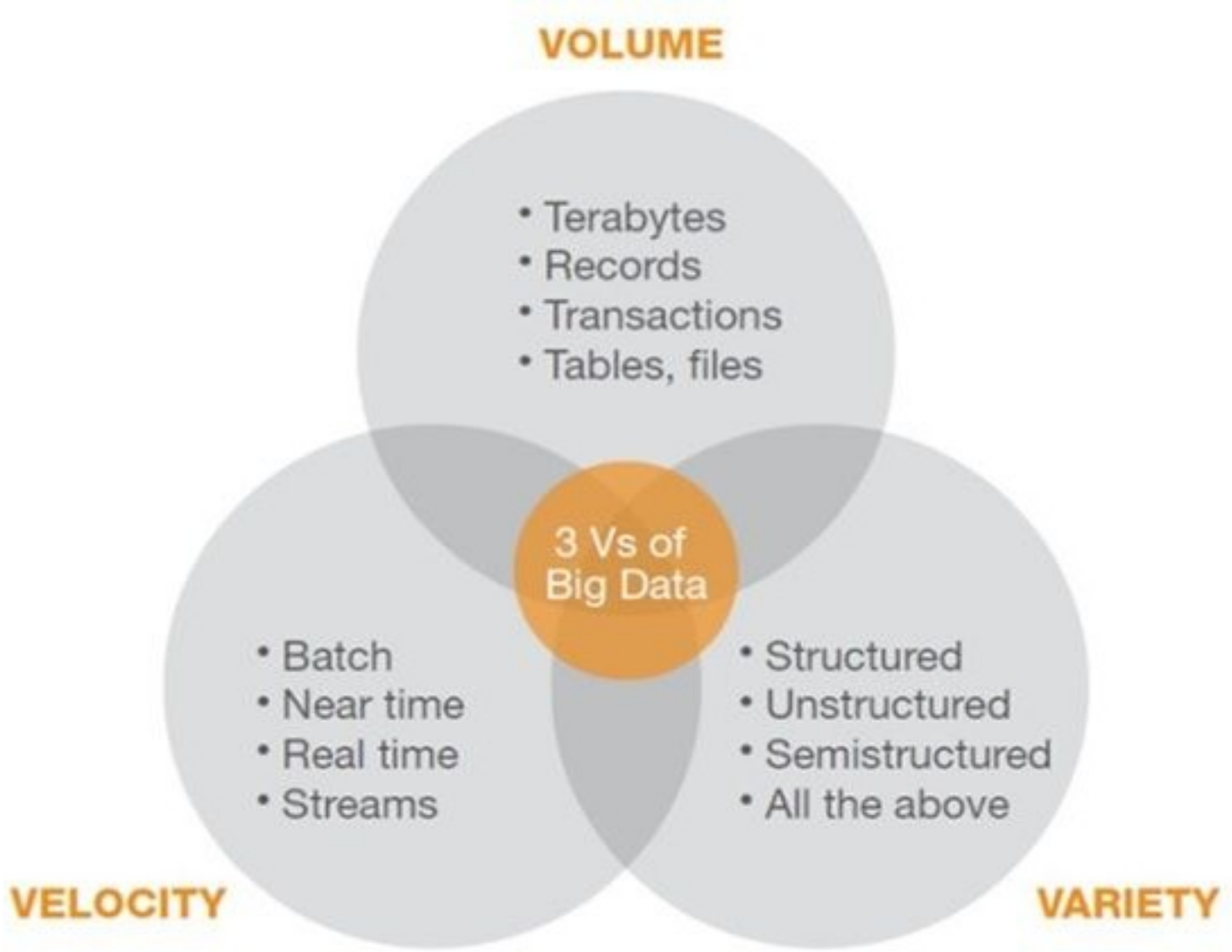
Лекция. Программная платформа Hadoop.
Распределенная файловая система HDFS

Основные темы

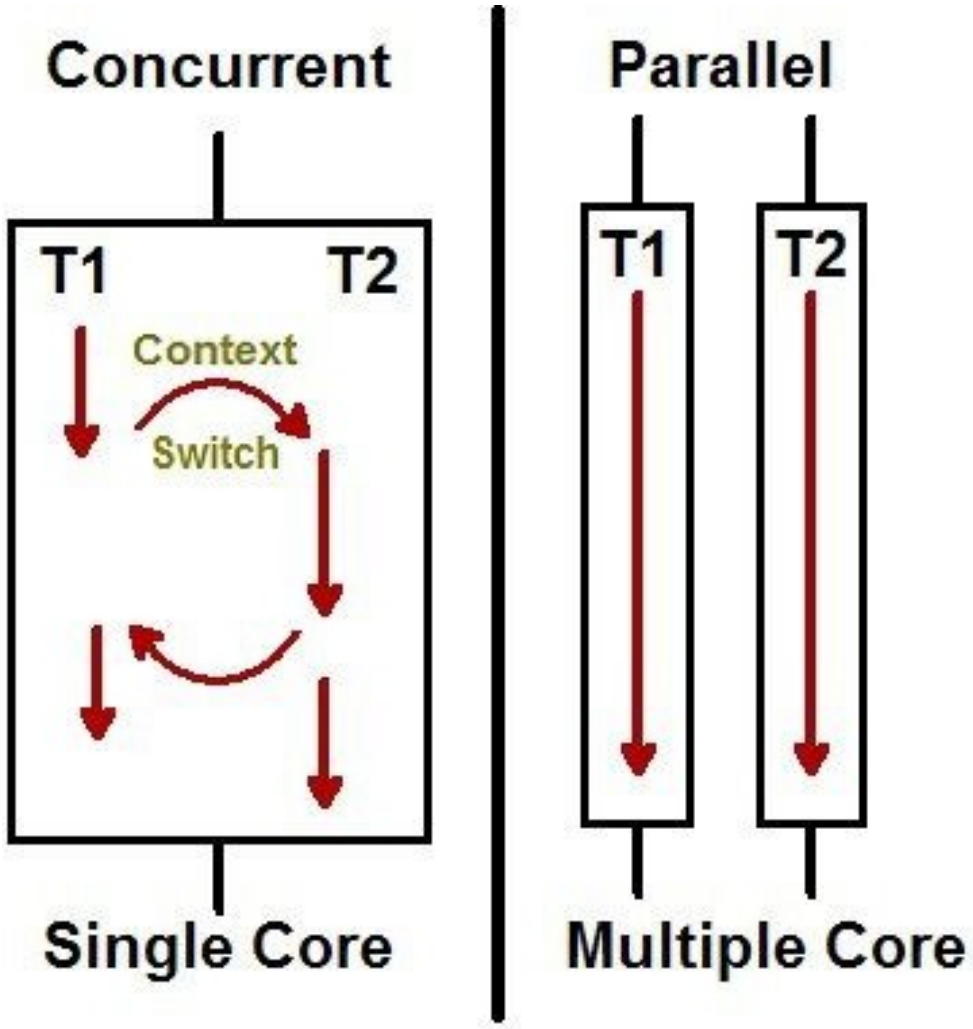
- Hadoop как система распределенной обработки данных
- Назначение и отличительные особенности HDFS
- Основные компоненты HDFS
- Операции чтение/запись в HDFS
- HDFS HA
- Hadoop 3.x – HDFS

