ МаксимаТелеком

@ Квант

@ ГПМ Дата

Зачем это занятие

После этого занятия вы будете знать

- Что такое Hadoop
- Из каких основных сервисов он состоит
- Как использовать эти сервисы

План

- Что такое Hadoop
- Zookeeper
- HDFS
- YARN
- MapReduce
- Hive

Что такое Hadoop

Hadoop

Под словом Hadoop могут подразумеваться

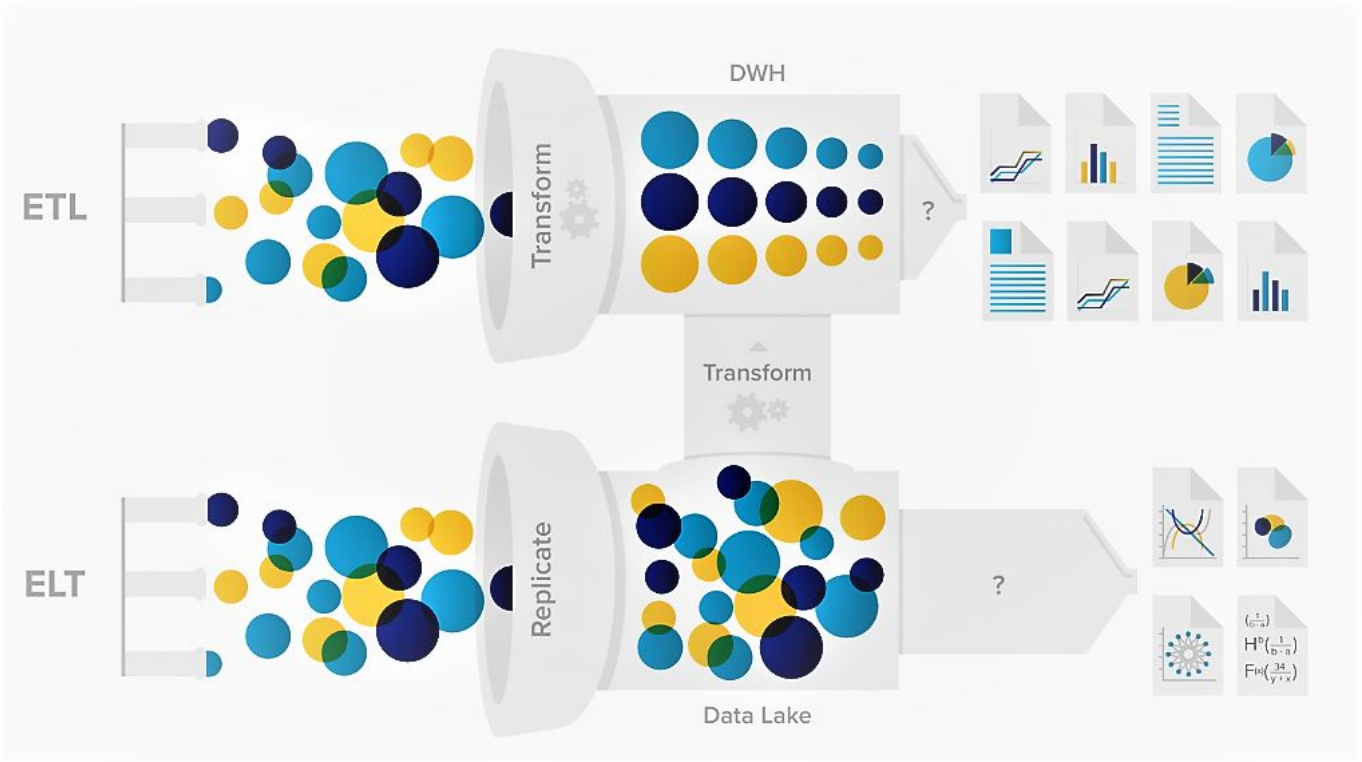
- Несколько сервисов, составляющих “ядро” Hadoop
- Вся экосистема сервисов Hadoop
- Кластер под управлением Hadoop

Предпосылки Hadoop

Предпосылки появления Hadoop

- Потребность в распределенных хранилищах
- Масштабирование вычислений
- Управление ресурсами

Data Lake



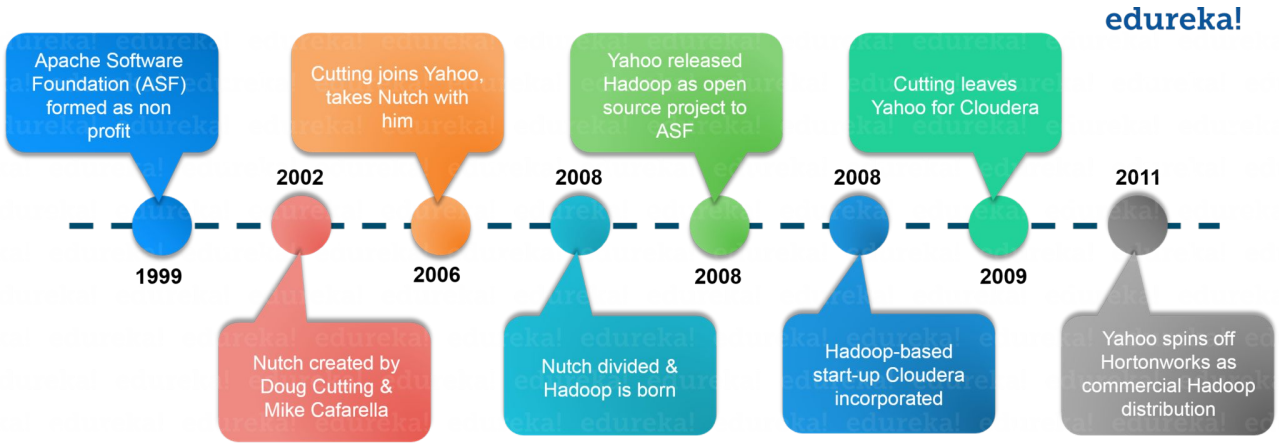
История Hadoop

- 2003 - Google File System paper
- 2004 - MapReduce paper
- 2006 - Hadoop project
- 2008 - Pig, Hive, HBase
- 2009 - Amazon EMR

Дистрибутивы Hadoop

Появление “спецов по Hadoop”

