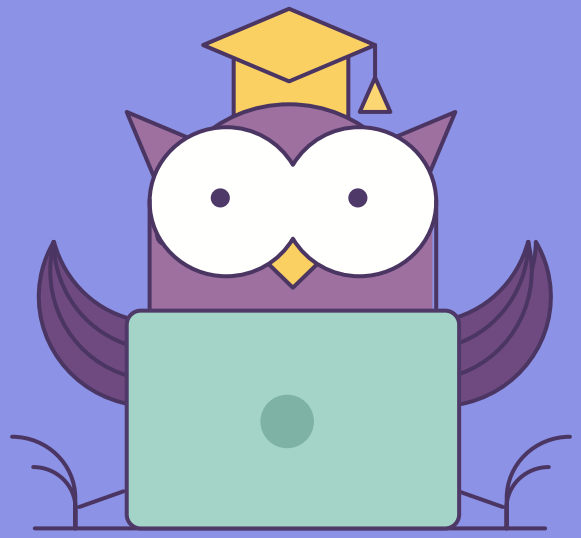
ОНЛАЙН-ОБРАЗОВАНИЕ

Меня хорошо слышно && видно?



Напишите в чат, если есть проблемы!

Ставьте + если все хорошо

Илья Маркин

Senior Software Engineer, insolar.io

amferiuss@gmail.com
@amferiuss

Правила вебинара

Задаем вопрос в чат или голосом

slack

Off-topic обсуждаем в Slack #канал группы или #general

Вопросы вижу в чате, могу ответить не сразу

OTUS

HDFS

Hadoop Distributed File System

Илья Маркин amferiuss@gmail.com @amferiuss

01

Файловая система

Фа́йловая систе́ма — порядок, определяющий способ организации, хранения и именования данных на носителях информации в компьютерах, а также в другом электронном оборудовании: цифровых фотоаппаратах, мобильных телефонах и т. п

Основные функции любой файловой системы нацелены на решение следующих задач:

- именование файлов;
- программный интерфейс работы с файлами для приложений;
- отображения логической модели файловой системы на физическую организацию хранилища данных;
- организация устойчивости файловой системы к сбоям питания, ошибкам аппаратных и программных средств;
- содержание параметров файла, необходимых для правильного его взаимодействия с другими объектами системы (ядро, приложения и пр.).

Примеры: FAT16, FAT32, NTFS, exFAT, ext4 etc.

- В многопользовательских системах появляется ещё одна задача: защита файлов одного пользователя от несанкционированного доступа другого пользователя, а также обеспечение совместной работы с файлами, к примеру, при открытии файла одним из пользователей, для других этот же файл временно будет доступен в режиме «только чтение».

- Файловая система, по большому счёту, состоит из таблицы файловых дескрипторов и области данных.

02

Распределенная файловая система

Распределённая файловая система - это тоже файловая система, которая поддерживается одним или более компьютерами. Также известна как сетевая файловая система.

В принципе то же самое что и обычная **файловая система**, только данные раскиданы по какому-то количеству компьютеров. И объединены одним сервисом, который отвечает за разыменование пути к необходимому файлу.

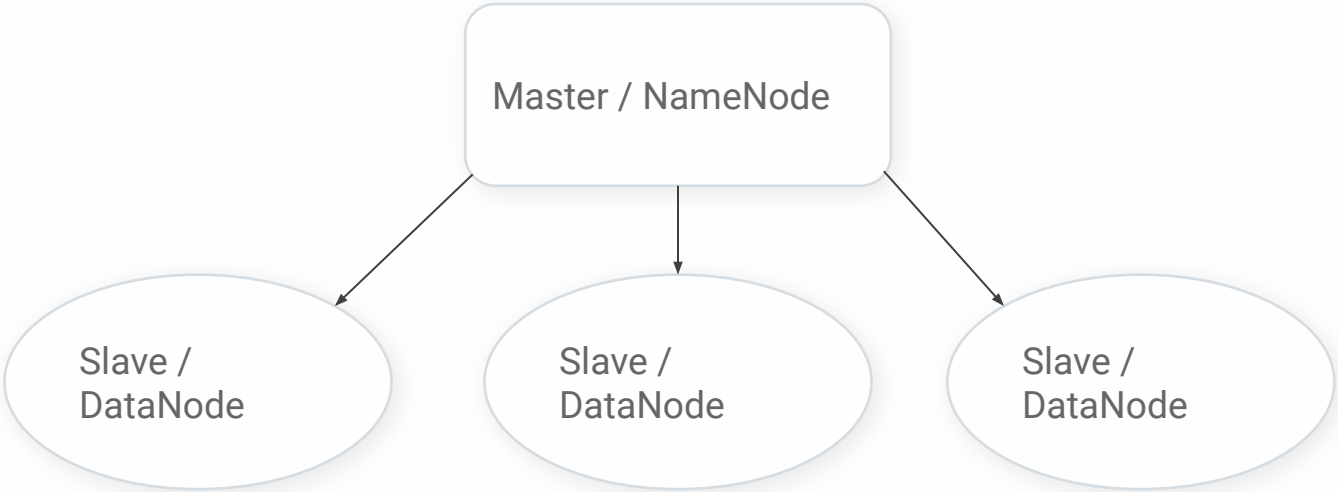
Примеры: HDFS, NFS, DFS, GFS, Ceph etc.

