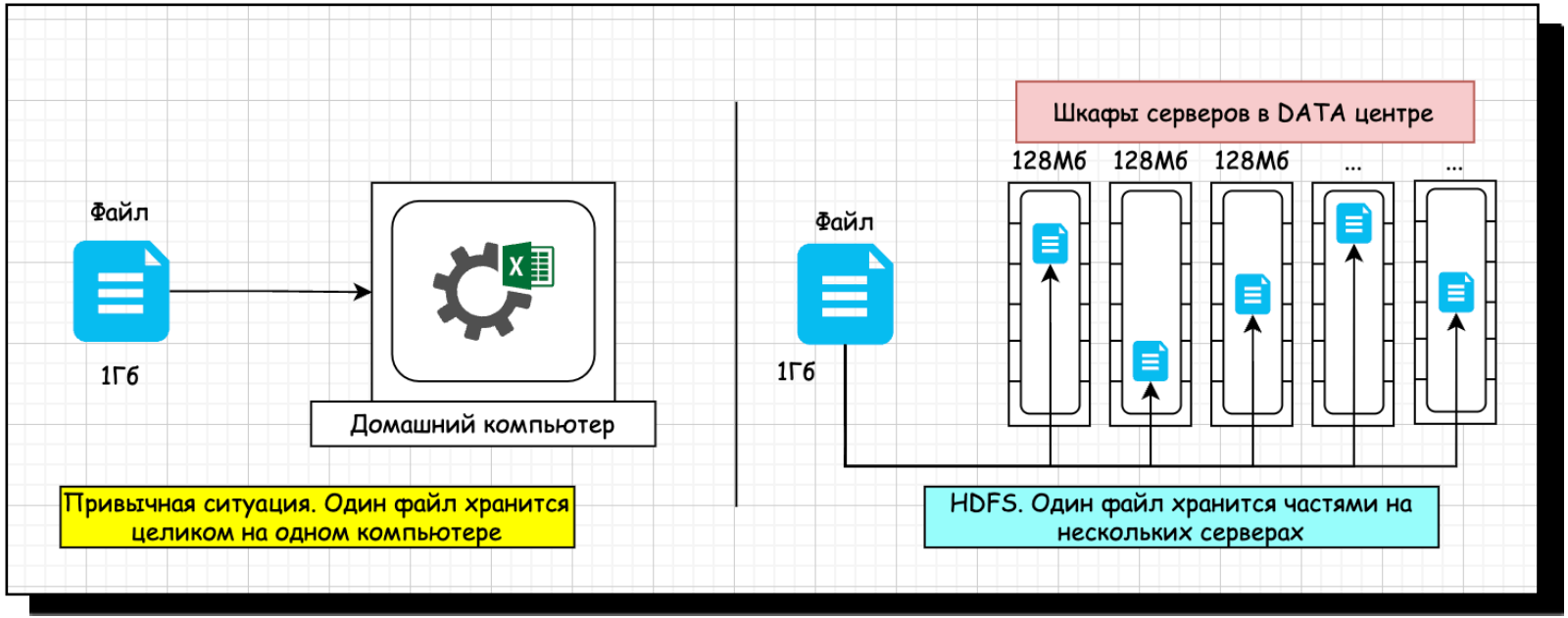
HDFS

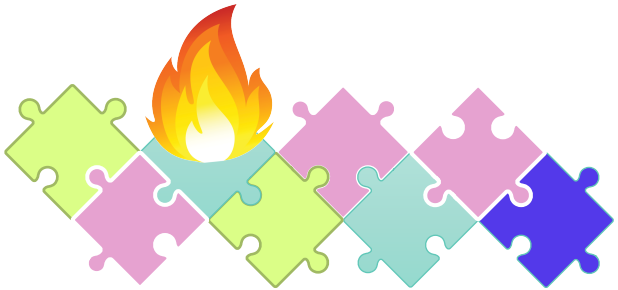
Hadoop Distributed File System

Блочное хранение

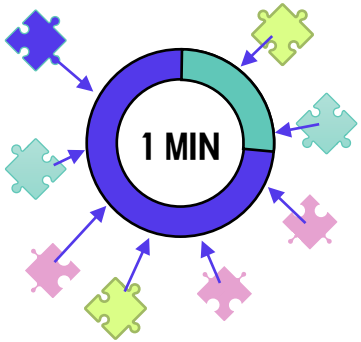


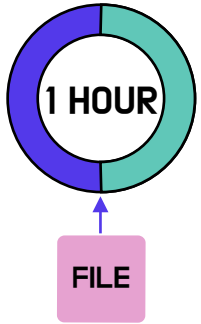
Зачем разбивать файл на блоки?

- Лучше сгорит один блок, чем сразу весь файл целиком



- Читать параллельно несколько мелких блоков быстрее, чем один, но огромный



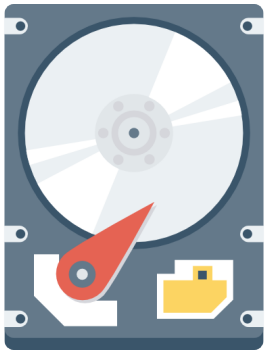


Каким должен быть размер блока файла?