- 2008 - создание Cloudera
- 2011 - выделение Hortonworks



Ядро Hadoop

Hadoop v1.0

MapReduce
Data Processing
& Resource Management

HDFS
Distributed File Storage

Hadoop v2.0

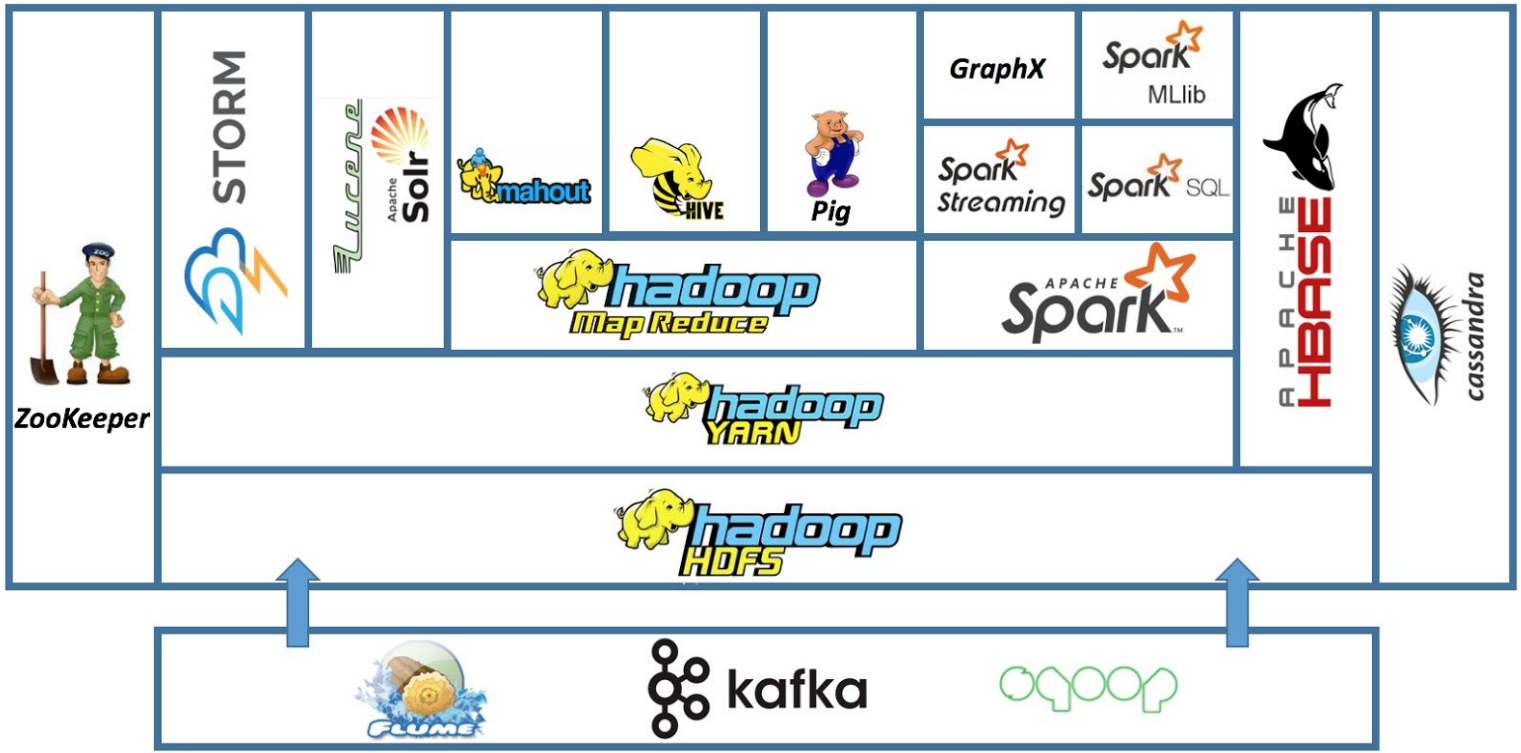
MapReduce

Other Data Processing Frameworks

YARN
Resource Management

HDFS
Distributed File Storage

Экосистема Hadoop

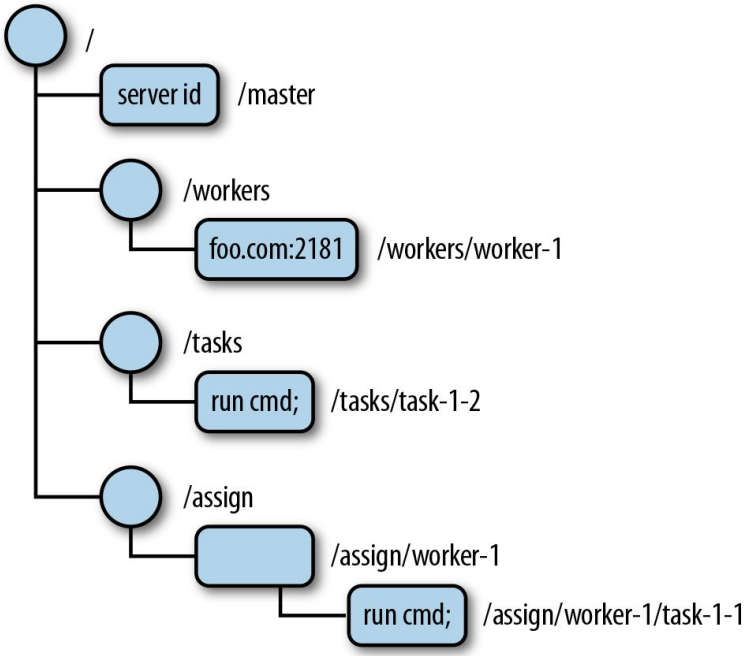


Zookeeper

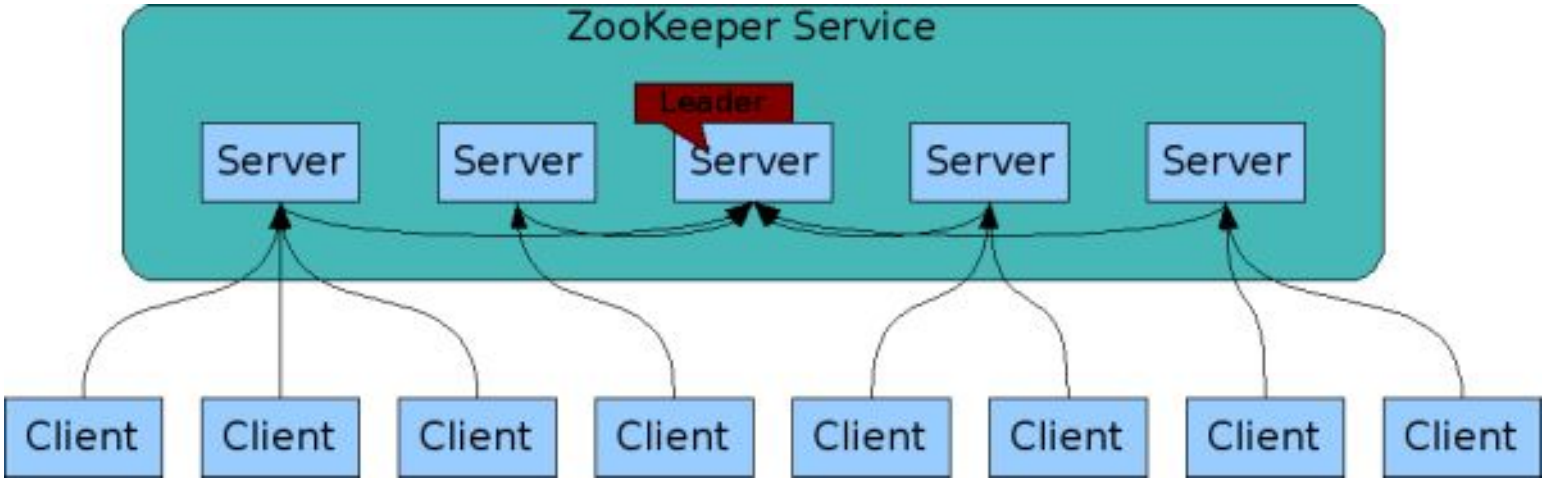
Zookeeper

Apache ZooKeeper - это централизованная служба для поддержки информации о конфигурации, именования, обеспечения распределенной синхронизации и предоставления групповых служб.

Zookeeper


/
server id
/master
/workers
foo.com:2181
/workers/worker-1
/tasks
run cmd;
/tasks/task-1-2
/assign
/assign/worker-1
run cmd;
/assign/worker-1/task-1-1

Zookeeper

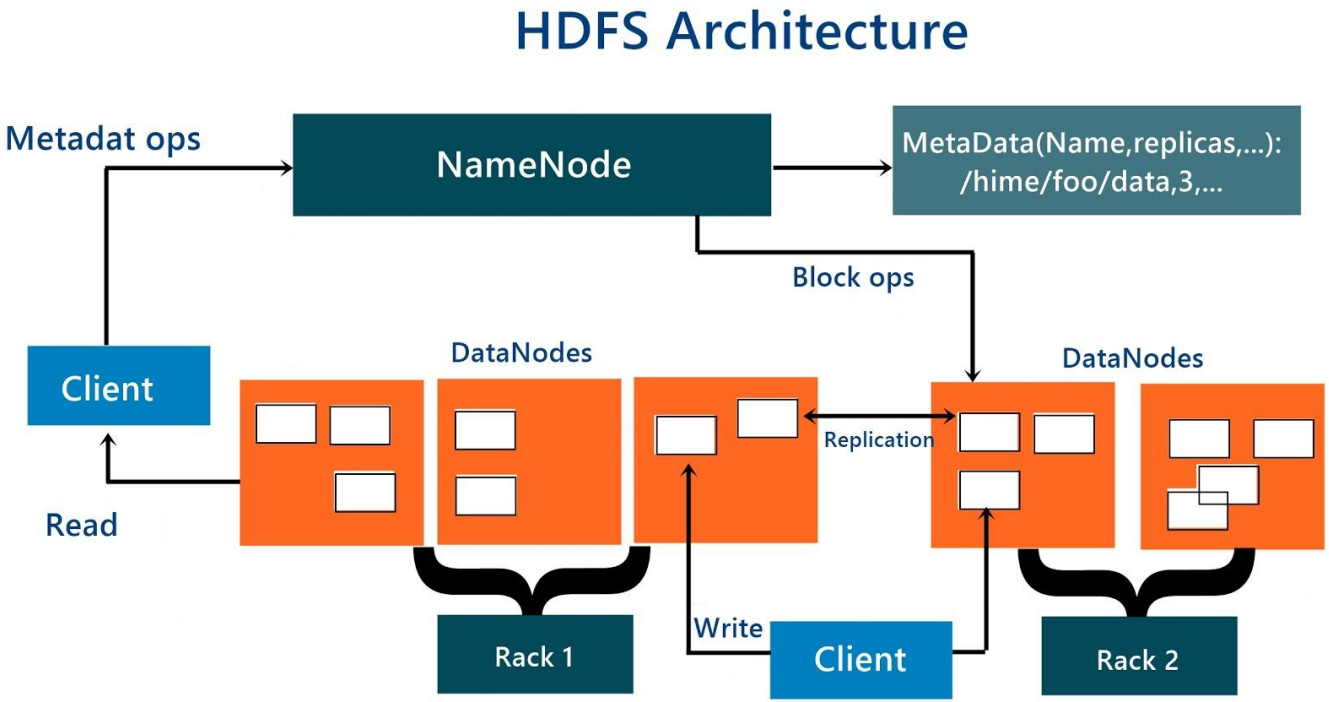

ZooKeeper Service
Leader
Server
Server
Server
Server
Server
Client
Client
Client
Client
Client
Client
Client
Client