Hadoop как система распределенной обработки данных

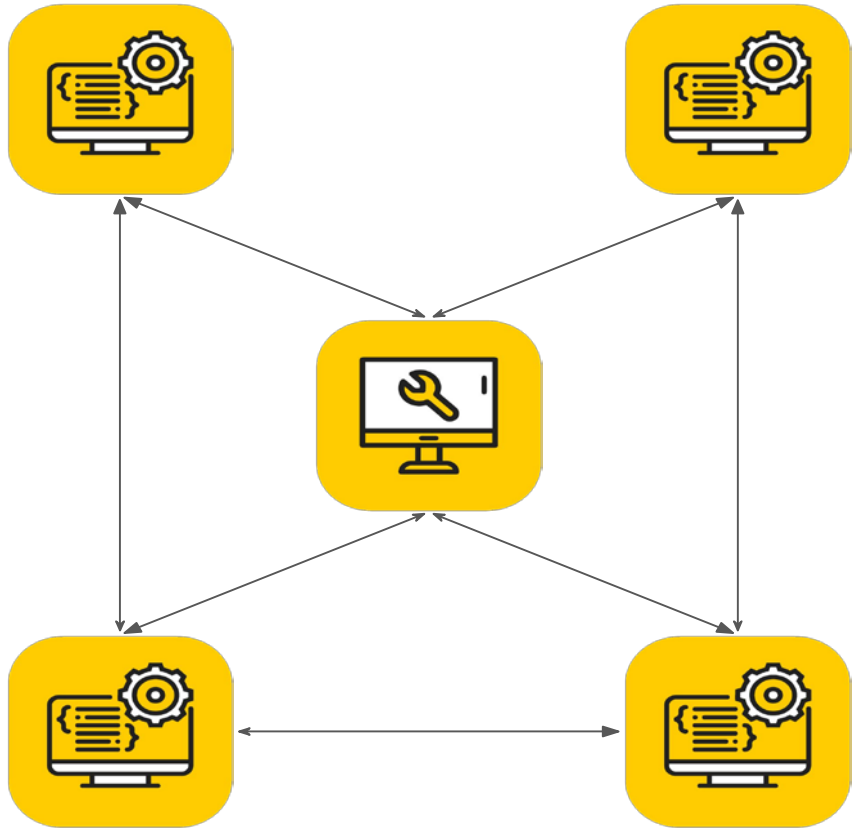
- Большой объем данных трудно хранить и анализировать на одной машине
- Excel - максимум 1 млн строк
- Другие инструменты: ограничения RAM



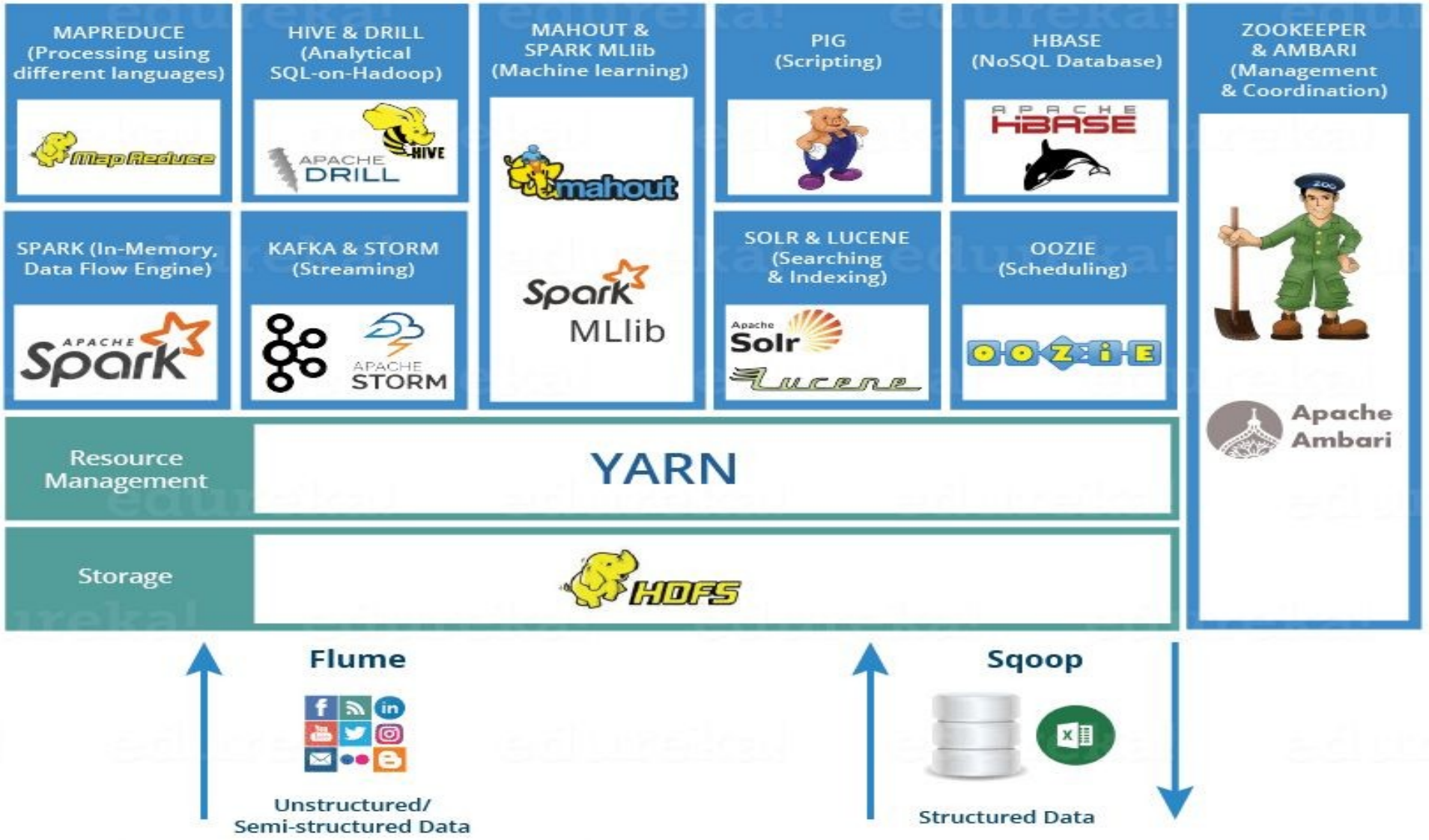
- **Конкурентность:**
  - Обеспечивается за счет механизма потоков (threads)
- **Параллелизм:**
  - Выполнение нескольких задач одновременно за счет многоядерности