03

HDFS

Архитектура

Архитектура (просто)



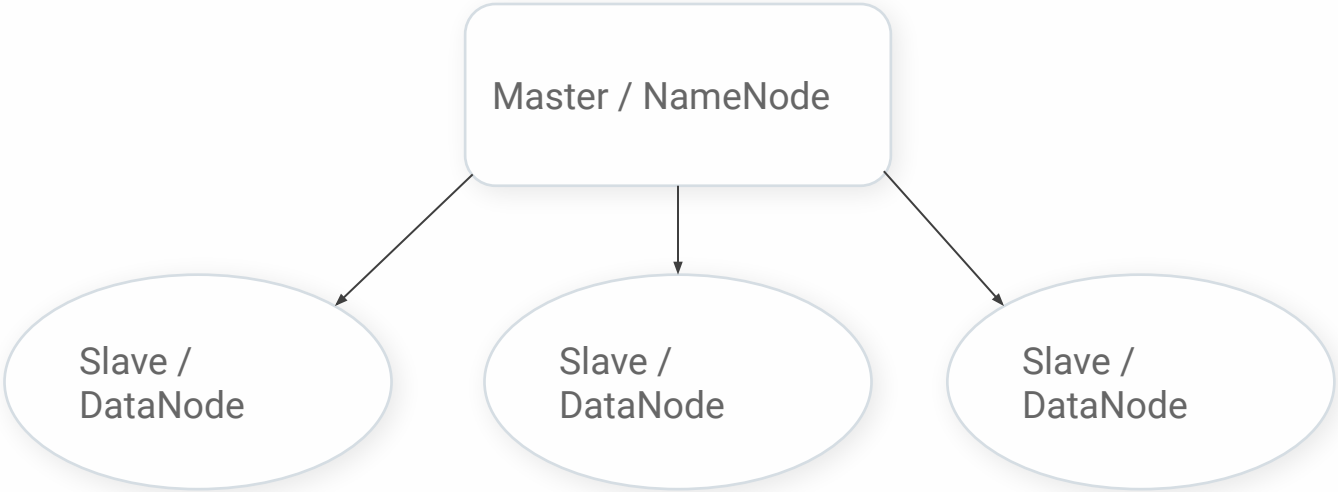
NameNode (просто)

- Выполняет операции по обслуживанию namespace - открытие, закрытие, переименование файлов и директорий
- Назначает права доступа на файлы и директории
- Определяет маппинг блоков данных на DataNodes

DataNode (просто)