Размер блока можно сделать любым! 1Kb, 1Gb, 100Tb, 1Mb

Время смены позиции головки диска = 10 ms

Скорость чтения диска = 100 Mb/s



1 файл размером 100Mb

1 sec + 0.01 sec = 1.01 sec ~ 1 sec

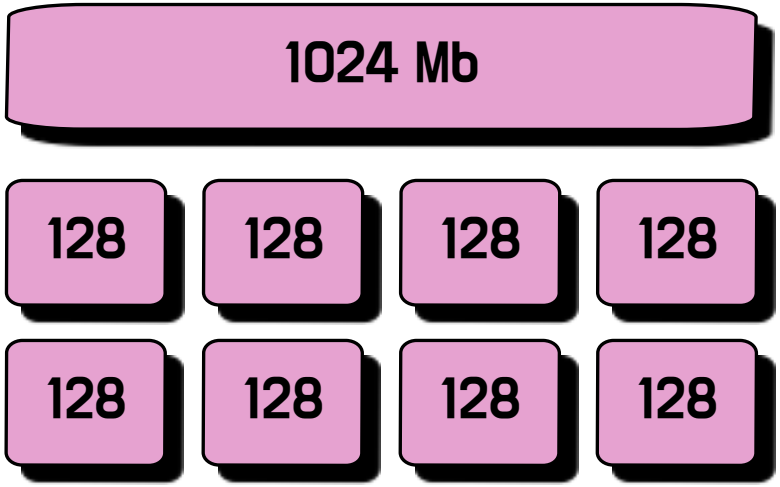
100 файлов по 1Mb

1 sec + 0.01 sec * 100 = 1 + 1 = 2 sec

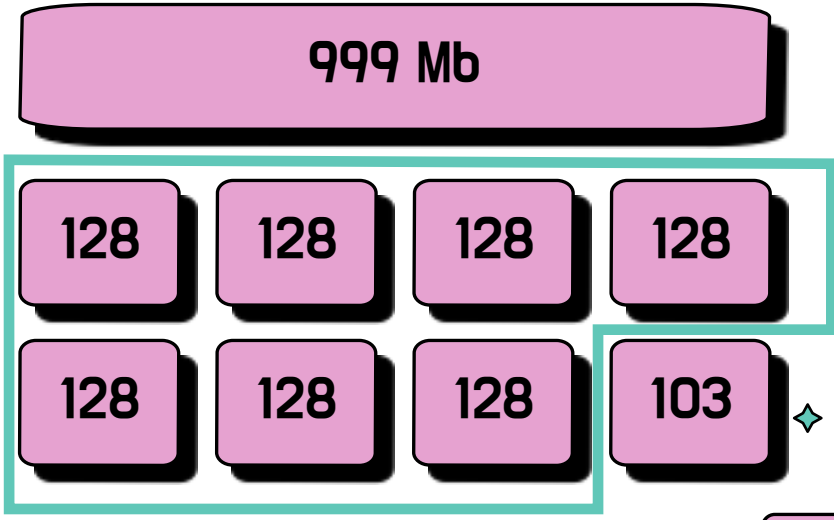
Делать размер блока слишком большим тоже не имеет смысла, так как мы не всегда читаем весь файл целиком, а только его часть

Как файл делится на блоки? Размер блока = 128 Mb

Файл 1024 Mb
8 равных блоков по 128 Mb



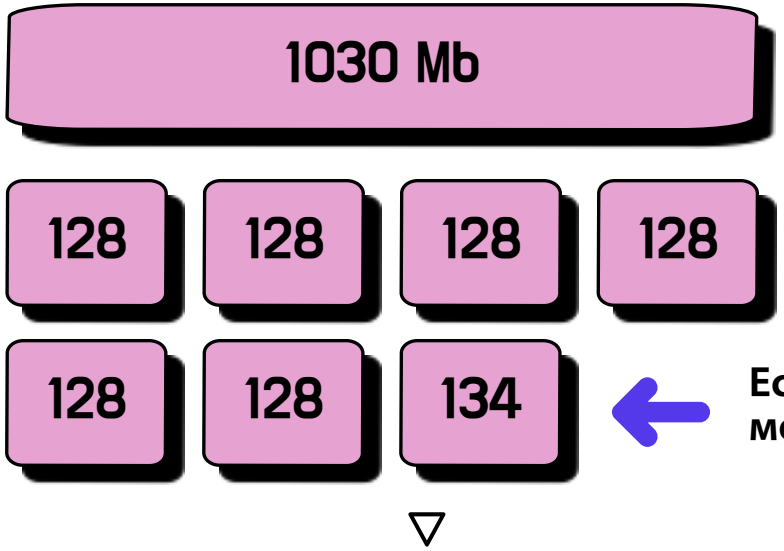
Файл 999 Mb
8 блоков



896 Mb + 103 Mb = 999 Mb

Как файл делится на блоки? Размер блока = 128 Mb

Файл 1030 Mb
8 блоков



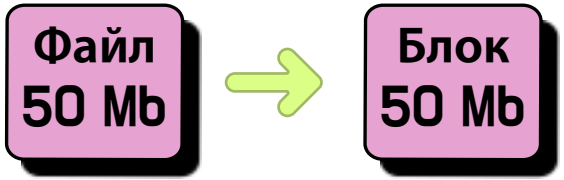
1030 Mb / 128 Mb = 8.046 блоков

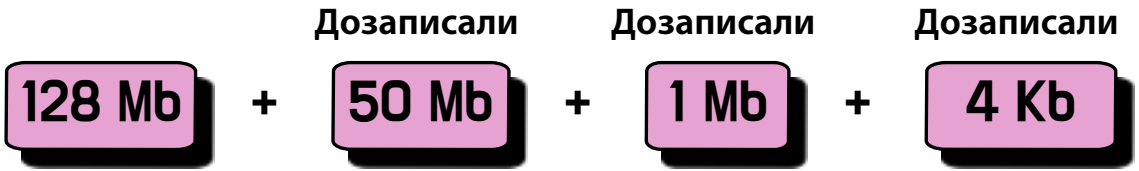
1030 Mb - 7 * 128 Mb = 6 Mb

Если размер хвостика меньше 1%, то он может быть приклеен к крайнему блоку

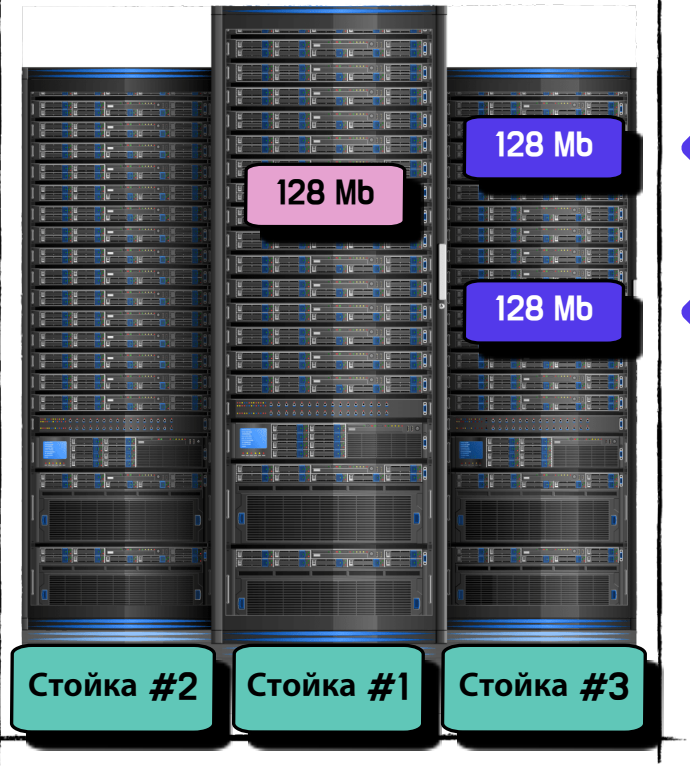
Как файл делится на блоки? Размер блока = 128 Mb