- Задачи должны быть разделены на независимые друг от друга подзадачи (≈ функциональное программирование)
- Один из компьютеров координирует работу вычислительных узлов
- Hadoop - один из примеров системы распределенных вычислений



Экосистема Hadoop



Особенности Hadoop

- Масштабируемость
- Отказоустойчивость
- Доступность
- Вычисления приближены к данными
- Высокая пропускная способность I/O

Распределенная файловая система Hadoop (HDFS)

HDFS

Распределенная файловая система Hadoop (Hadoop Distributed File System – **HDFS**) – файловая система, разработанная для хранения больших массивов данных и запускаемая на кластере с серверами общего назначения (commodity hardware)

- Когда данные не помещаются на локальный диск, их можно распределить на несколько узлов
- **Hadoop Distributed File System (HDFS)** -распределенная файловая система, позволяющая хранить очень большой объем данных (>100Gb до нескольких петабайт)
- Стратегия “write-once, read-many”
- Не требует специализированного железа

Для чего НЕ подходит HDFS

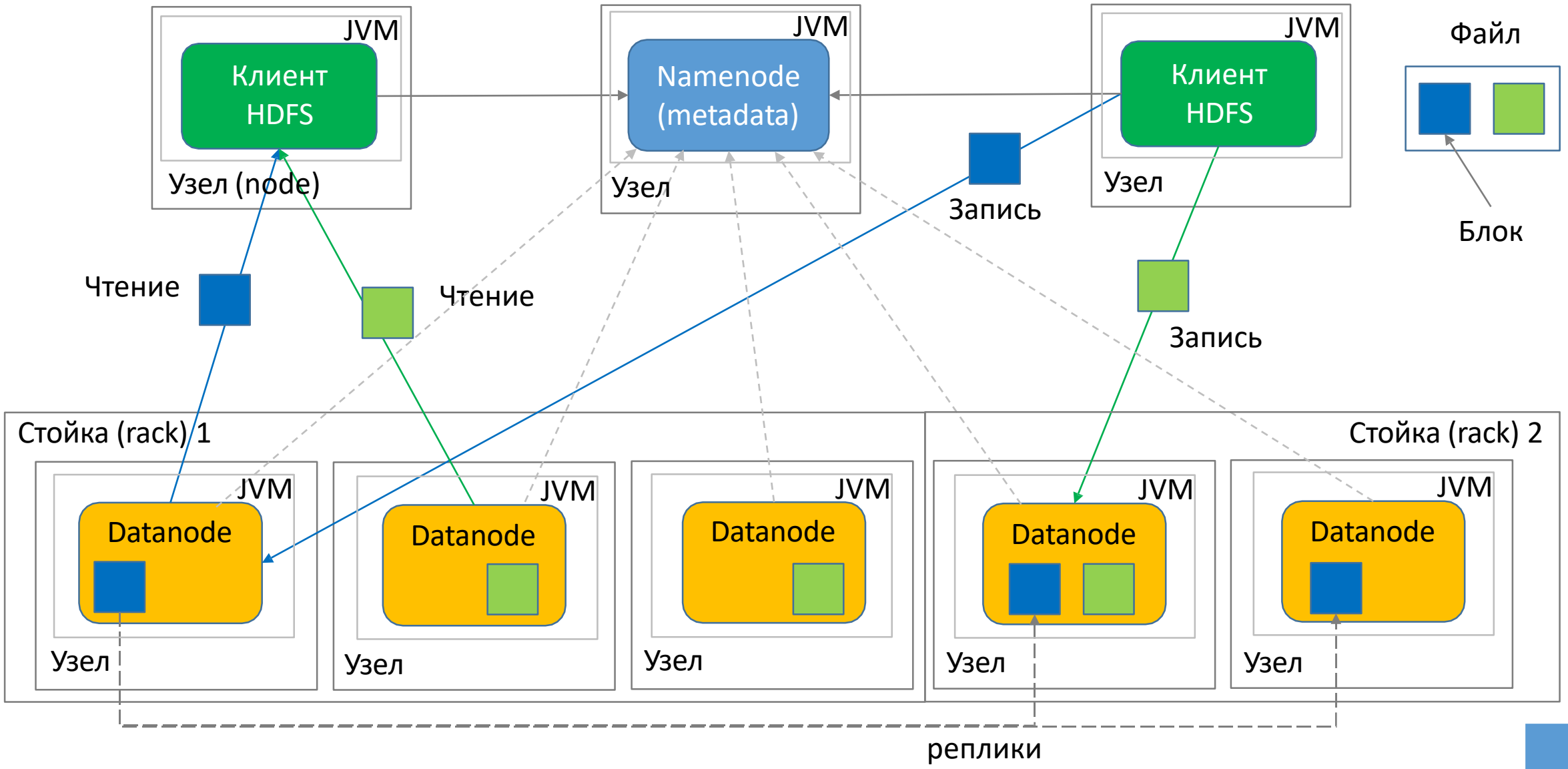
- Высокая скорость доступа к данным (например, OLTP)
- Хранение большого количества маленьких файлов (почему?)
- Конкурентное изменение файлов (single writer only)
- Произвольное изменение файлов (append only)

Основные компоненты HDFS

Словарь HDFS

- Данные
- Блок данных
- Клиент
- NameNode
- Secondary NameNode
- DataNode
- Реплика
- JournalNode
- Federation