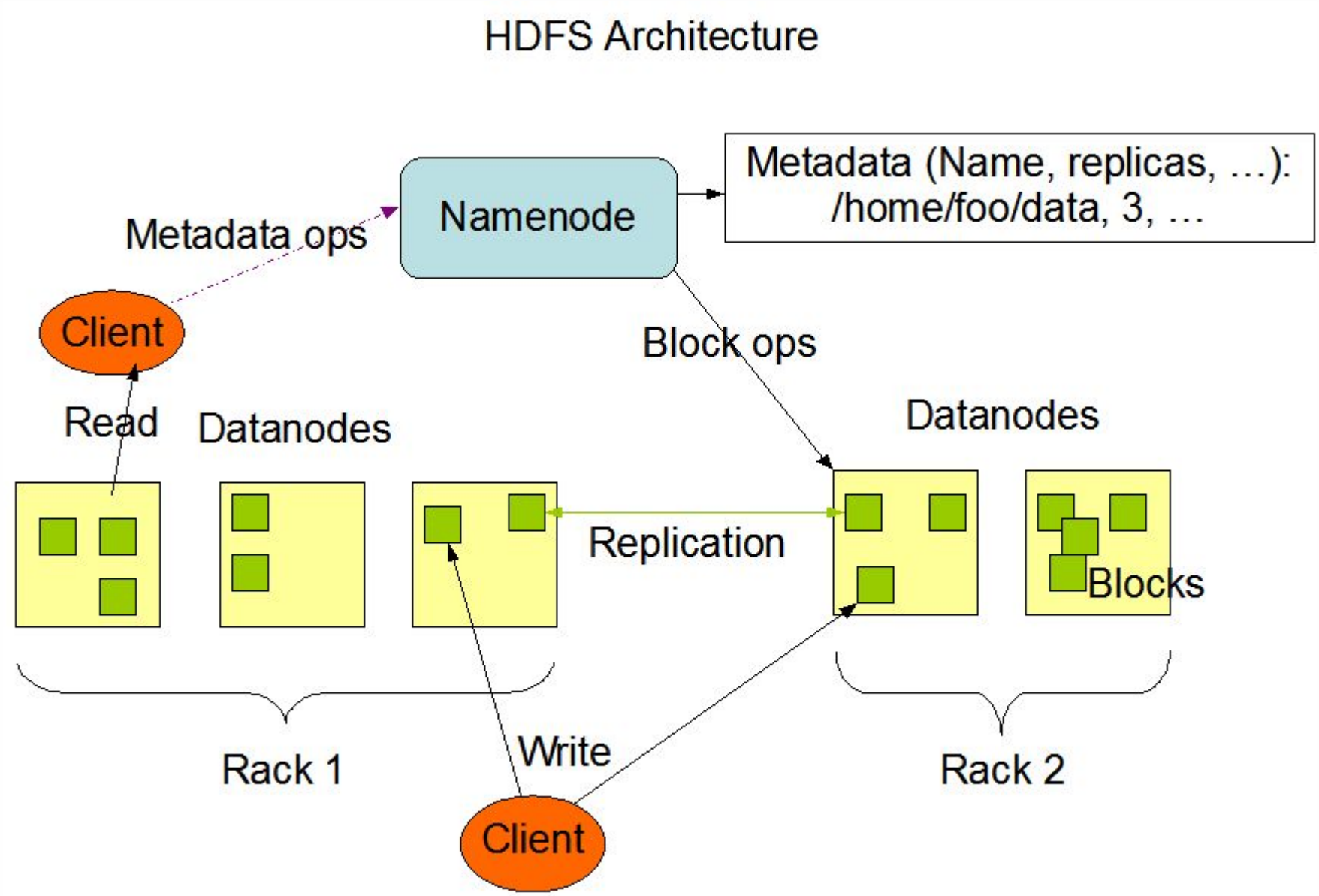
- Отвечает за чтение и запись запросов от клиентов
- Создает, удаляет и реплицирует блоки данных под командой NameNode

Архитектура (просто)



… А в чем подвох?

Архитектура (сложнее)