HDFS

HDFS

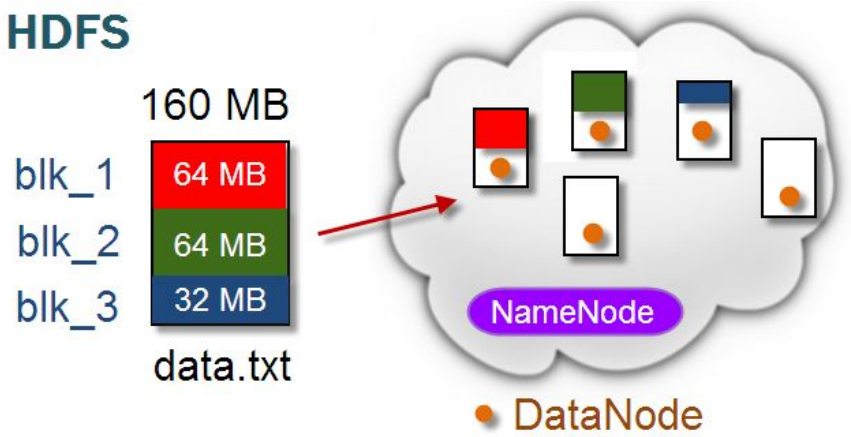
HDFS (Hadoop Distributed File System) — файловая система, предназначенная для хранения файлов больших размеров, поблочно распределённых между узлами вычислительного кластера.

Архитектура HDFS

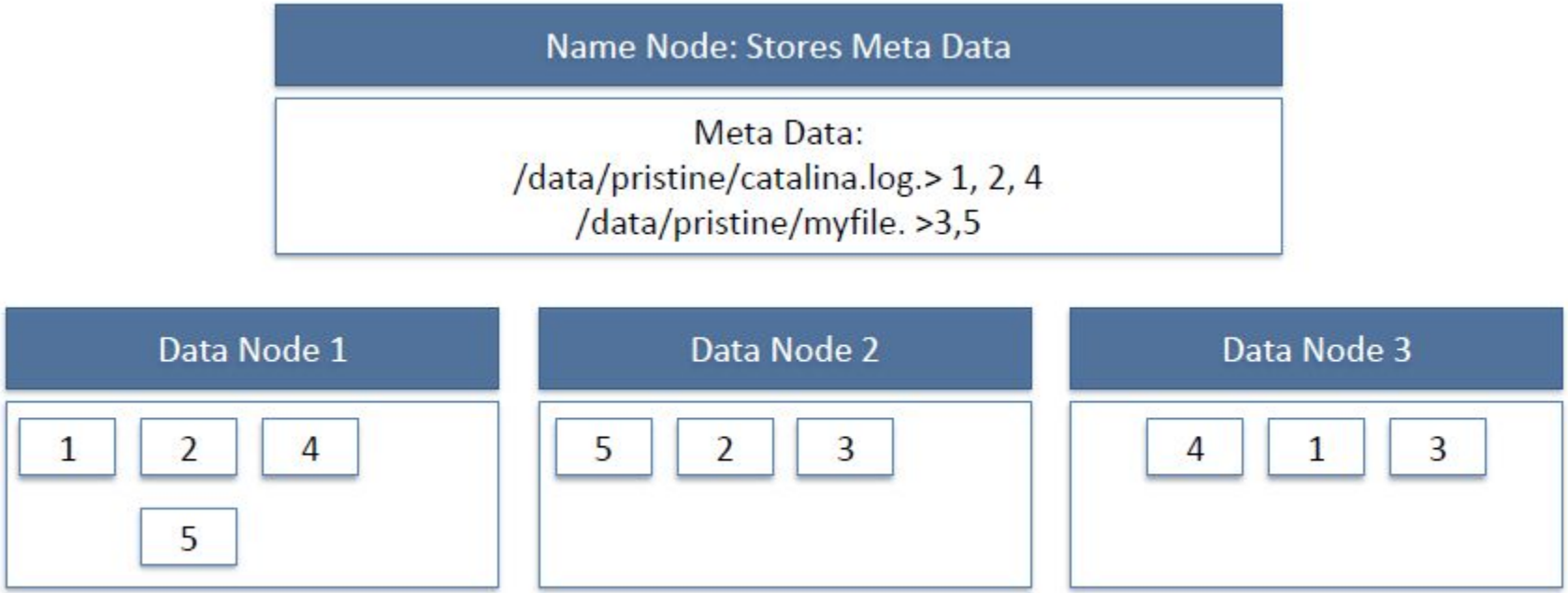


Блоки и файлы

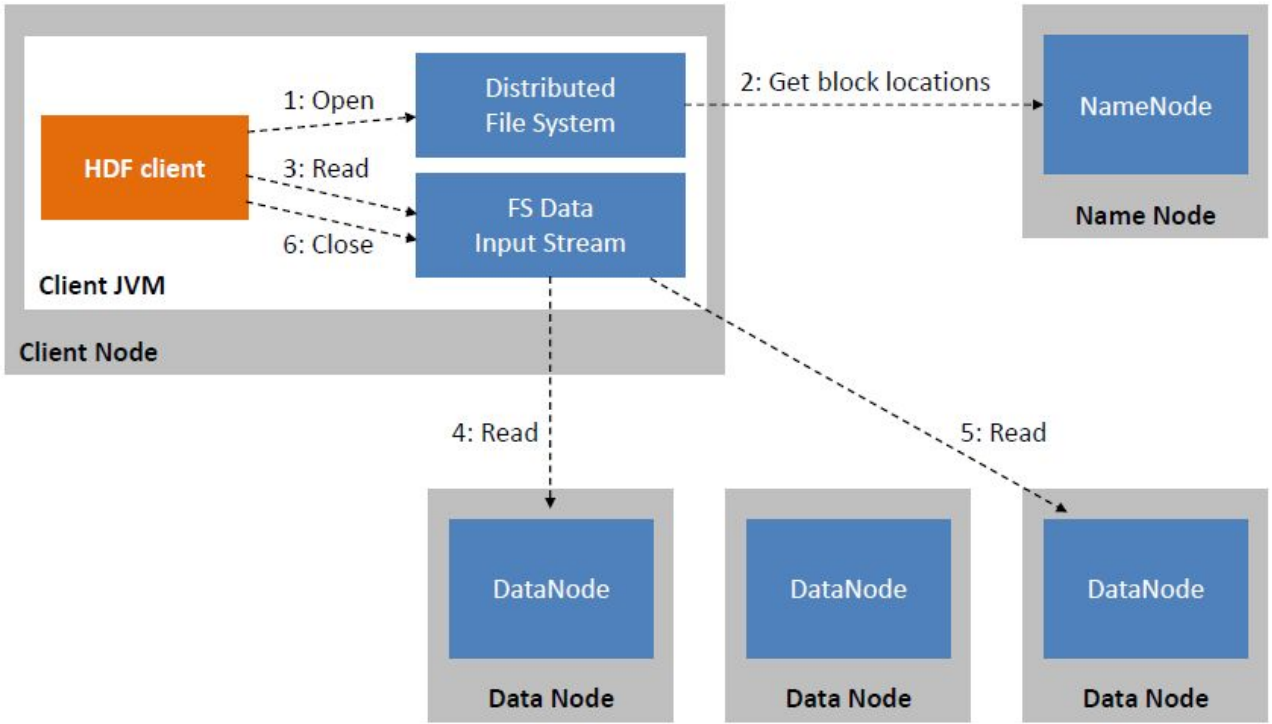
- **Файл** - это только запись в метаданных
- Содержимое файла хранится в нескольких **блоках** одинакового размера