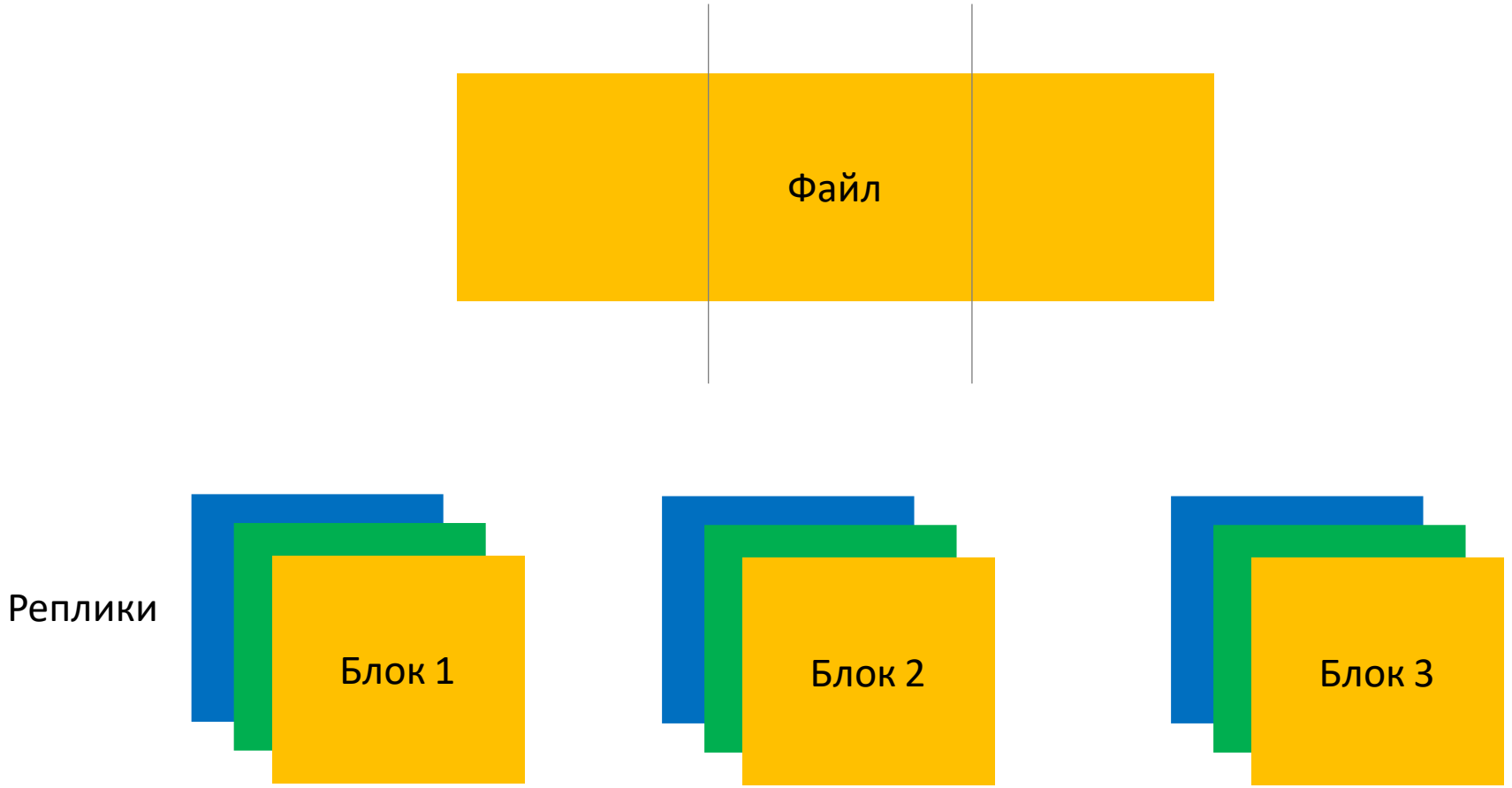
Архитектура HDFS



Блок данных

Блоки данных – объем данных для чтения/записи

64/128/256МБ



Блок данных

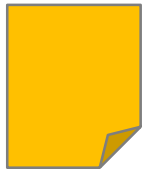
Зачем нужны блоки?

Если слишком маленький/большой размер блока?

Namenode

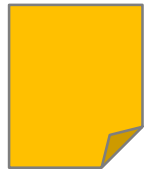
Namenode управляет пространством файловой системы и поддерживает

- дерево файловой системы
- метаданные о всех файлах и папках
- списки: файл -> блоки -> datanodes



Namespace Image

Fsimage



Edit Log

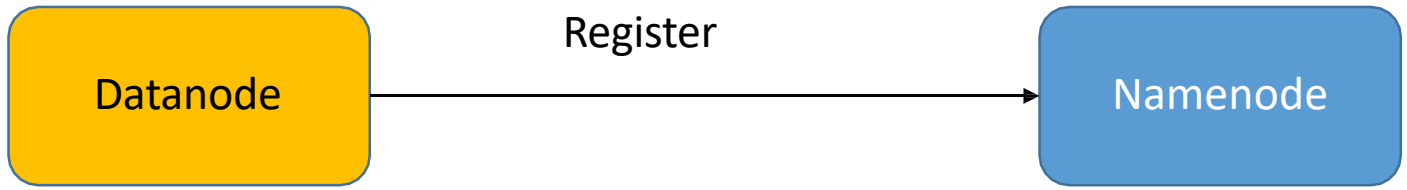
Datanode

Datanode – компонент HDFS для работы с блоками данных

- хранит и извлекает блоки по запросу
- передает namenode информацию о своих блоках
- периодически посылает namenode heartbeats

Кластер может иметь тысячи Datanodes и десятки тысяч клиентов.

Каждый Datanode может выполнять множество тасков приложений параллельно



Передаваемые сообщения. Block report и heartbeat

- **Heartbeats** от DN к NN (~ секунды – 3) передают следующую информацию:
  - Доступный объем дискового пространства
  - Сколько используется для хранения
  - Количество передаваемых данных в текущий момент

- **Block report** от DN к NN (~ часы – 1) содержит информацию о хранящихся репликах блоков данных:
  - Id блока,
  - Метка поколения
  - Размер каждого блока

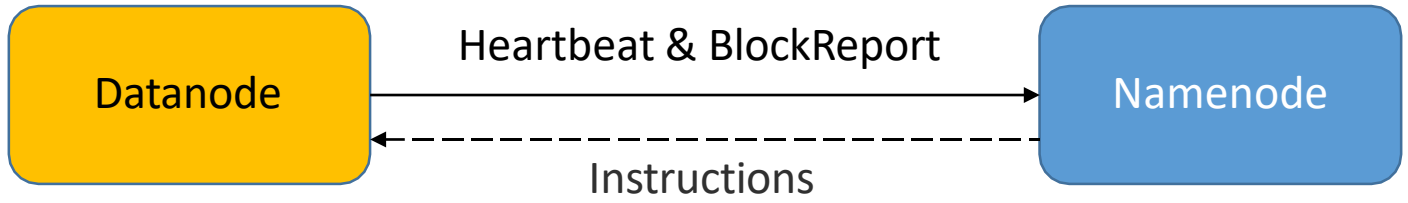
NN может обрабатывать тысячи hearbeat´ов в секунду без воздействия на другие операции NN