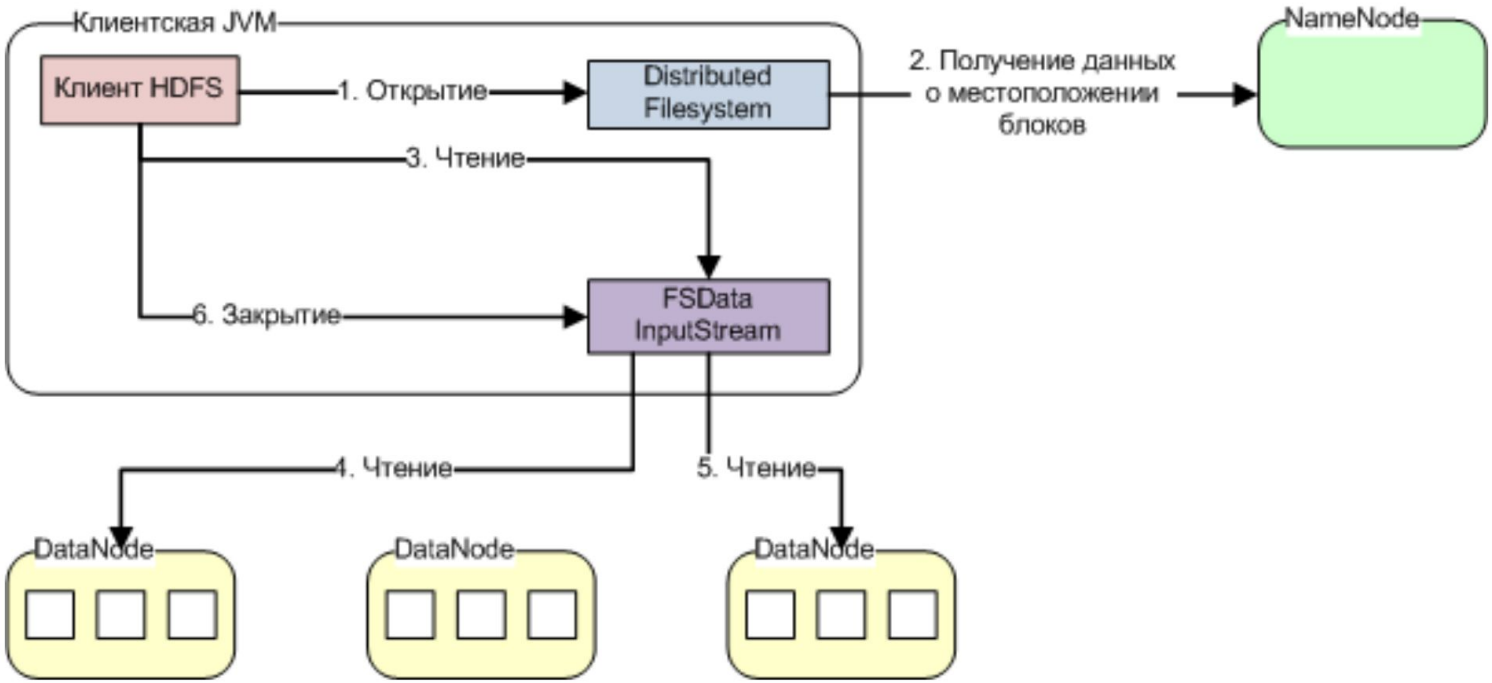
source: https://hadoop.apache.org/

04

HDFS

Чтение

Чтение HDFS

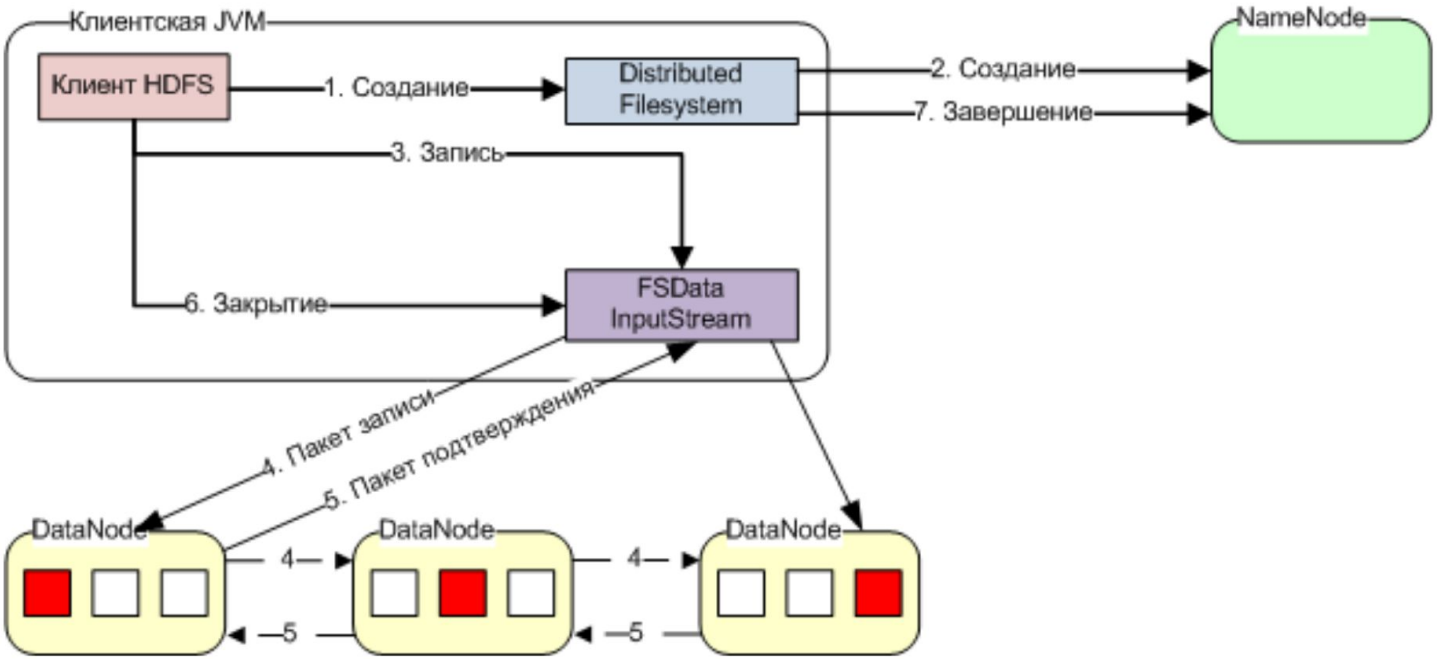

Клиентская JVM
Клиент HDFS
1. Открытие
Distributed Filesystem
2. Получение данных о местоположении блоков
NameNode
3. Чтение
6. Закрытие
FSData InputStream
4. Чтение
5. Чтение
DataNode
DataNode
DataNode