Если файл меньше размера блока, то размер блока = размеру файла





Репликация



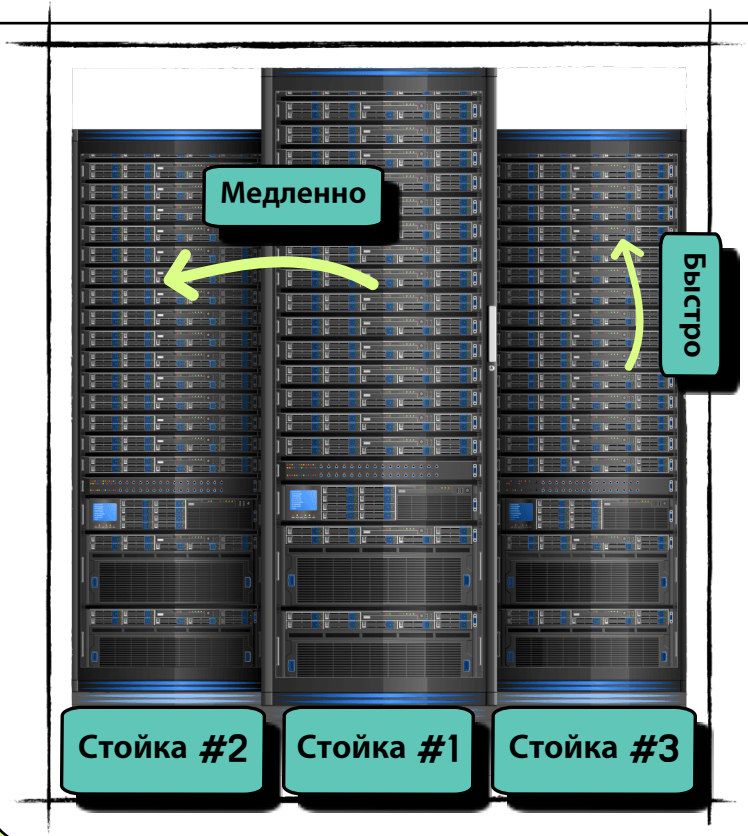
- Фактор репликации - 3 (можно менять)

Реплика #2

Реплика #3

При выходе из строя первой стойки (сегмента), данные не потеряются, потому что их копии есть на соседней стойке

Топология сети



Зачем HDFS знать топологию сети:

- Система знает, где хранятся блоки
- Оптимизация вычислений

Пропускная способность между стойками меньше, чем между узлами одной стойки

Важно для тех, кто использует несколько стоек в работе

Расстояние между узлами = Сумма расстояний до общего предка

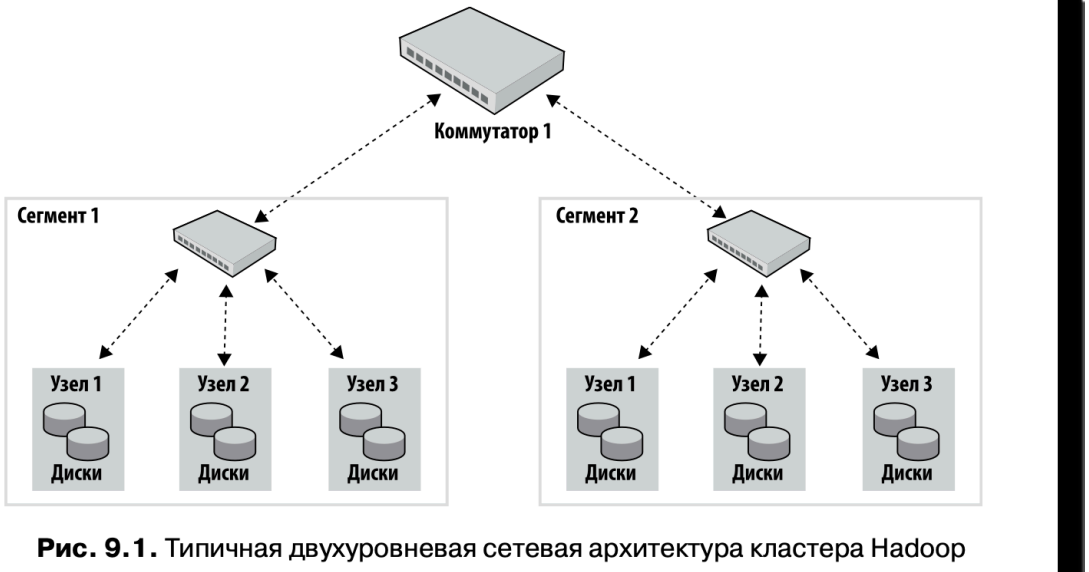
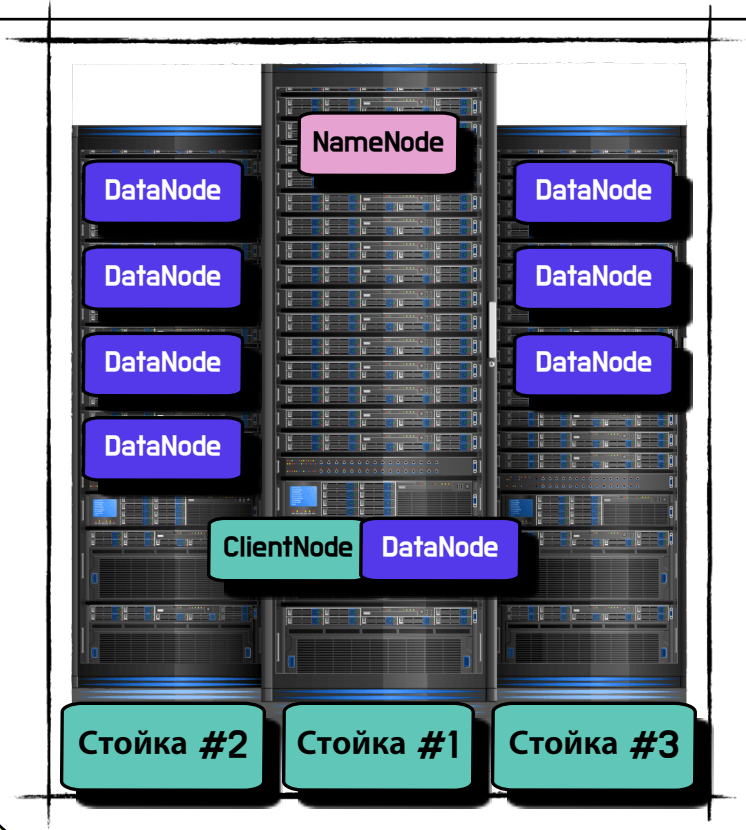