Управление DataNode через NameNode

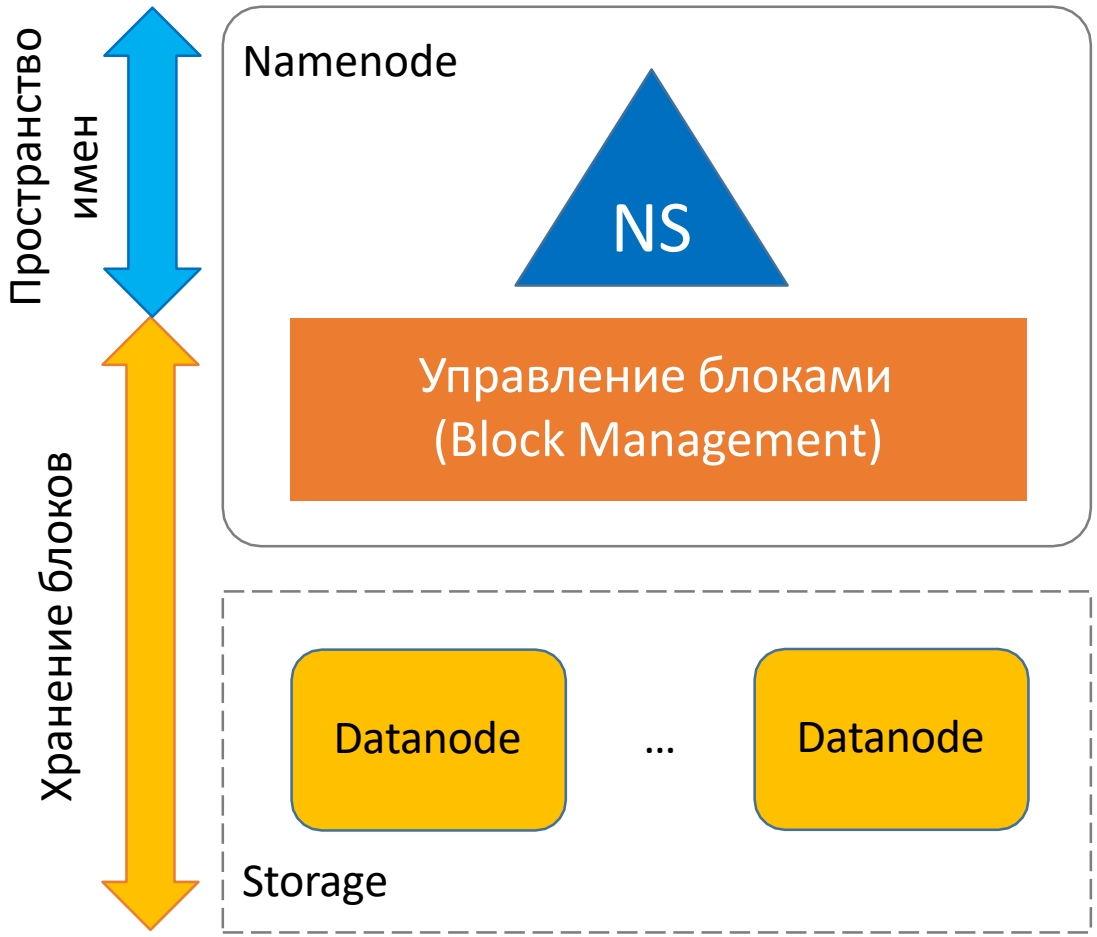
NameNode использует ответ на heartbeat’ы, чтобы отправлять **инструкции** на Datanode’ы.

Инструкции включают **команды** на:

- Репликацию блоков на другие узлы
- Удаление реплик;
- Повторную регистрацию и завершение работы Datanode;
- Отправку block report



Пространство имен и хранение блоков



Пространство имен (Namespace)

- директории, файлы и блоки
- создание, удаление, изменение, вывод список файлов и директорий.

Службы хранения блоков

Управление блоками

- Регистрация DN и обработка heartbeat'ов
- block reports и обработка информации о расположении блоков
- Операции над блоками: создание, удаление, изменение и получение расположения блока
- Управление репликами

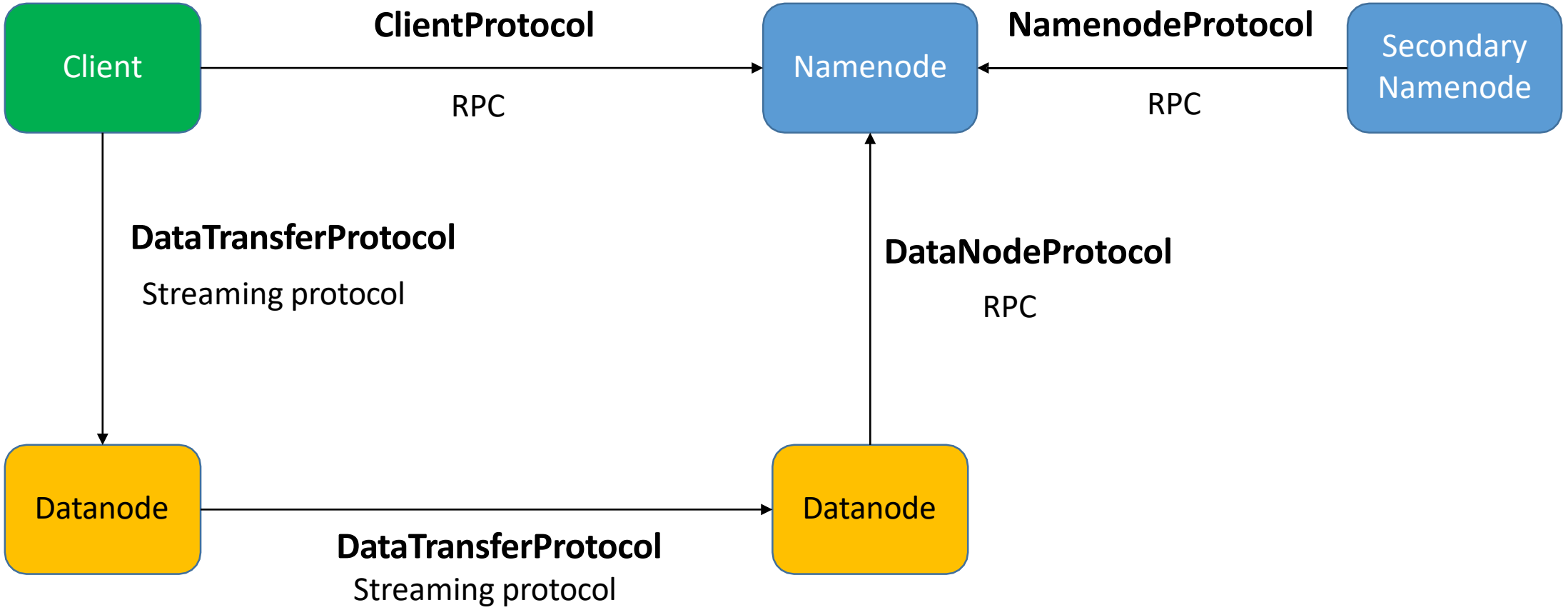
Хранение (Storage)