05

HDFS

Запись

Запись HDFS

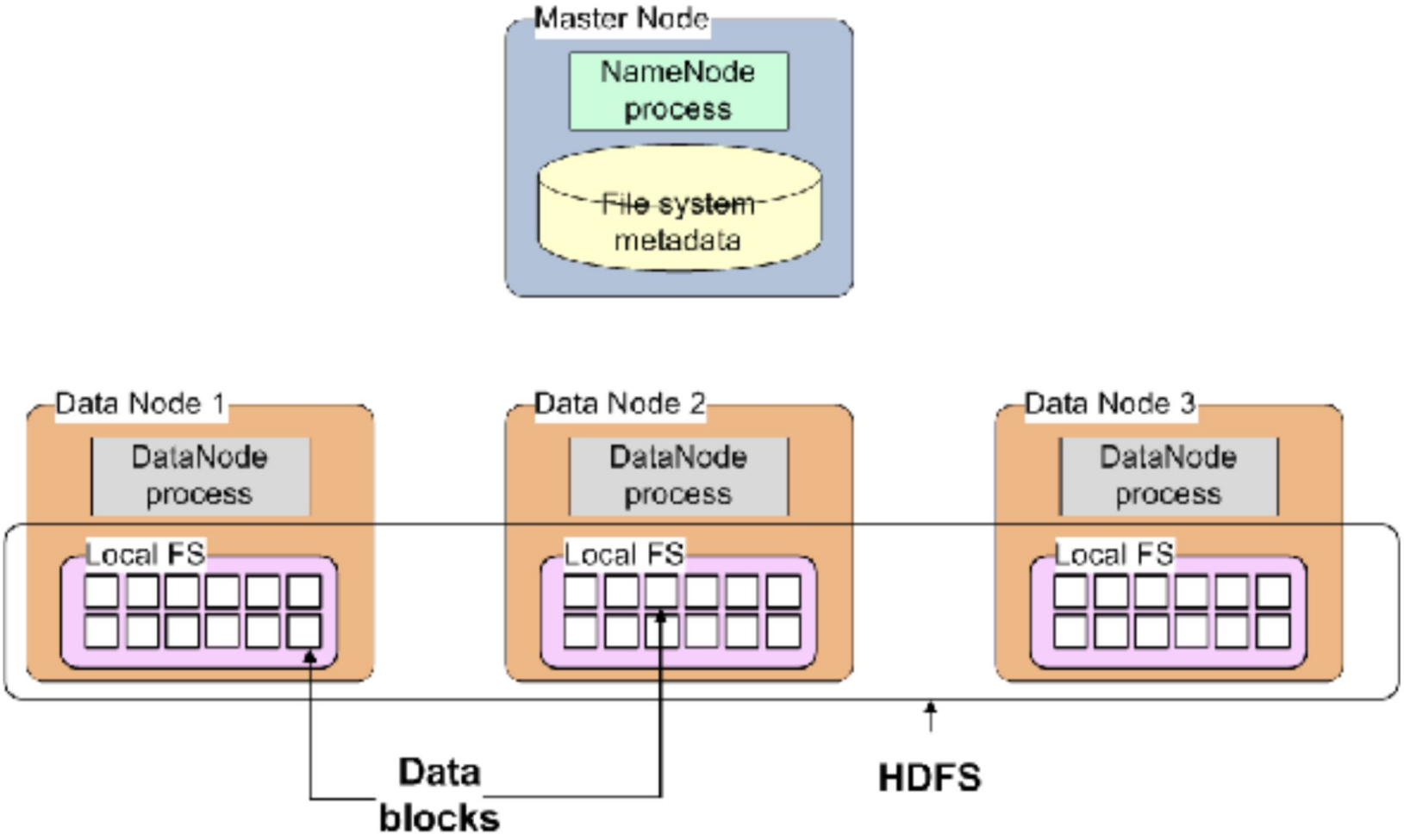

Клиентская JVM
Клиент HDFS
1. Создание
Distributed Filesystem
2. Создание
7. Завершение
NameNode
3. Запись
6. Закрытие
FSData InputStream
4. Пакет записи
5. Пакет подтверждения
DataNode
DataNode
DataNode
4
4
5
5

06

HDFS

Хранение

Хранение


Master Node
NameNode process
File system metadata
Data Node 1
DataNode process
Local FS
Data Node 2
DataNode process
Local FS
Data Node 3
DataNode process
Local FS
Data blocks
HDFS