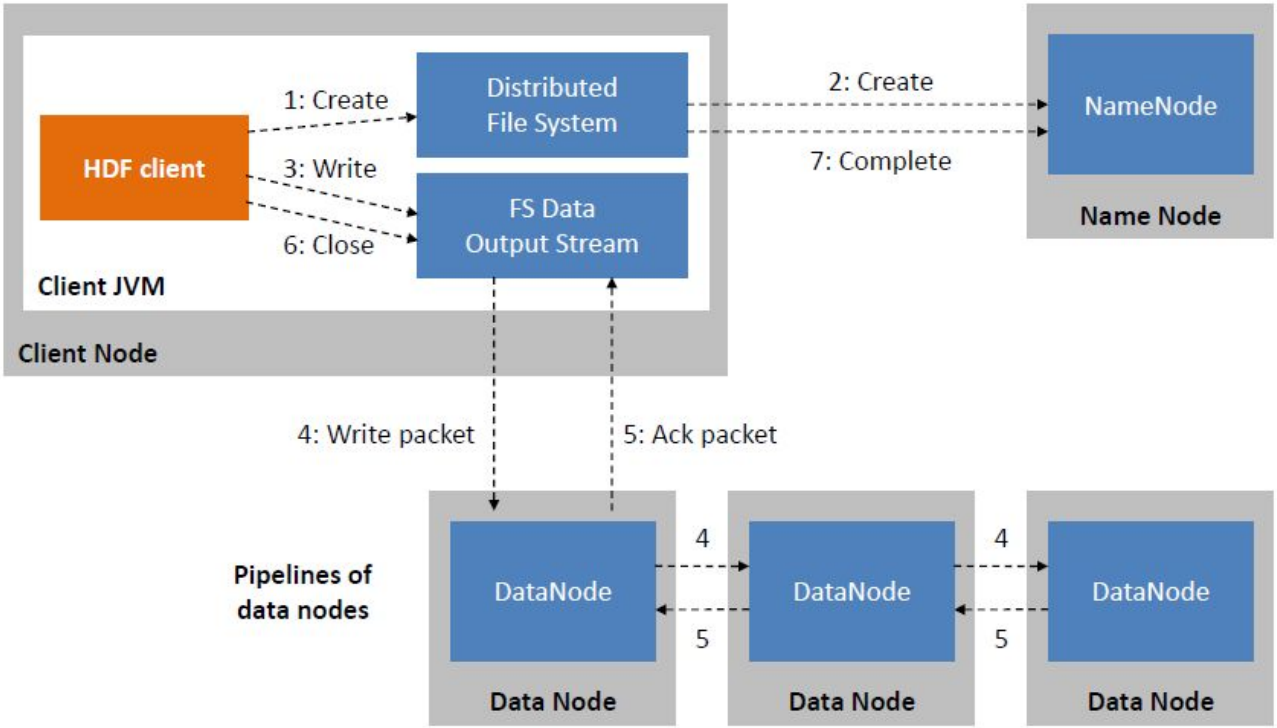
Репликация


Name Node: Stores Meta Data
Meta Data:
/data/pristine/catalina.log.> 1, 2, 4
/data/pristine/myfile. >3,5
Data Node 1
1
2
4
5
Data Node 2
5
2
3
Data Node 3
4
1
3

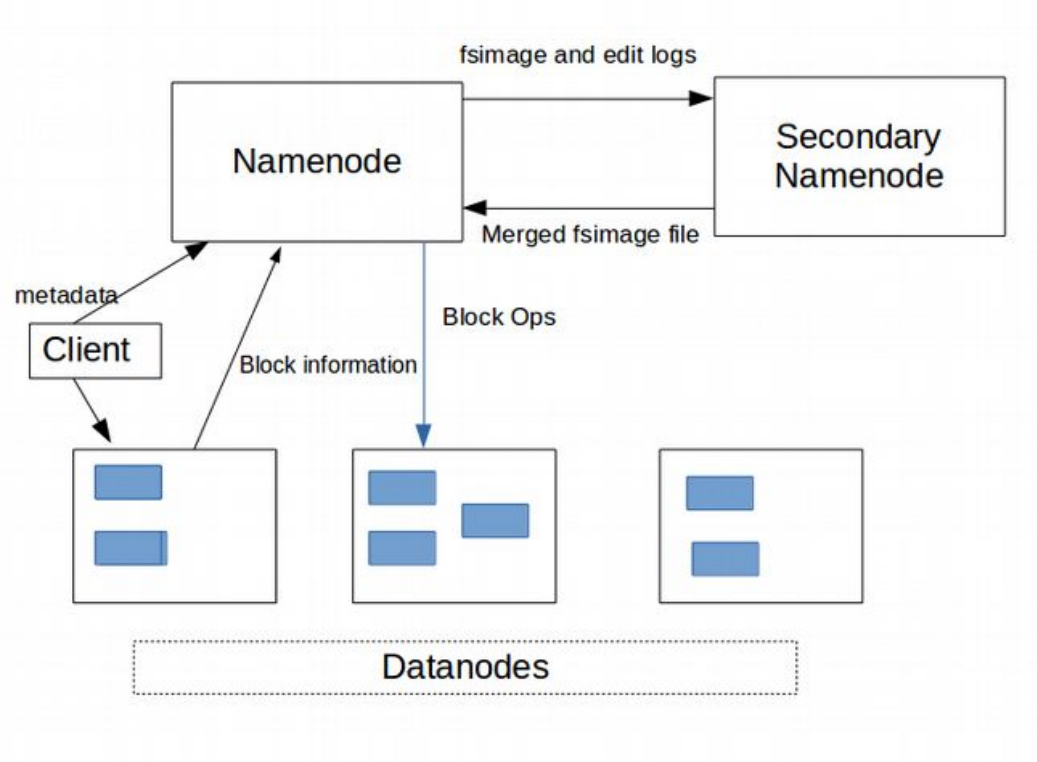
Чтение


HDF client
1: Open
3: Read
6: Close
Distributed File System
FS Data Input Stream
Client JVM
Client Node
2: Get block locations
NameNode
Name Node
4: Read
5: Read
DataNode
Data Node
DataNode
Data Node
DataNode
Data Node

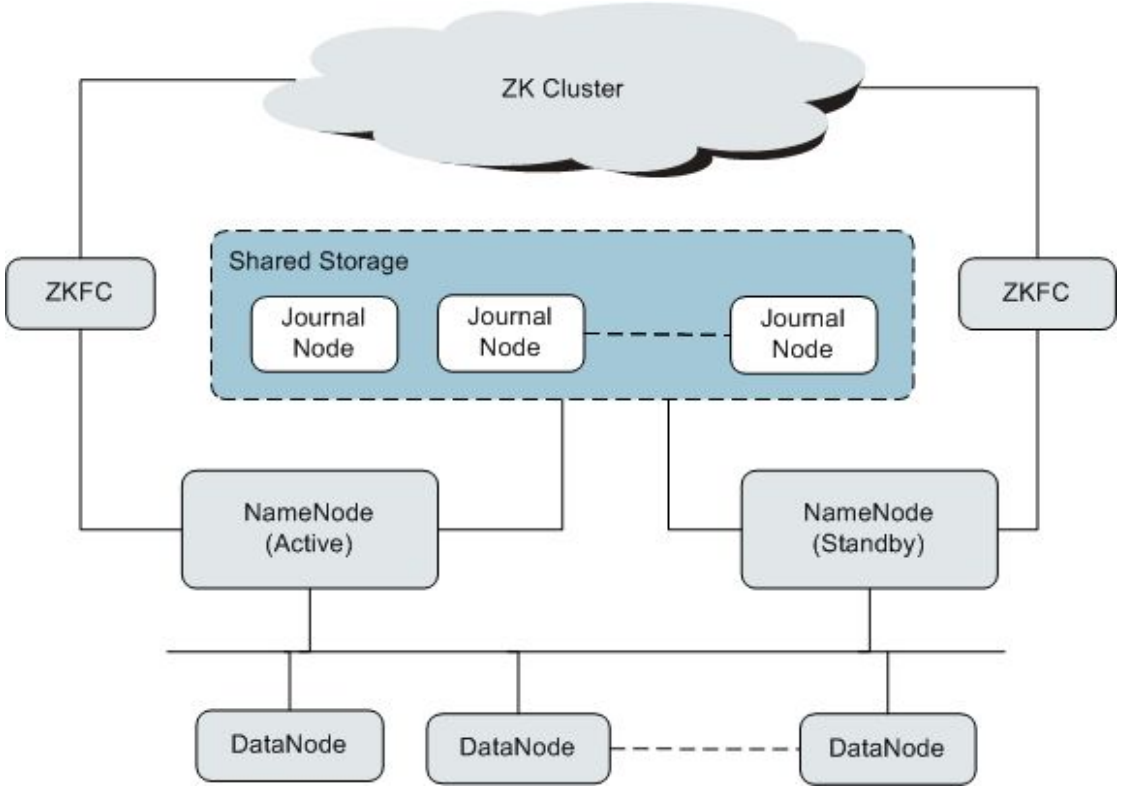
Запись


HDF client
1: Create
3: Write
6: Close
Distributed File System
FS Data Output Stream
Client JVM
Client Node
2: Create
7: Complete
NameNode
Name Node
4: Write packet
5: Ack packet
Pipelines of data nodes
DataNode
DataNode
DataNode
4
5
4
5
Data Node
Data Node
Data Node