Рис. 9.1. Типичная двухуровневая сетевая архитектура кластера Hadoop

NameNode - DataNode - ClientNode



NameNode, DataNode, ClientNode – это все сервера

NameNode — Хранит данные о том, где хранятся блоки, разрешения, а также снимки файловой системы и лог изменений

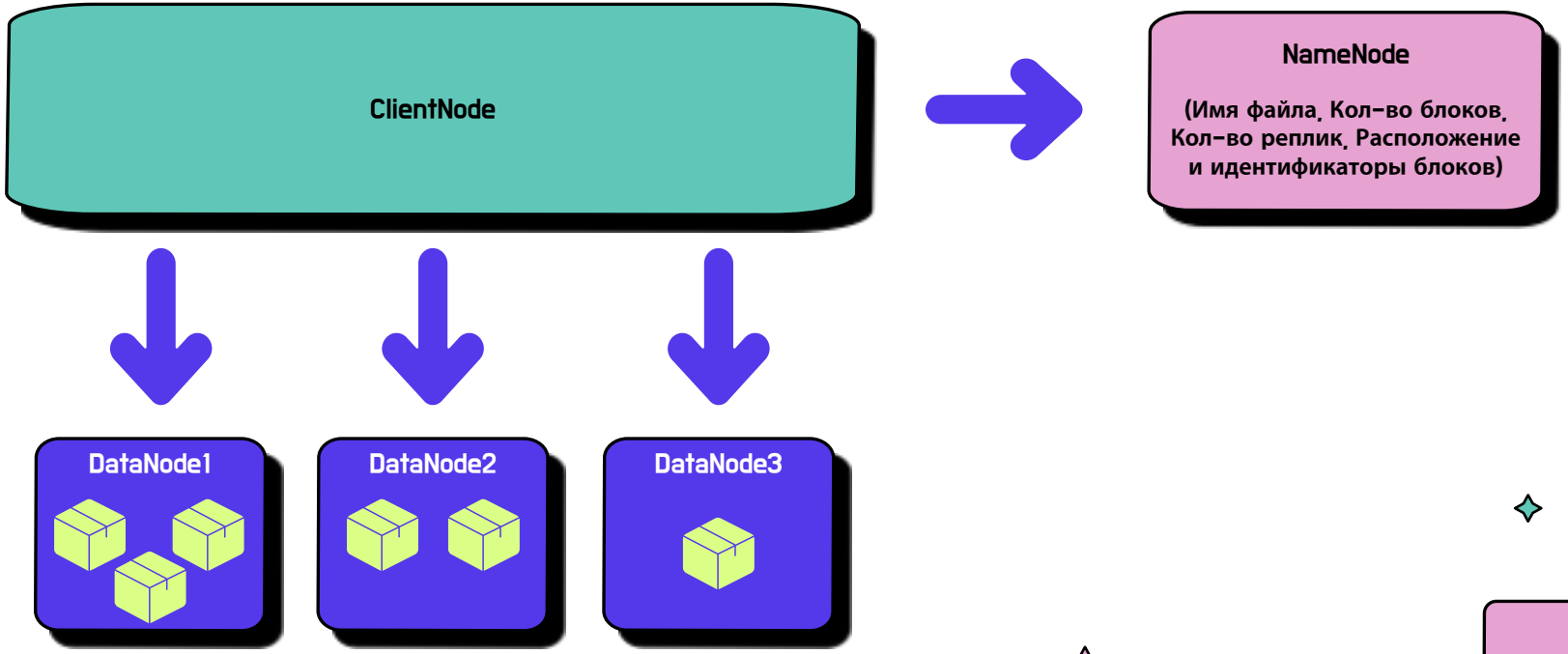
Если NameNode сгорит, то все данные потеряются

DataNode — Хранит сами блоки. Сама пишет и реплицирует блоки

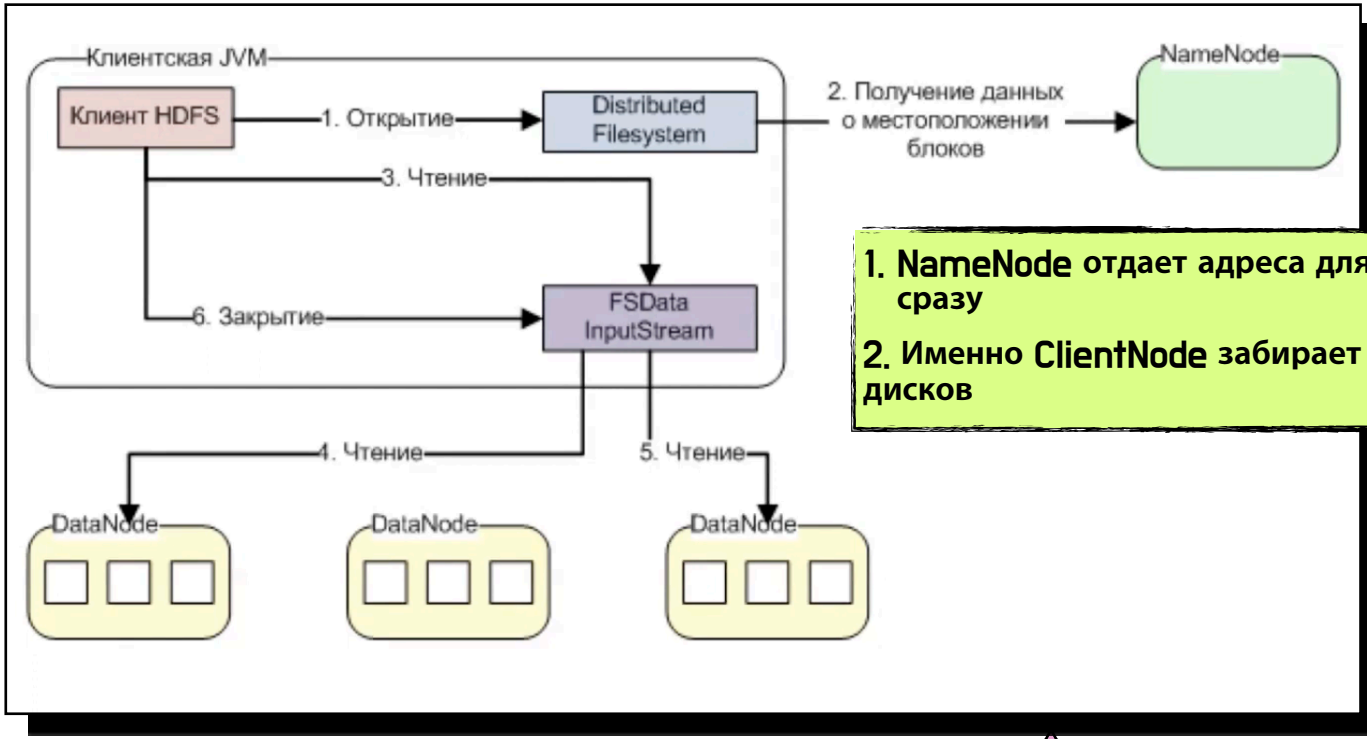
ClientNode — Управляется пользователем для чтения, записи, удаления данных с DataNode