NameNode: Структура хранения

OTUS

${dfs.namenode.name.dir}/

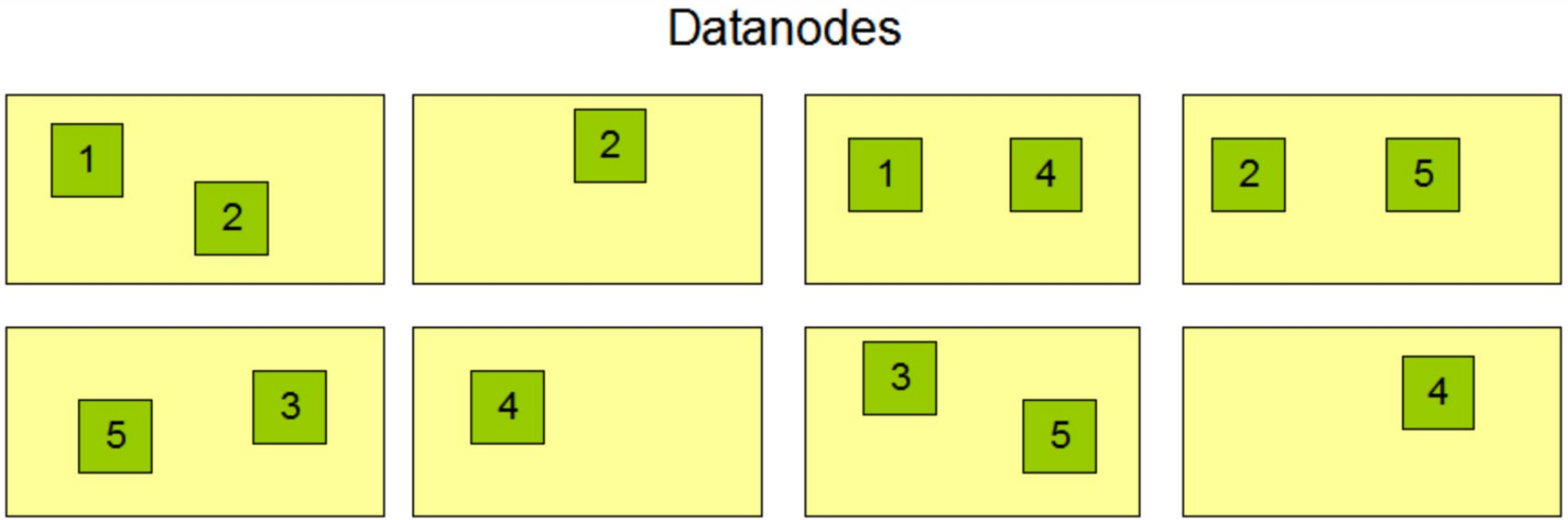
- VERSION (информация о версии HDFS)
- edits (журнал изменений)
- fsimage (контрольная точка метаданных)
- fstime (время создания контрольной точки)

DataNode: Структура хранения

${dfs.datanode.data.dir}/

- ....

(хранятся блоки)



DataNode: Структура хранения

- DataNode ничего не знает о HDFS файлах. Она хранит каждый блок данных в разных файлах в локальной файловой системе. НЕ хранит все в одной папке. На основе определенных эвристик определяет необходимое количество файлов для оптимизации производительности поиска по файловой системе.

Сведения о блоках файла

OTUS

hdfs fsck /user/hive/warehouse/foo_bar/000000_0 -files -blocks -locations

```
[root@hdp-15 ~]# hdfs fsck /user/hive/warehouse/big_cdr_parquet/000000_0 -files -blocks -locations
Connecting to namenode via http://hdp-7:50070
FSCK started by root (auth:SIMPLE) from /192.168.91.141 for path /user/hive/warehouse/big_cdr_parquet/000000_0
at Mon May 18 14:00:22 MSK 2015
/user/hive/warehouse/big_cdr_parquet/000000_0 1133129924 bytes, 9 block(s):  OK
0. BP-1972162810-192.168.91.133-1428693610895:blk_1073747244_6432 len=134217728 repl=3
[192.168.91.141:50010, 192.168.91.139:50010, 192.168.91.133:50010]
1. BP-1972162810-192.168.91.133-1428693610895:blk_1073747245_6433 len=134217728 repl=3
[192.168.91.136:50010, 192.168.91.139:50010, 192.168.91.141:50010]
2. BP-1972162810-192.168.91.133-1428693610895:blk_1073747246_6434 len=134217728 repl=3
[192.168.91.142:50010, 192.168.91.136:50010, 192.168.91.141:50010]
3. BP-1972162810-192.168.91.133-1428693610895:blk_1073747247_6435 len=134217728 repl=3
[192.168.91.134:50010, 192.168.91.142:50010, 192.168.91.137:50010]
4. BP-1972162810-192.168.91.133-1428693610895:blk_1073747248_6436 len=134217728 repl=3
[192.168.91.135:50010, 192.168.91.133:50010, 192.168.91.137:50010]
5. BP-1972162810-192.168.91.133-1428693610895:blk_1073747249_6437 len=134217728 repl=3
[192.168.91.140:50010, 192.168.91.137:50010, 192.168.91.142:50010]
6. BP-1972162810-192.168.91.133-1428693610895:blk_1073747250_6438 len=134217728 repl=3
[192.168.91.142:50010, 192.168.91.139:50010, 192.168.91.141:50010]
7. BP-1972162810-192.168.91.133-1428693610895:blk_1073747251_6439 len=134217728 repl=3
[192.168.91.139:50010, 192.168.91.140:50010, 192.168.91.135:50010]
8. BP-1972162810-192.168.91.133-1428693610895:blk_1073747252_6440 len=59388100 repl=3 [192.168.91.141:50010,
192.168.91.137:50010, 192.168.91.135:50010]
```