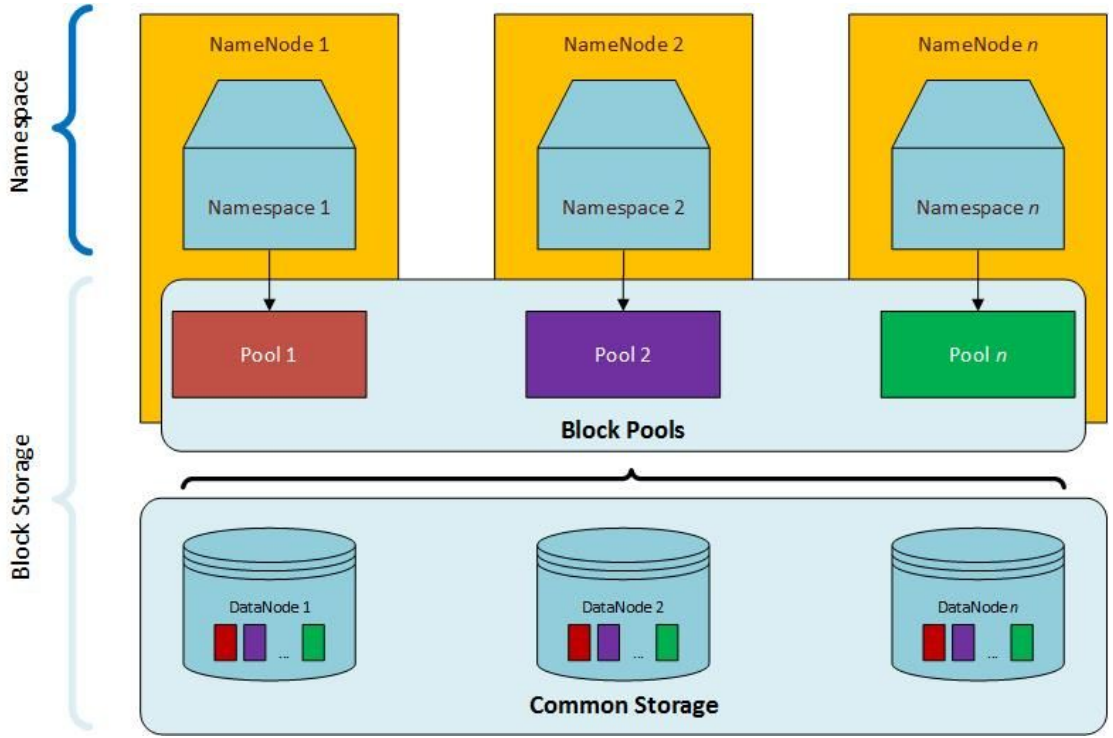
Secondary Namenode



High-Availability



Federation


Namespace
Block Storage
NameNode 1
NameNode 2
NameNode n
Namespace 1
Namespace 2
Namespace n
Pool 1
Pool 2
Pool n
Block Pools
DataNode 1
DataNode 2
DataNode n
Common Storage

HDFS

Несколько фактов:

- Не любит мелкие файлы
- Не поддерживает редактирование файлов
- Работает на картошке

HDFS API

У HDFS есть несколько API

- libhdfs - C API
- FileSystem - Java API
- WebHDFS/HttpFS - REST API
- HDFS CLI

HDFS API

Для Java/Scala можно использовать FileSystem из стандартного Hadoop API

https://hadoop.apache.org/docs/current/api/org/apache/hadoop/fs/FileSystem.html

HDFS API

На основе этих API есть несколько Python-библиотек

- hdfs
- hdfs3
- snakebite

HDFS API

```
from hdfs3 import HDFileSystem


hdfs = HDFileSystem(host, port, user)

with hdfs.open('/path/to/file', 'rb') as f:

    ...
```

HDFS CLI

```
hdfs dfs -cat hdfs://nn1.example.com/file1

hdfs dfs -ls <args>

hdfs dfs -cp /user/hadoop/file1 /user/hadoop/file2
```

YARN

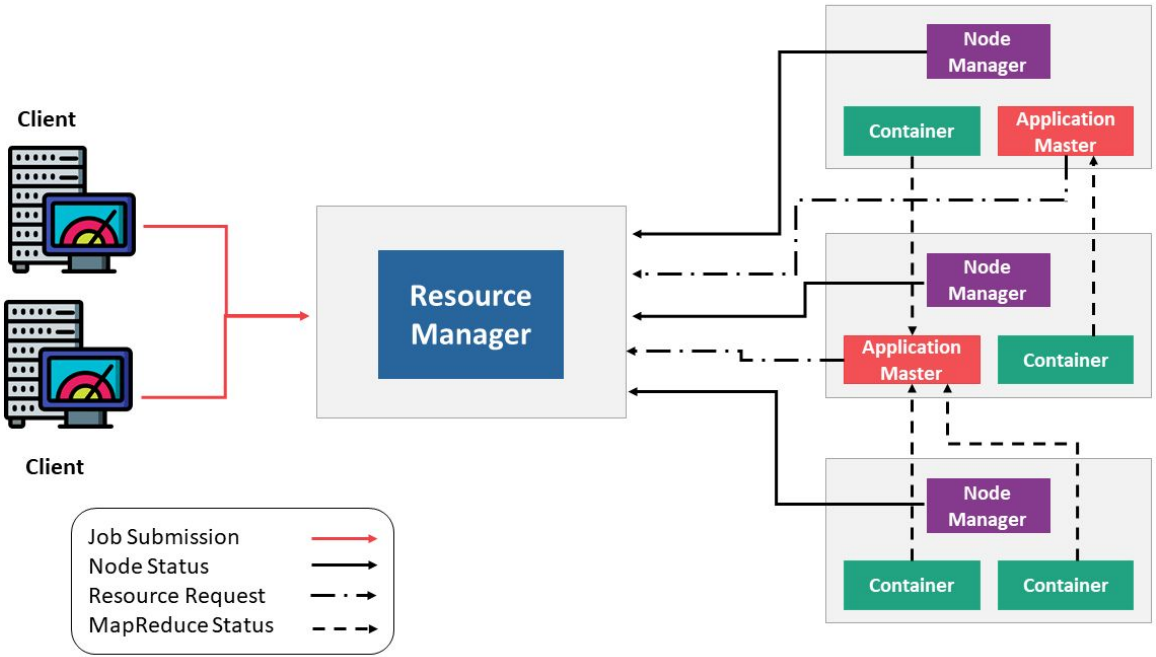
Системы контейнеризации

- Kubernetes
- Mesos
- YARN

YARN

YARN (англ. Yet Another Resource Negotiator) — модуль, отвечающий за управление ресурсами кластеров и планирование заданий.

Контейнеры


Client
Client
Resource Manager
Node Manager
Container
Application Master
Node Manager
Application Master
Container
Node Manager
Container
Container
Job Submission
Node Status
Resource Request
MapReduce Status

Очереди

Разделение ресурсов происходит по очередям.

Например,

- Отдельная очередь для долгих тяжелых задач
- Отдельная очередь для мелких ad-hoc запросов
- Отдельная очередь для обучения моделей
- Отдельная очередь на каждый отдел и т. д.

Задачка

Задачка

1. Представьте, что вы в команде из 6 человек.
2. Команда получила пакет с набором из стикеров 4 разных цветов.
3. Нужно подсчитать количество каждого цвета.

На обдумывание алгоритма 2-3 минуты

MapReduce

MapReduce

MapReduce - подход к организации распределенных вычислений, предложенный Google в работе “MapReduce: Simplified Data Processing on Large Clusters”