NameNode - DataNode - ClientNode

Получение данных о местоположении блоков на DataNode

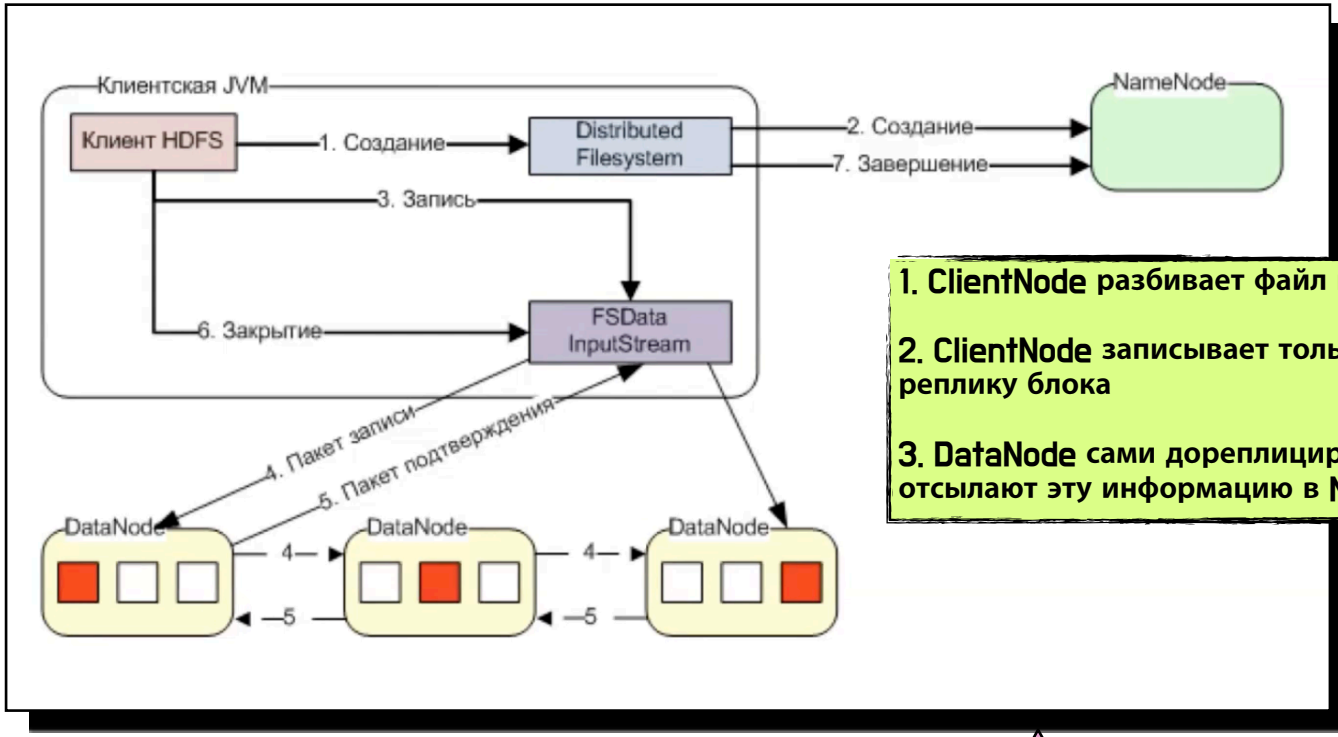


Чтение в HDFS



1. **NameNode** отдает адреса для всех реплик сразу
2. Именно **ClientNode** забирает данные с дисков

Запись в HDFS



1. **ClientNode** разбивает файл на блоки

2. **ClientNode** записывает только первую реплику блока

3. **DataNode** сами дореплицируют блоки и отсылают эту информацию в **NameNode**

Как выглядит блок на DataNode?

```
# hdfs fsck /path -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]
```

Хеш сумма нужна для проверки состояния блока.
Если хеш не совпадет, блок помечается как corrupted