Сведения о блоках файла (продолжение)

hdfs fsck /user/hive/warehouse/foo_bar/000000_0 -files -blocks -locations

```
Status: HEALTHY
 Total size:    1133129924 B
 Total dirs:    0
 Total files:   1
 Total symlinks:                0
 Total blocks (validated):      9 (avg. block size 125903324 B)
 Minimally replicated blocks:   9 (100.0 %)
 Over-replicated blocks:        0 (0.0 %)
 Under-replicated blocks:       0 (0.0 %)
 Mis-replicated blocks:         0 (0.0 %)
 Default replication factor:    3
 Average block replication:     3.0
 Corrupt blocks:                0
 Missing replicas:              0 (0.0 %)
 Number of data-nodes:          10
 Number of racks:               1
FSCK ended at Mon May 18 14:00:22 MSK 2015 in 1 milliseconds
```

07

HDFS

Репликация

Репликация

- NameNode периодически опрашивает (собирает HeartBeat) и blockreport с каждой DataNode. Blockreport содержит список всех блоков со всех DataNode.

- DataNode рассказывает о своем состоянии и блоках.

Репликация

```
Namenode (Filename, numReplicas, block-ids, ...)
/users/sameerp/data/part-0, r:2, {1,3}, ...
/users/sameerp/data/part-1, r:3, {2,4,5}, ...
```

dfs.replication	3	Default block replication. The actual number of replications can be specified when the file is created. The default is used if replication is not specified in create time.
dfs.replication.max	512	Maximal block replication.
dfs.namenode.replication.min	1	Minimal block replication.
dfs.blocksize	134217728	The default block size for new files, in bytes. You can use the following suffix (case insensitive): k(kilo), m(mega), g(giga), t(tera), p(peta), e(exa) to specify the size (such as 128k, 512m, 1g, etc.), Or provide complete size in bytes (such as 134217728 for 128 MB).

08

HDFS

Checkpoints

Checkpoints

- Цель - убедиться, что HDFS находится в консистентном состоянии

Checkpoints

- **EditLog.** NameNode использует transaction log, называемый EditLog. В него записываются все изменения метаданных системы *(например: создание нового файла, изменение фактора репликации файла etc)*

- **FsImage.** Хранит маппинг блоков файлов и свойства файлов.

Checkpoints

- **EditLog.** NameNode использует transaction log, называемый EditLog. В него записываются все изменения метаданных системы *(например: создание нового файла, изменение фактора репликации файла etc)*

- **FsImage.** Хранит маппинг блоков файлов и свойства файлов.

Checkpoints

- NameNode поднимает с диска ранее сохраненный FsImage и EditLog, применяет все записи из EditLog в памяти на FsImage с диска.
- Далее удаляет выполняет очистку (truncate) EditLog.

Checkpoints

```
# время между checkpoints в сек.
dfs.namenode.checkpoint.period

# количество транзакций между checkpoints
dfs.namenode.checkpoint.txns
```

(если настроены оба - срабатывает первый достигнутый)

09

HDFS

Работа с HDFS в CLI

Работа с HDFS (CLI)

```
# просмотреть корневую директорию: локально и на HDFS
ls /
hadoop fs -ls /

# оценить размер директории
du -sh mydata
hadoop fs -du -s -h mydata

# вывести на экран содержимое всех файлов в директории
cat mydata/*
hadoop fs -cat mydata/*
```

https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/FileSystemShell.html

Работа с HDFS (CLI)

Команда	Пример
appendToFile	hdfs dfs -appendToFile localfile /user/hadoop/hadoopfile
cat	hdfs dfs -cat hdfs://nn1.example.com/file1
copyFromLocal **copyToLocal**	hdfs dfs -copyFromLocal localfile /user/hadoop/data/ hdfs dfs -copyToLocal localfile /tmp/data/ localfile
cp	hdfs dfs -cp [-f] [-p | -p[topax]] URI [URI ...] <dest>
du	hdfs dfs -du -s /tmp/test.data
expunge	hdfs dfs -expunge
get **getmerge**	hdfs dfs -get /user/hadoop/file localfile hdfs dfs -getmerge <src> <localdst> [addnl]
ls	hdfs dfs -ls /user/hadoop/file1
mkdir	hdfs dfs -mkdir /user/hadoop/dir1 /user/hadoop/dir2
mv	hdfs dfs -mv /user/hadoop/file1 /user/hadoop/file2

https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/FileSystemShell.html

Работа с HDFS (CLI)

Команда	Пример
put	hdfs dfs -put localfile /user/hadoop/hadoopfile
rm	hdfs dfs -rm [-f] [-r|-R] [-skipTrash] URI [URI ...]
tail	hdfs dfs -tail pathname
setrep	hdfs dfs -setrep [-R] [-w] <numReplicas> <path>

https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/FileSystemShell.html

Права доступа

Команда	Пример
chmod	hdfs dfs chmod [-R] mode file
chgrp	hdfs dfs chgrp [-R] group file ...
chown	hdfs dfs chown [-R] [owner][:[group]] file ...