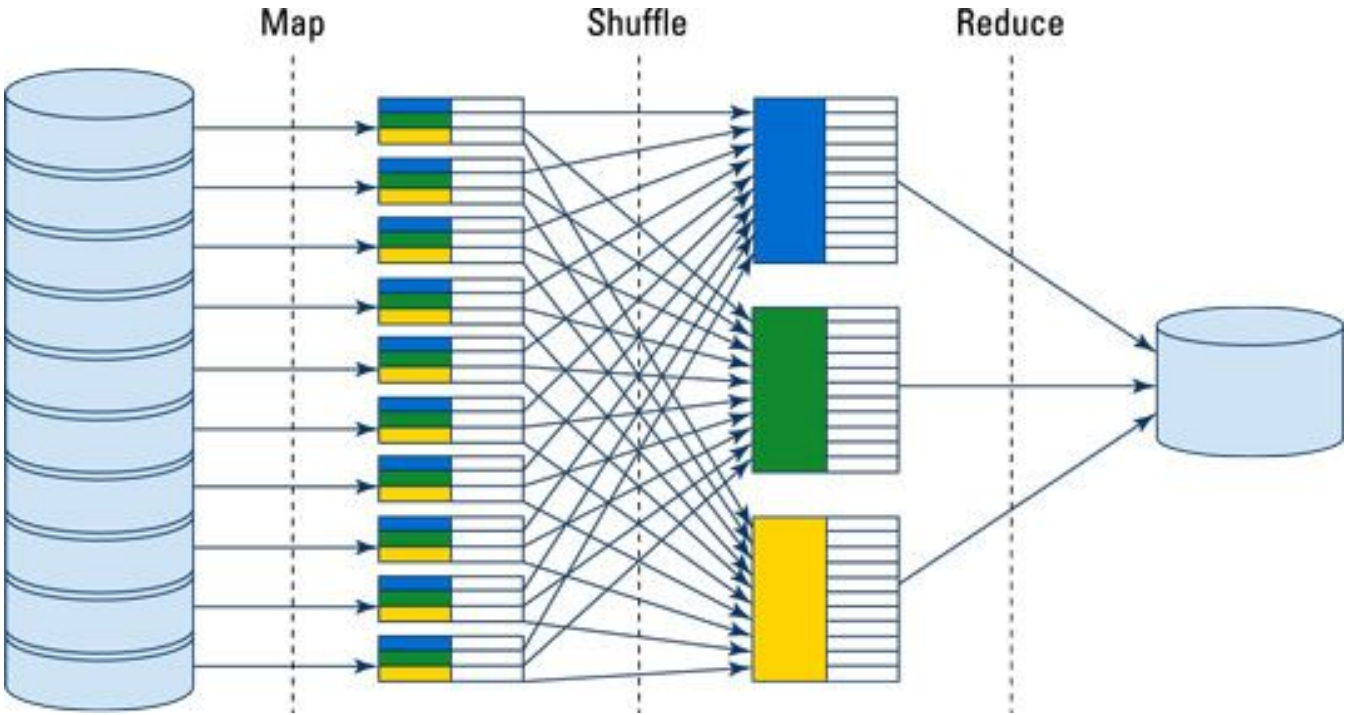
MapReduce

Главная идея:

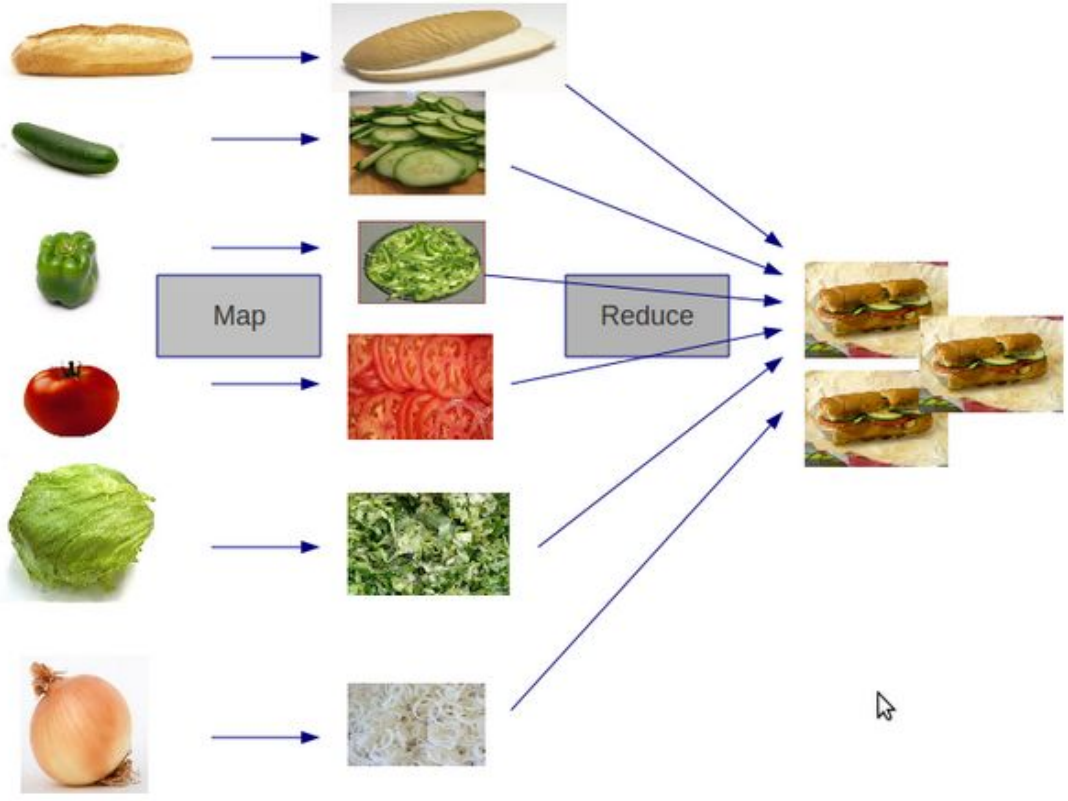
Вычисления можно представить как последовательность операций

- Map - трансформаций
- Shuffle - перераспределения данных
- Reduce - агрегаций

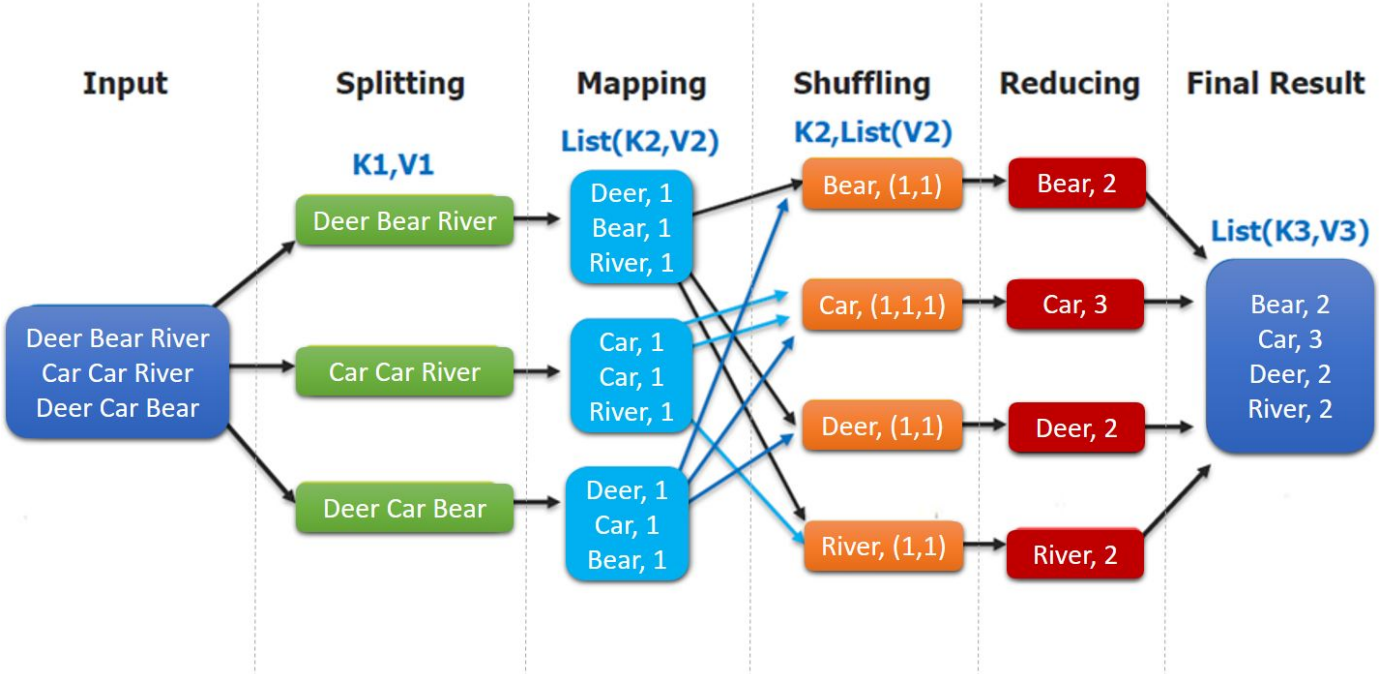
MapReduce



MapReduce



MapReduce



MapReduce Word Count Process

Рассмотрим пример WordCount

MapReduce

```
hadoop jar hadoop-streaming.jar \
-input input_dir\
-output output_dir\
-mapper "python mapper.py"\
-reducer "python reducer.py"\
-combiner "python reducer.py"\
-file "mapper.py"\
-file "reducer.py"
```

MapReduce

```
def map(doc):
    for word in doc:
        yield word, 1

def reduce(word, values):
    yield word, sum(values)
```