blk_13434_234

1. hash_sum_1
2. hash_sum_2
3. hash_sum_3

DataBlockScanner

Каждые 504 часа проводится проверка блоков

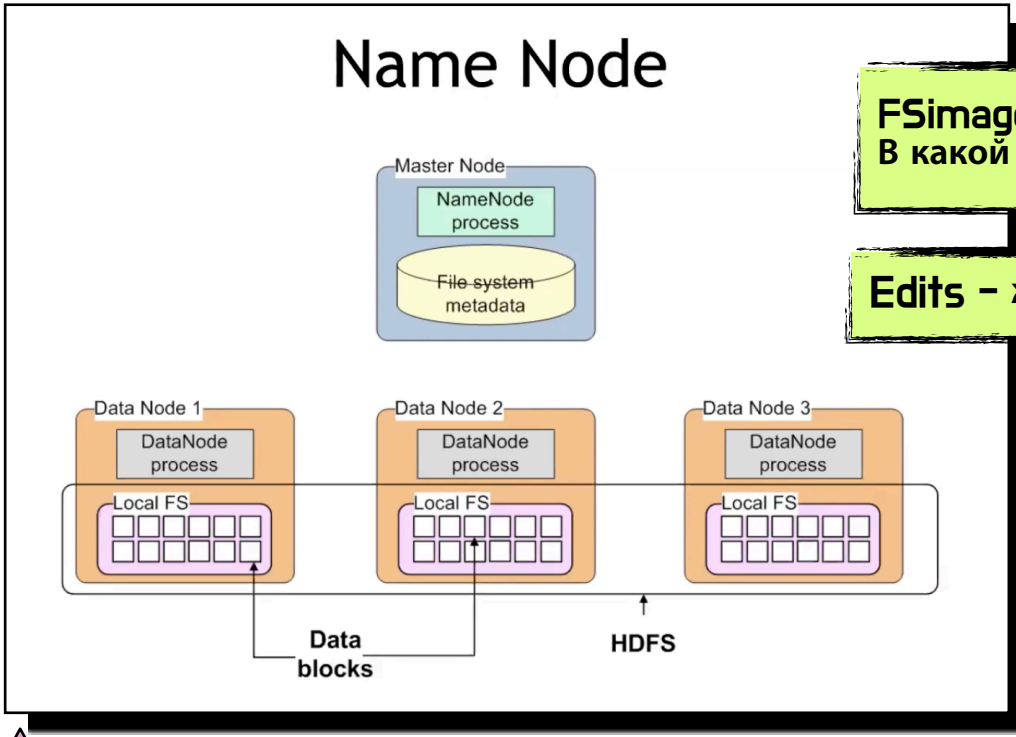
```
Total Blocks                :  21131
Verified in last hour       :     70
Verified in last day        :   1767
Verified in last week       :   7360
Verified in last four weeks :  20057
Verified in SCAN_PERIOD     :  20057
Not yet verified            :   1074
Verified since restart      :  35912
Scans since restart         :   6541
Scan errors since restart   :      0
Transient scan errors       :      0
Current scan rate limit KBps :   1024
Progress this period        :    109%
Time left in cur period     :  53.08%
```

```
blk_6035596358209321442    : status : ok     type : none
                             scan time : 0     not yet verified
blk_3065580480714947643    : status : ok     type : remote
                             scan time : 1215755306400  2008-07-11 05:48:26,400
blk_8729669677359108508    : status : ok     type : local
                             scan time : 1215755727345  2008-07-11 05:55:27,345
```

Если блок поврежден, то эта информация отправляется в NameNode и дальше она дает команду DataNode на пересоздание новой реплики

Поврежденный блок не удаляется. Его место теперь зарезервировано для новых блоков

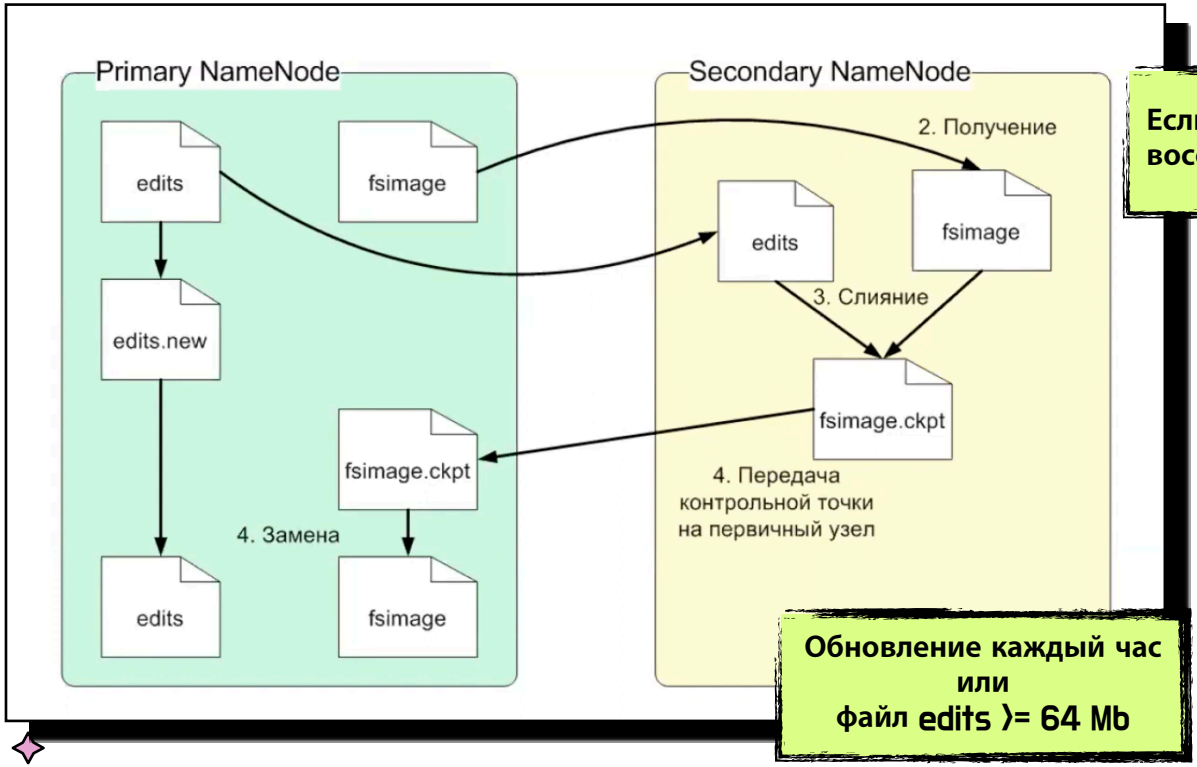
NameNode



FSimage – хранит структуру католожного дерева. В какой последовательности собирать блоки

Edits – журнал изменений

Secondary NameNode



Если NameNode сгорит, то можно восстановить файлы до какого-то периода

Обновление каждый час
или
файл edits >= 64 Mb

Что внутри NameNode?

```
${dfs.name.dir}/
└── current/
    ├── VERSION
    ├── edits
    ├── fsimage
    └── fstime
```

```
#Tue Mar 10 19:21:36 GMT 2009
namespaceID=134368441
cTime=0
storageType=NAME_NODE
layoutVersion=-18
```

edits – журнал изменений

fsimage – хранит структуру католожного дерева. В какой последовательности собирать блоки

fstime – время создания контрольной точки

Что внутри NameNode?