10

HDFS
WEB API

Команда	
CREATE	Создание и запись данных в файл
APPEND	Дописываение файла
CONCAT	Объединение файлов
OPEN	Открытие и чтение файла
MKDIRS	Создание каталога
RENAME	Переименование файла/каталога
DELETE	Удаление файла/каталога
GETFILESTATUS	Получение информации о файле/каталоге
LISTSTATUS	Просмотр информации о каталоге
…	

WEB API (пример)

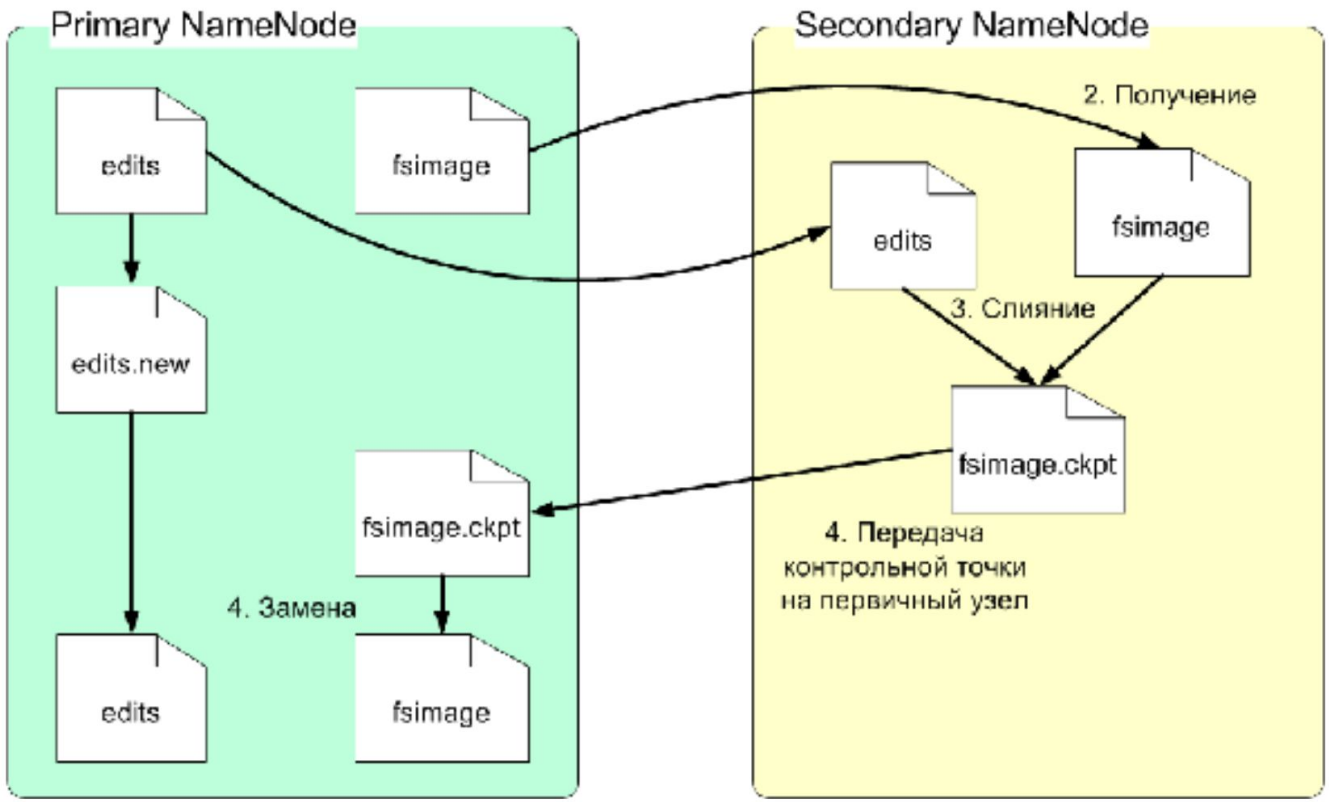
```
[root@hdp-15 ~]# curl -i "http://192.168.91.139:14000/webhdfs/v1/user/hive?
op=LISTSTATUS&user.name=hdfs"
{"FileStatuses":
        {"FileStatus":
        [{
                "pathSuffix":"user.csv",
                "type":"FILE",
                "length":22628,
                "owner":"hive",
                "group":"hive",
                "permission":"644",
                "accessTime":1429262046873,
                "modificationTime":1429262048992,
                "blockSize":134217728,
                "replication":3
        }]
        }
}
```

11

HDFS

Secondary NameNode

Secondary Name Node



12

HDFS

Точки отказа

Типы сбоев

- NameNode failure
- DataNode failure
- Network partitions

Вопросы?

Илья Маркин

Senior Software Engineer, insolar.io

amferiuss@gmail.com
@amferiuss

Заполните, пожалуйста, опрос о занятии

Спасибо
за внимание!