Hive

Hive

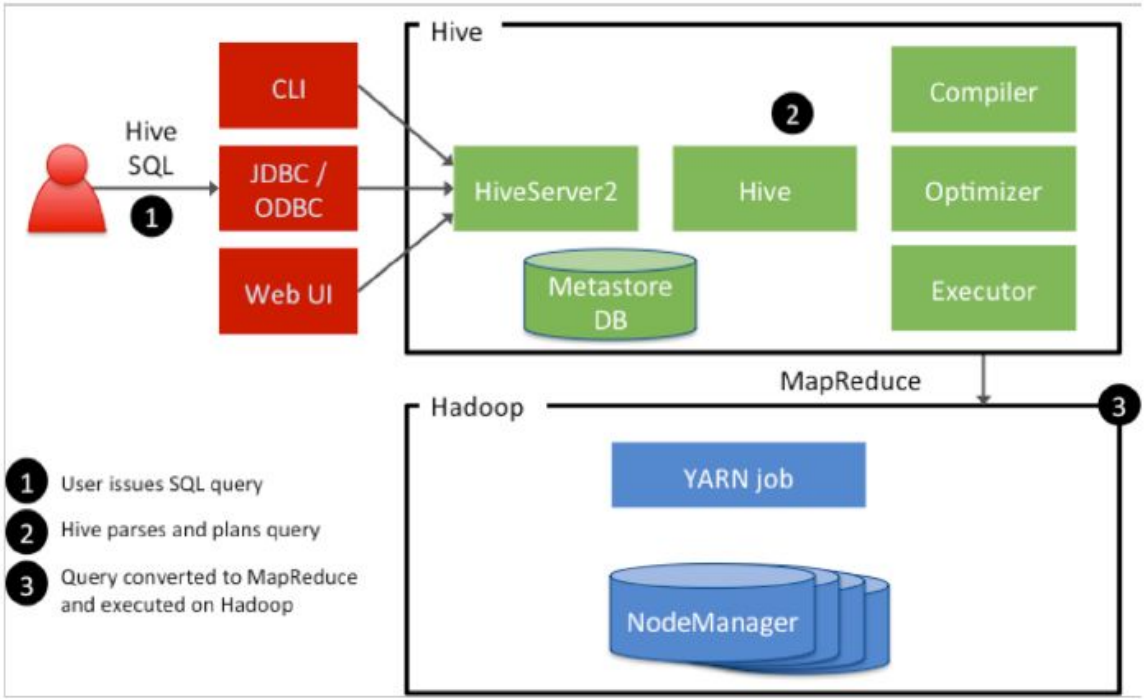
- Позволяет выполнять запросы к слабоструктурированным данным
- Для запросов используется HiveQL
- Имеет свой metastore для хранения “данных о данных” (структур таблиц)

Пример запроса

```
CREATE TABLE u_data_new (
  userid INT,
  movieid INT,
  rating INT,
  weekday INT)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t';
```

```
SELECT weekday, COUNT(*)
FROM u_data_new
GROUP BY weekday;
```

Архитектура



Hive Architecture

Партиции

Партиции - разбиение данных на уровне файловой системы

SELECT ... WHERE year = '2009';

date	amount
11/11/09	6
06/05/09	12
...	...
12/02/09	8

Min: 01/01/09
Max: 12/31/09

date	amount
03/13/08	96
04/21/08	17
...	...
12/02/08	7

Min: 01/01/08
Max: 12/31/08

date	amount
07/12/07	43
03/02/07	45
...	...
12/02/07	68

Min: 01/01/07
Max: 12/31/07

Партиции

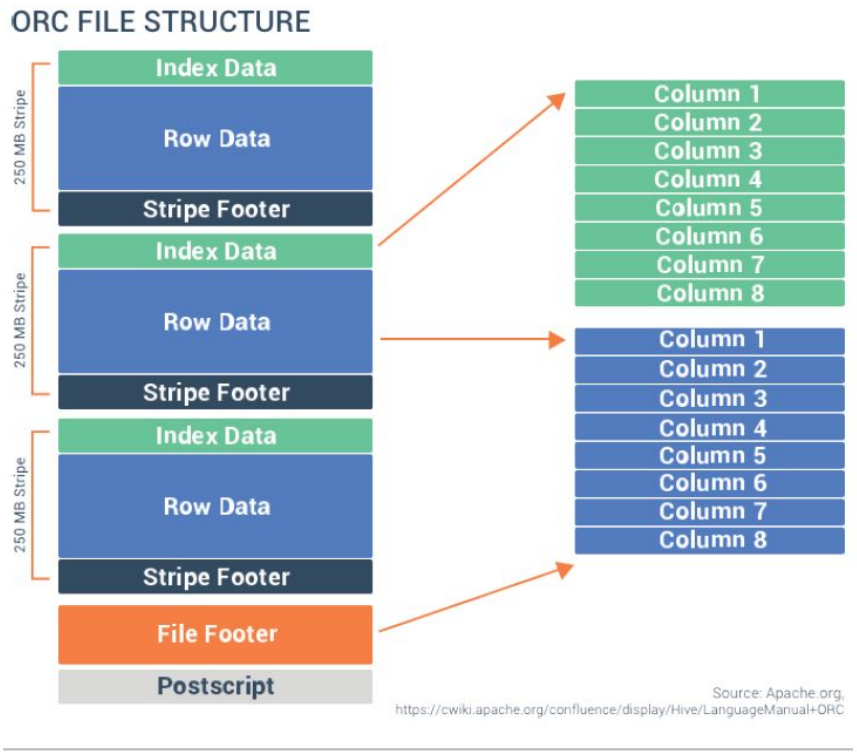
```
CREATE TABLE page_view (
 viewTime INT, userid BIGINT, page_url STRING
) PARTITIONED BY (dt STRING, country STRING)
STORED AS SEQUENCEFILE;
```

Storage Formats

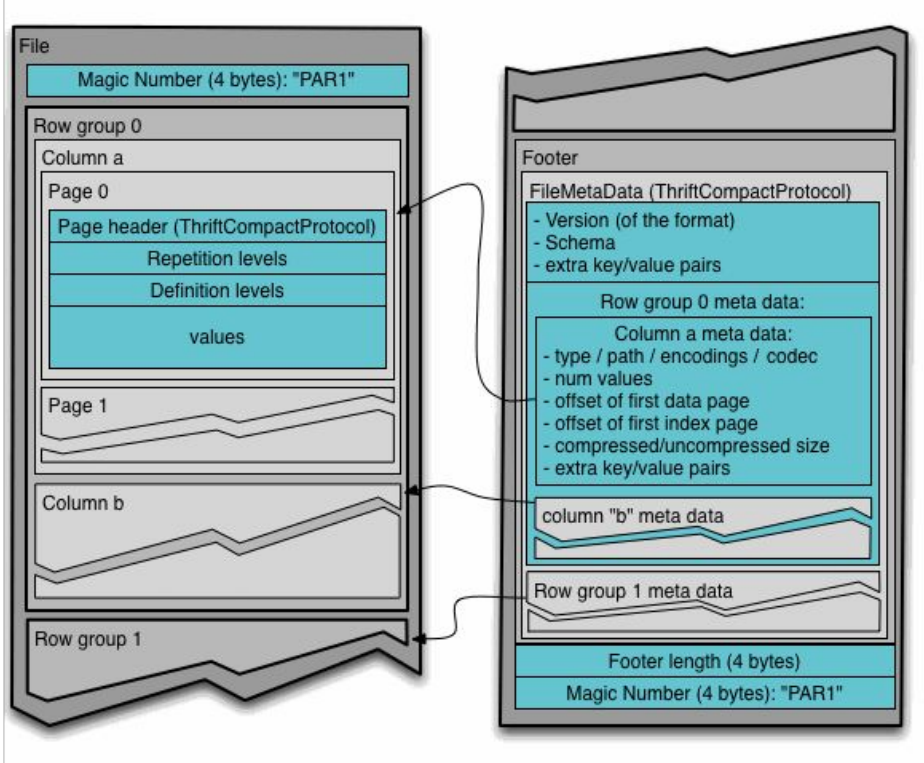
```
CREATE TABLE my_table(a string, b BIGINT, ...)
```

- STORED AS TEXTFILE;
- STORED AS PARQUET;
- STORED AS ORC;
- ...