edits_0000000000111294742-0000000000111294767	3 KB	09.02.2024 16:41:44
edits_0000000000111294768-0000000000111294775	1 KB	09.02.2024 16:43:44
edits_0000000000111294776-0000000000111294801	3 KB	09.02.2024 16:45:45
edits_0000000000111294802-0000000000111294827	3 KB	09.02.2024 16:47:46
edits_0000000000111294828-0000000000111294835	1 KB	09.02.2024 16:49:46
edits_0000000000111294836-0000000000111294861	3 KB	09.02.2024 16:51:47
edits_0000000000111294862-0000000000111294888	3 KB	09.02.2024 16:53:47
edits_0000000000111294889-0000000000111294899	2 KB	09.02.2024 16:55:48
edits_inprogress_0000000000111294900	1 024 KB	09.02.2024 16:55:59
fsimage_0000000000111288151	44 457 KB	09.02.2024 7:48:06
fsimage_0000000000111288151.md5	1 KB	09.02.2024 7:48:06
fsimage_0000000000111292002	44 480 KB	09.02.2024 13:48:08
fsimage_0000000000111292002.md5	1 KB	09.02.2024 13:48:08
seen_txid	1 KB	09.02.2024 16:55:48
VERSION	1 KB	17.12.2023 7:43:18

Что внутри Secondary NameNode?

```
${fs.checkpoint.dir}/
├── current/
│   ├── VERSION
│   ├── edits
│   ├── fsimage
│   └── fstime
└── previous.checkpoint/
    ├── VERSION
    ├── edits
    ├── fsimage
    └── fstime
```

Содержит предыдущий checkpoint

Можно использовать, как основную NameNode

Что внутри DataNode?

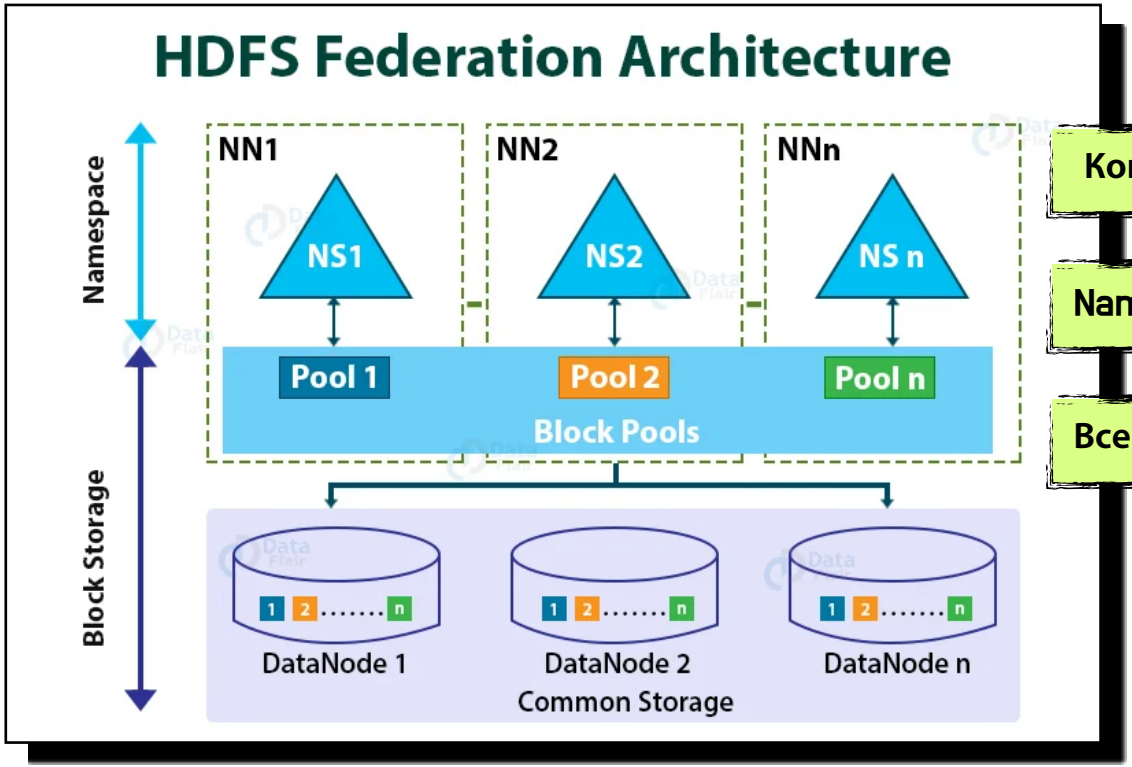
```
${dfs.data.dir}/
└── current/
    ├── VERSION
    ├── blk_<id_1>
    ├── blk_<id_1>.meta
    ├── blk_<id_2>
    ├── blk_<id_2>.meta
    ├── ...
    ├── blk_<id_64>
    ├── blk_<id_64>.meta
    ├── subdir0/
    ├── subdir1/
    ├── ...
    └── subdir63/
```

```
#Tue Mar 10 21:32:31 GMT 2009
namespaceID=134368441
storageID=DS-547717739-172.16.85.1-50010-1236720751627
cTime=0
storageType=DATA_NODE
layoutVersion=-18
```

blk_<id_1> – сам блок от файла
blk_<id_1>.meta – метаданные с контрольными суммами