- Хранение блоков в локальной файловой системе и обеспечение доступа на чтение/запись

Сервисы/демоны в HDFS

Daemon	Environment Variable
NameNode	HADOOP_NAMENODE_OPTS
DataNode	HADOOP_DATANODE_OPTS
Secondary NameNode	HADOOP_SECONDARYNAMENODE_OPTS

Протоколы в HDFS



Основные команды в HDFS

“hadoop fs” vs “hdfs dfs”

hadoop fs

- Hadoop Distributed File System (HDFS)
- Local FS,
- HFTP FS,
- S3 FS, and others

Манипуляции с данными

hadoop fs -mkdir [-p] <paths>

hadoop fs -copyFromLocal <localsrc> URI

hadoop fs -ls [-d] [-h] [-R] <args>

hadoop fs -copyToLocal [-ignorecrc] [-crc] URI <localdst>

hadoop fs -cp [-f] [-p | -p[topax]] URI [URI ...] <dest>

hadoop fs -df [-h] URI [URI ...]

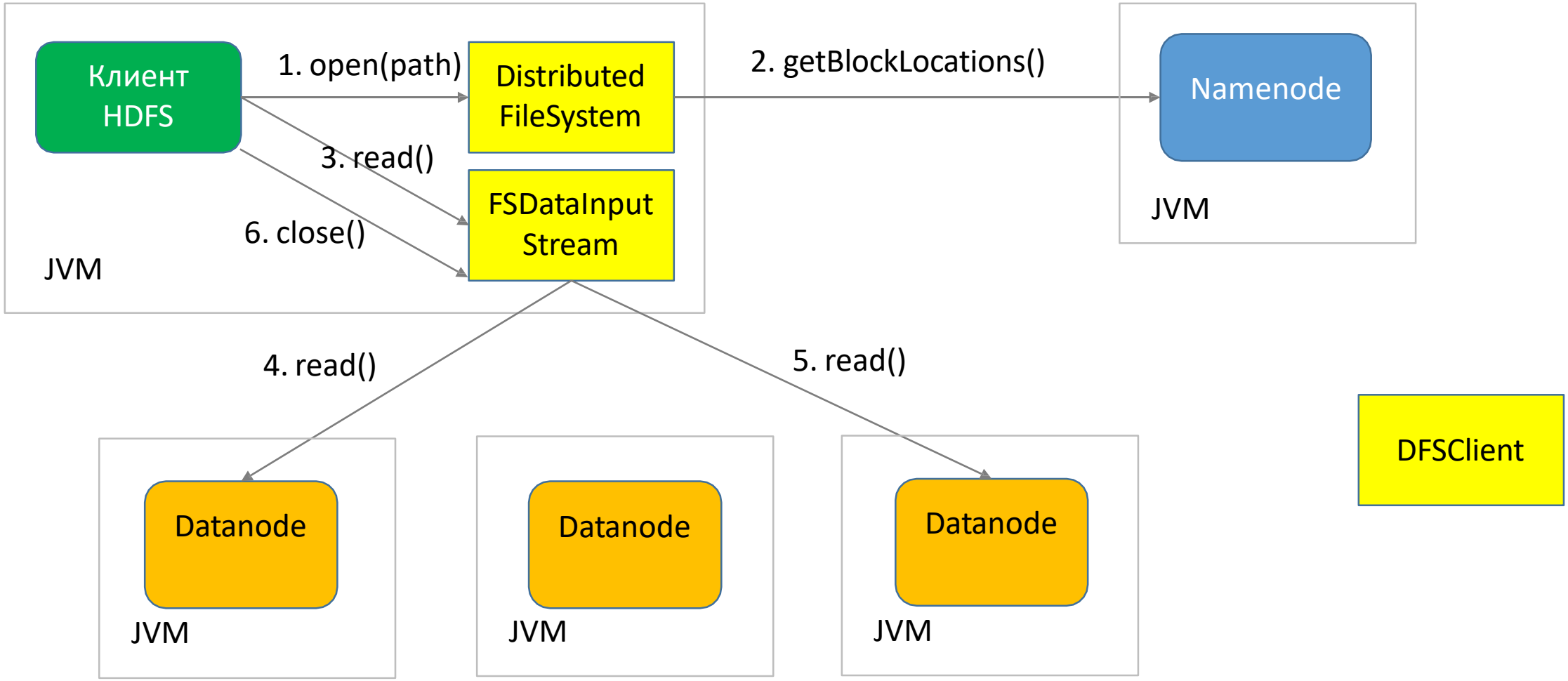
hadoop fs -du [-s] [-h] URI [URI ...]

https://hadoop.apache.org/docs/r2.7.2/hadoop-project-dist/hadoop-common/FileSystemShell.html