Columnar formats: Orc



Columnar formats: Parquet



Avro

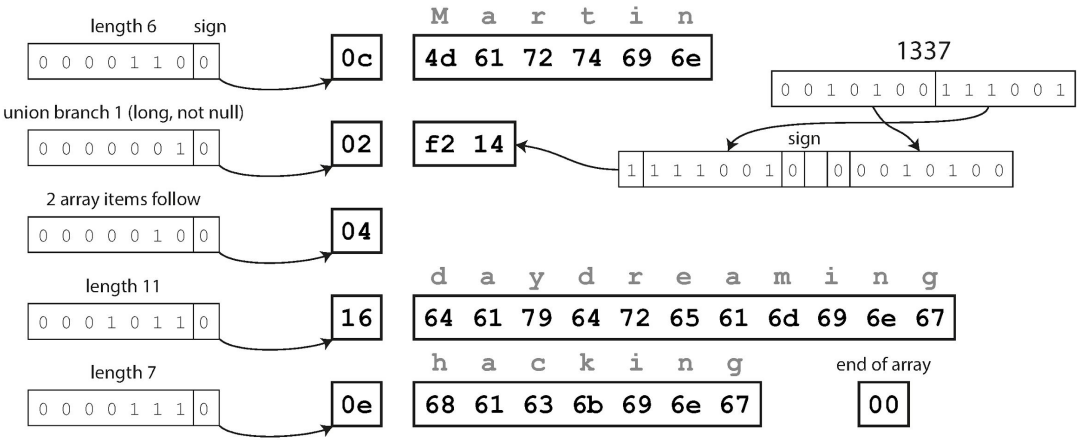
Avro

Byte sequence (32 bytes):

0c	4d	61	72	74	69	6e	02	f2	14	04	16	64	61	79	64	72	65	61	6d

69	6e	67	0e	68	61	63	6b	69	6e	67	00

Breakdown:



SerDe

```
CREATE TABLE my_table(a string, b BIGINT, ...)

ROW FORMAT SERDE 'org.apache.hive.hcatalog.data.JsonSerDe'

STORED AS TEXTFILE;
```

Внешние таблицы

```
CREATE EXTERNAL TABLE page_view (
    viewTime INT, userid BIGINT, page_url STRING
) STORED AS PARQUET LOCATION '<hdfs_location>';
```

MSCK REPAIR TABLE

Часто встречающаяся ситуация

- есть внешняя таблица Hive
- туда были записаны данные
- данных не видно в запросах SQL

Поможет команда `MSCK REPAIR TABLE table_name`

API

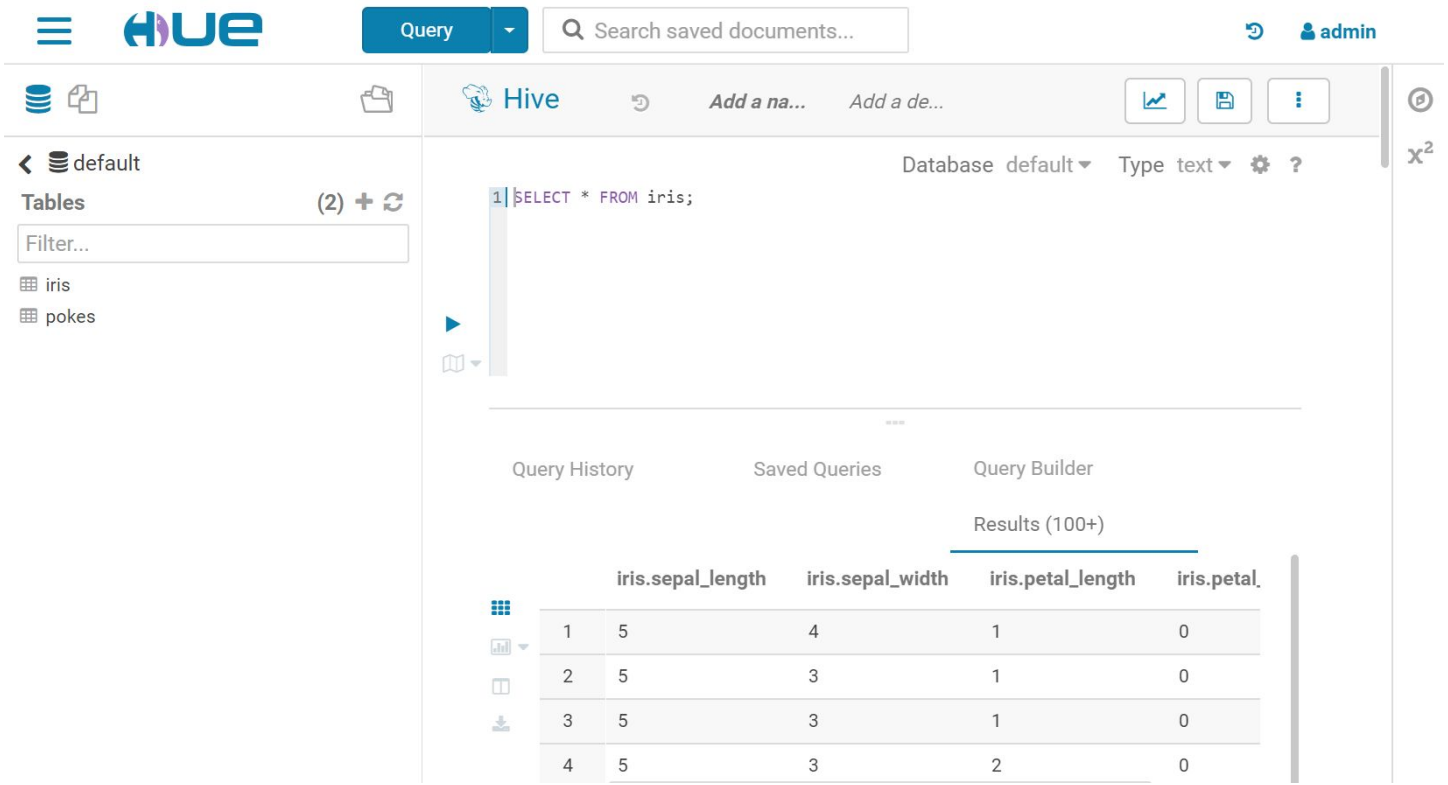
С Hive можно взаимодействовать через

- Beeline CLI
- HUE
- Библиотеки (JDBC, pyhive)

CLI

```
% bin/beeline
beeline> !connect jdbc:hive2://localhost:10000 user1 password1
0: jdbc:hive2://localhost:10000> show tables;
+--------------------+
|     tab_name       |
+--------------------+
| primitives         |
| srcbucket2         |
| srcpart            |
+--------------------+
```

HUE


HUE
Query
Search saved documents...
admin
Hive
Add a na...
Add a de...
default
Tables (2)
Filter...
iris
pokes
Database default
Type text
1 SELECT * FROM iris;
Query History
Saved Queries
Query Builder
Results (100+)
iris.sepal_length
iris.sepal_width
iris.petal_length
iris.petal_
1 5 4 1 0
2 5 3 1 0
3 5 3 1 0
4 5 3 2 0

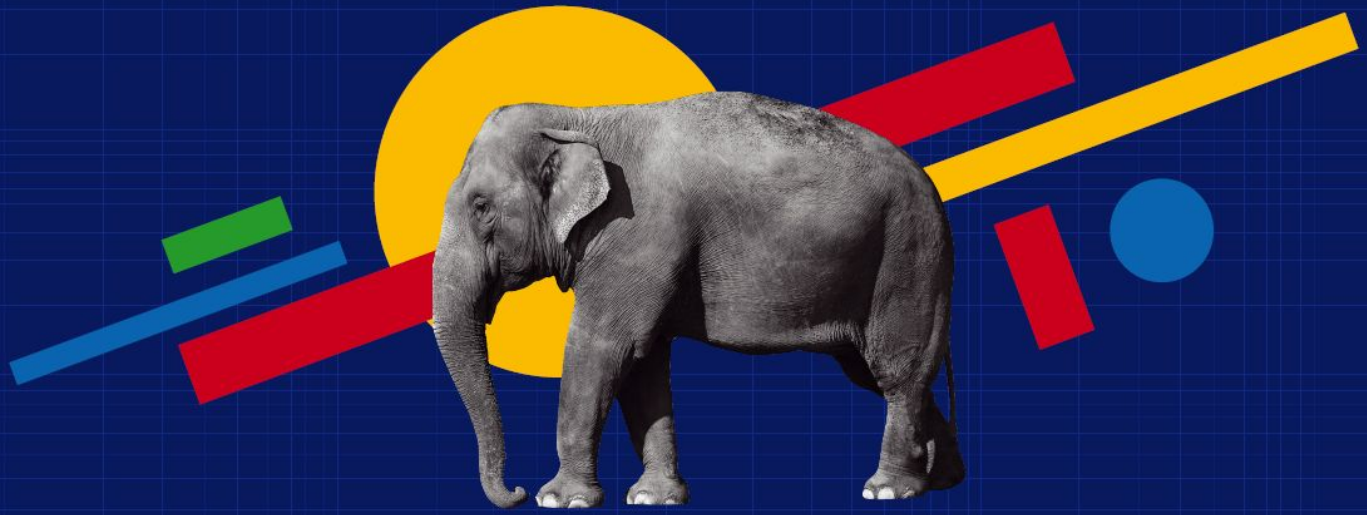
Подведем итоги

Зачем это занятие

После этого занятия вы будете знать

- Что такое Hadoop
- Из каких основных сервисов он состоит
- Как использовать эти сервисы

Егор Матешук
egor@mateshuk.com



Big Data is Love

NEWPROLAB.COM