В одном каталоге не более 64 блоков

HDFS Federation



Когда памяти одной NameNode мало

NameNode между собой никак не связаны

Все DataNode - как единое хранилище

Проблема мелких файлов в HDFS

File size >= BlockSize

Файл	Размер блока	Кол-во блоков	Размер метаданных на NameNode (1 запись / 1 Mb для упрощения)
1 файл 500 Mb	100 Mb	5	5 Mb
500 файлов по 1 Mb	1 Mb	500	500 Mb

```
!hdfs fsck    /user/cristoffer_nollan/data.csv          -files -blocks -locations
Last executed at 2024-01-23 14:28:45 in 4.42s
Connecting to namenode via    http://super_server.ru:12345/

FSCK started by          halltape          (auth:KERBEROS_SSL) from /10.114.24.133 for path       /user/cristoffer_nollan/data.csv       at Tue Jan 2
3 14:28:45 MSK 2024
 /user/cristoffer_nollan/data.csv            169112385 bytes, replicated: replication=2, 2 block(s):  OK
0. BP-362455954-10.114.24.133-1630329010090:blk_1081818192_8264647 len=134217728 Live_repl=2  [DatanodeInfoWithStorage[10.114.24.147:1019,
DS-cebb1823-fee2-4391-a9e5-d5d9a0a8b96a,DISK], DatanodeInfoWithStorage[10.114.24.128:1019,DS-19467f46-7596-4f68-b86e-a684fe175fbb,DISK]]
1. BP-362455954-10.114.24.133-1630329010090:blk_1081818193_8264648 len=34894657 Live_repl=2  [DatanodeInfoWithStorage[10.114.24.128:1019,D
S-19467f46-7596-4f68-b86e-a684fe175fbb,DISK], DatanodeInfoWithStorage[10.114.24.149:1019,DS-48943f09-32c8-4b4b-996f-ffdef6185bdc,DISK]]


Status: HEALTHY
 Number of data-nodes:  9
 Number of racks:               1
 Total dirs:                    0
 Total symlinks:                0

Replicated Blocks:
 Total size:    169112385 B
 Total files:   1
 Total blocks (validated):      2 (avg. block size 84556192 B)
 Minimally replicated blocks:   2 (100.0 %)
 Over-replicated blocks:        0 (0.0 %)
 Under-replicated blocks:       0 (0.0 %)
 Mis-replicated blocks:         0 (0.0 %)
 Default replication factor:    2
 Average block replication:     2.0
 Missing blocks:                0
 Corrupt blocks:                0
 Missing replicas:              0 (0.0 %)
```

Файл 161 Mb поделился на два блока
128 Mb + 33Mb = 161Mb

Команда hdfs fsck -files -blocks -locations

Проблема мелких файлов в HDFS

```
Status: HEALTHY
 Number of data-nodes:  75
 Number of racks:               1
 Total dirs:                    1
 Total symlinks:                0

Replicated Blocks:
 Total size:    520543075511 B
 Total files:   27612
 Total blocks (validated):      27611 (avg. block size 18852742 B)
 Minimally replicated blocks:   27611 (100.0 %)
 Over-replicated blocks:        0 (0.0 %)
 Under-replicated blocks:       0 (0.0 %)
 Mis-replicated blocks:         0 (0.0 %)
 Default replication factor:    2
 Average block replication:     2.0
 Missing blocks:                0
 Corrupt blocks:                0
 Missing replicas:              0 (0.0 %)
```

11 522 045 162 rows

484 Gb

27611 files

COUNT(*) 2m 39s

!hdfs fsck -blocks -locations

Проблема мелких файлов в HDFS [coalesce]

```
Status: HEALTHY
 Number of data-nodes:  75
 Number of racks:               1
 Total dirs:                    1
 Total symlinks:                0

Replicated Blocks:
 Total size:    550310331017 B
 Total files:   4445
 Total blocks (validated):      6037 (avg. block size 91156258 B)
 Minimally replicated blocks:   6037 (100.0 %)
 Over-replicated blocks:        0 (0.0 %)
 Under-replicated blocks:       0 (0.0 %)
 Mis-replicated blocks:         0 (0.0 %)
 Default replication factor:    2
 Average block replication:     2.0
 Missing blocks:                0
 Corrupt blocks:                0
 Missing replicas:              0 (0.0 %)
```

11 522 045 162 rows

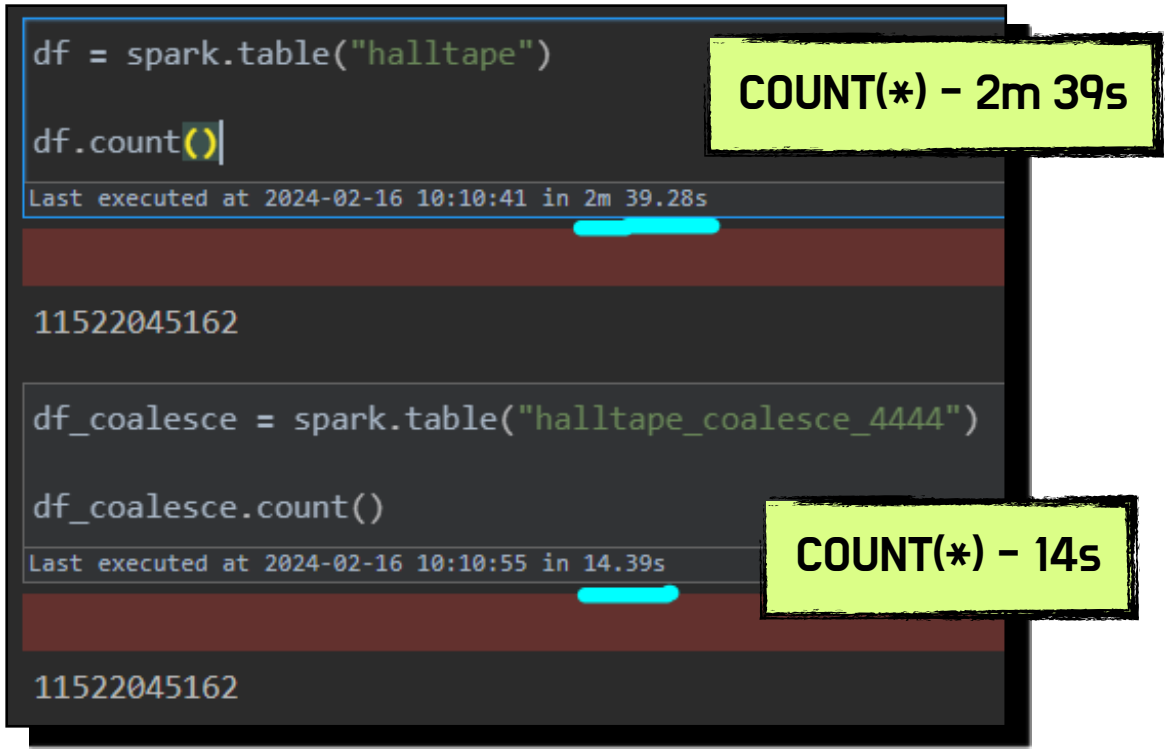
512 Gb

6037 files

COUNT(*) 14s

!hdfs fsck -blocks -locations

Проблема мелких файлов в HDFS [coalesce]



Спасибо за внимание!