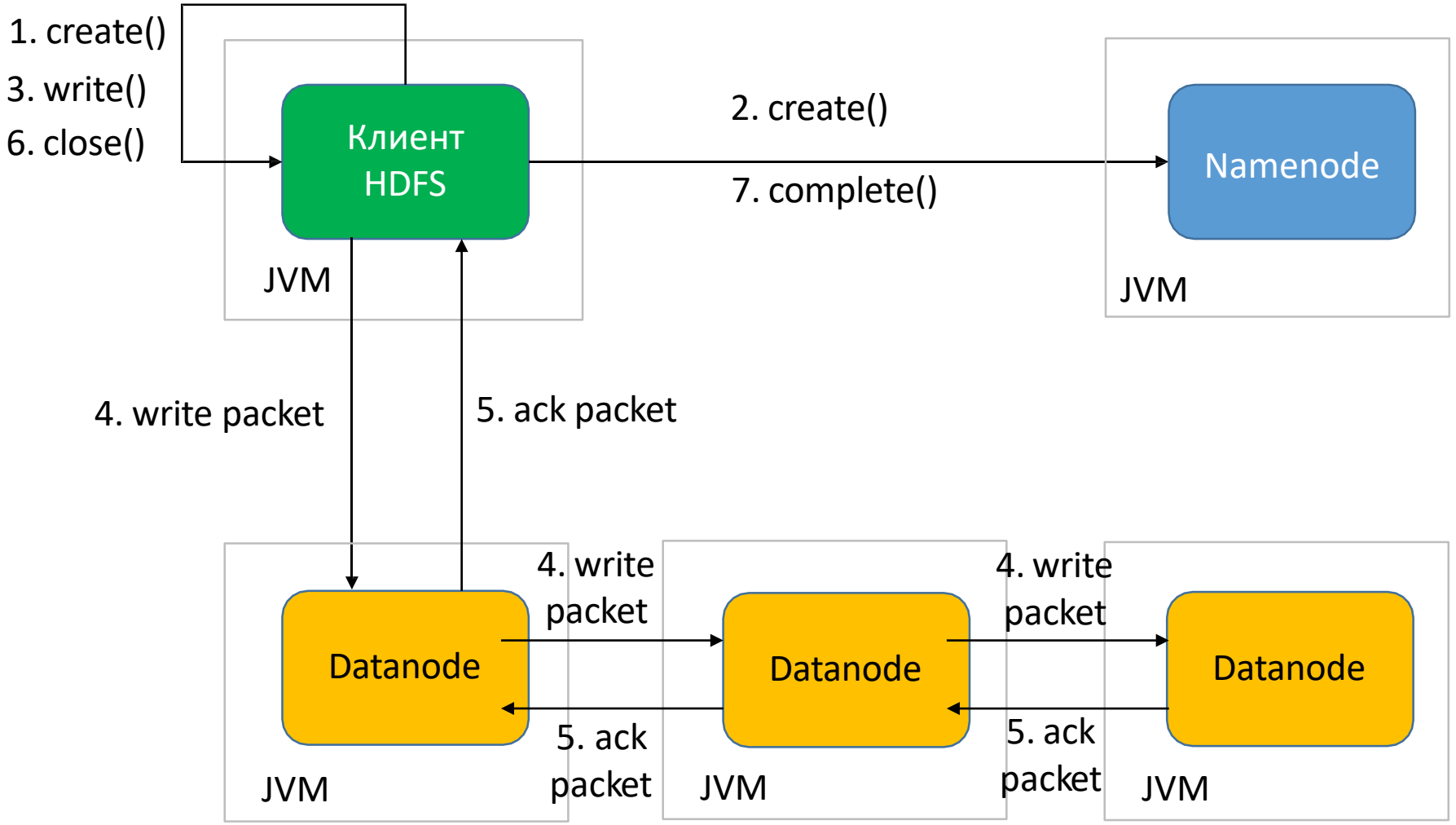
https://hadoop.apache.org/docs/r2.7.0/hadoop-project-dist/hadoop-hdfs/HDFSCommands.html

Операции чтение/запись в HDFS

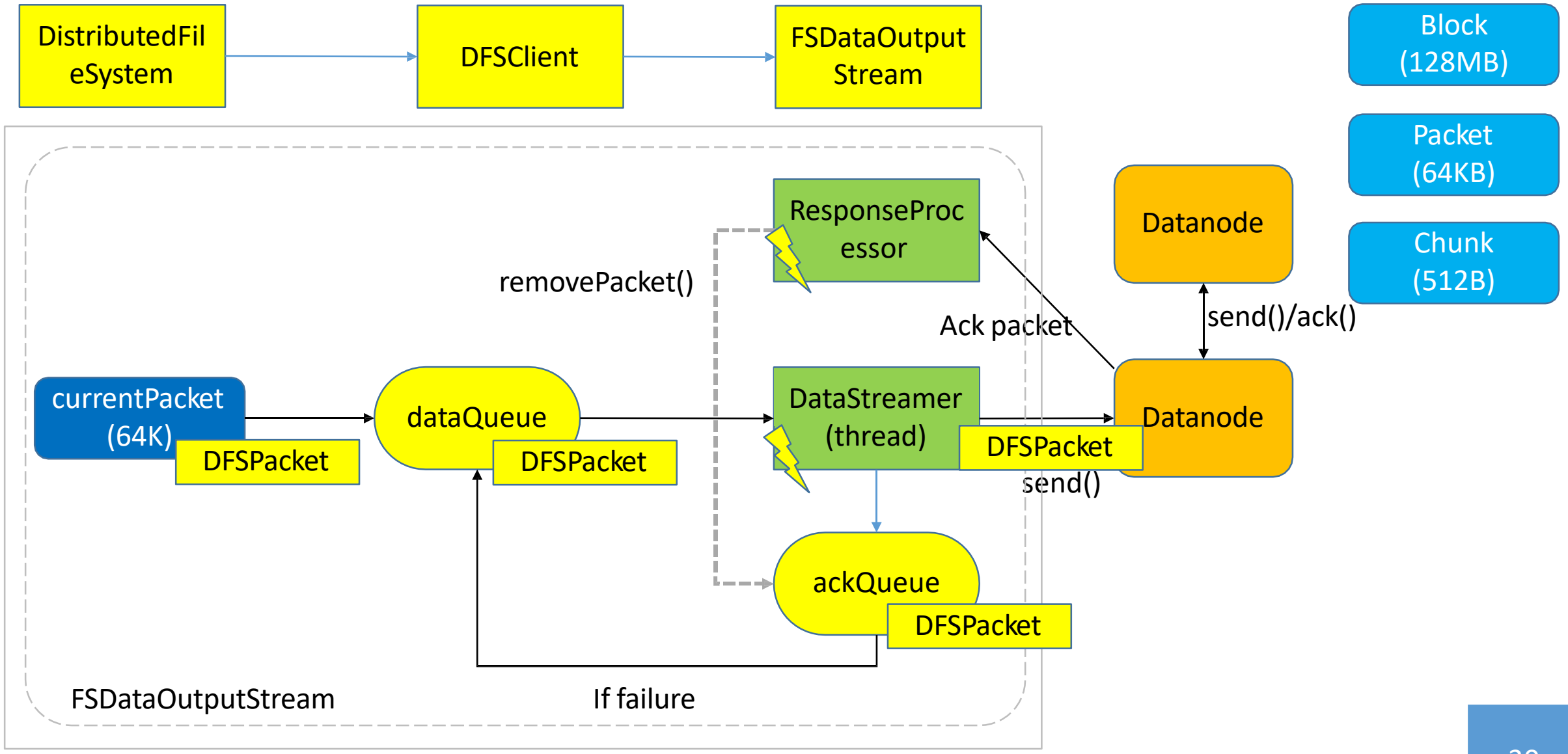
Чтение данных



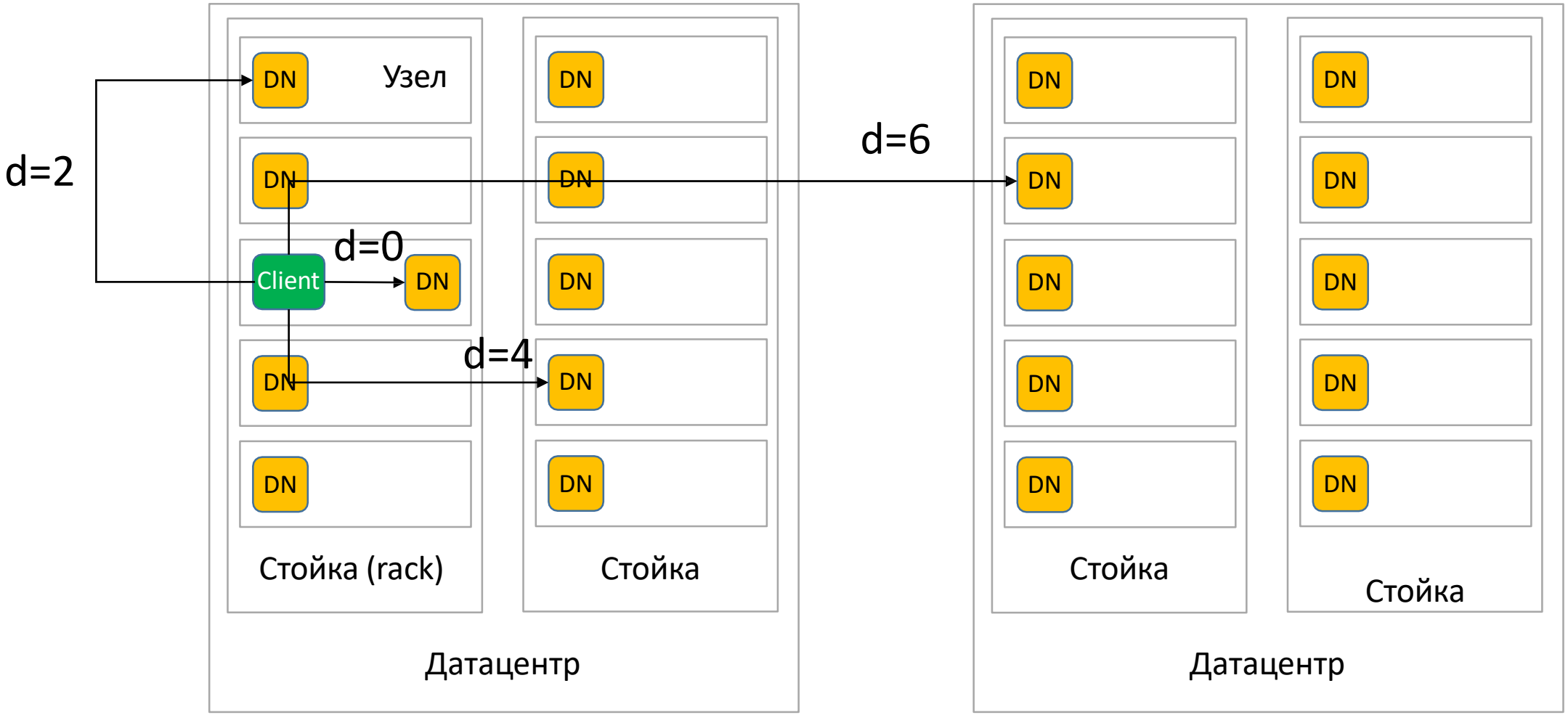
Запись данных



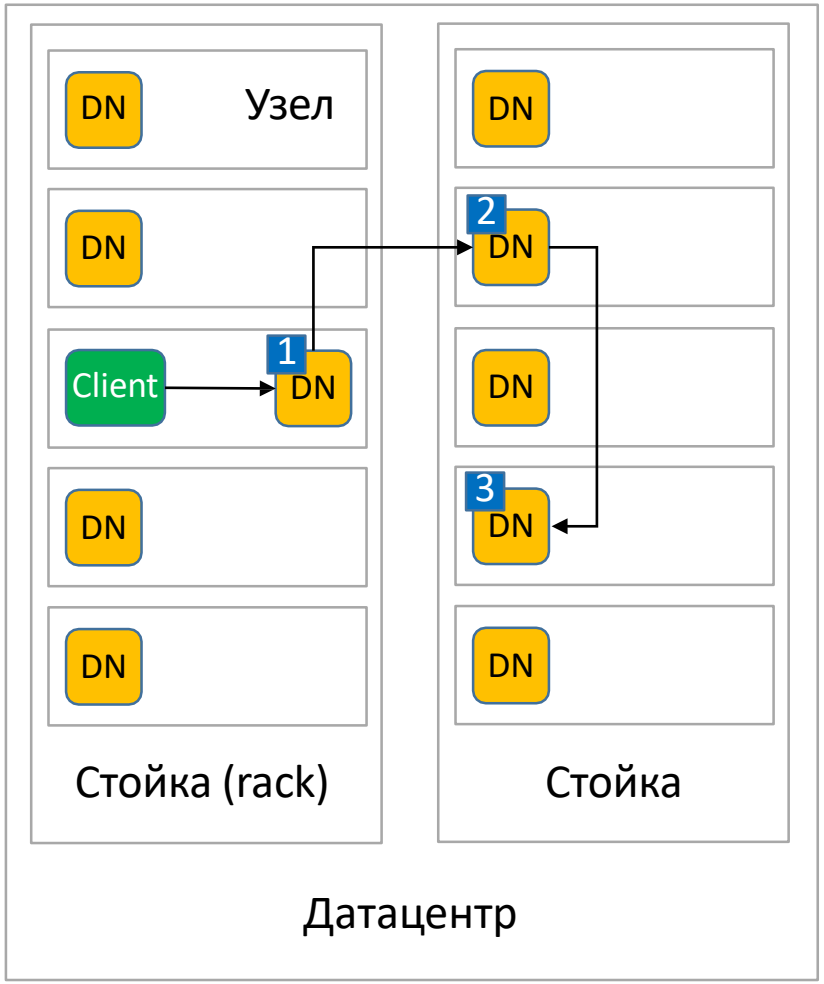
Запись данных. Write(), writePacket(), ackPacket()



Определение расстояния



Расположение реплик



Namenode vs Datanode

Что теряем, если выйдет из строя Datanode/Namenode?

Дополнительные возможности

- Secondary NameNode
- High Availability
- Federation

Secondary Namenode

Namenode хранит образ всего пространства имен файловой системы и распределение блоков по файлам в памяти

- Fsimage: checkpoint пространства имен файловой системы
- Edit logs: содержит изменения пространства имен

Checkpoint – процесс слияния старого представления fsimage в памяти с edits и запись на диск нового fsimage

Secondary Namenode загружает fsimage и edits из **Namenode**, создает новый fsimage и передает его обратно в **Namenode**

По умолчанию интервал 3600 сек., размер edits для старта 64MB

HDFS High Availability

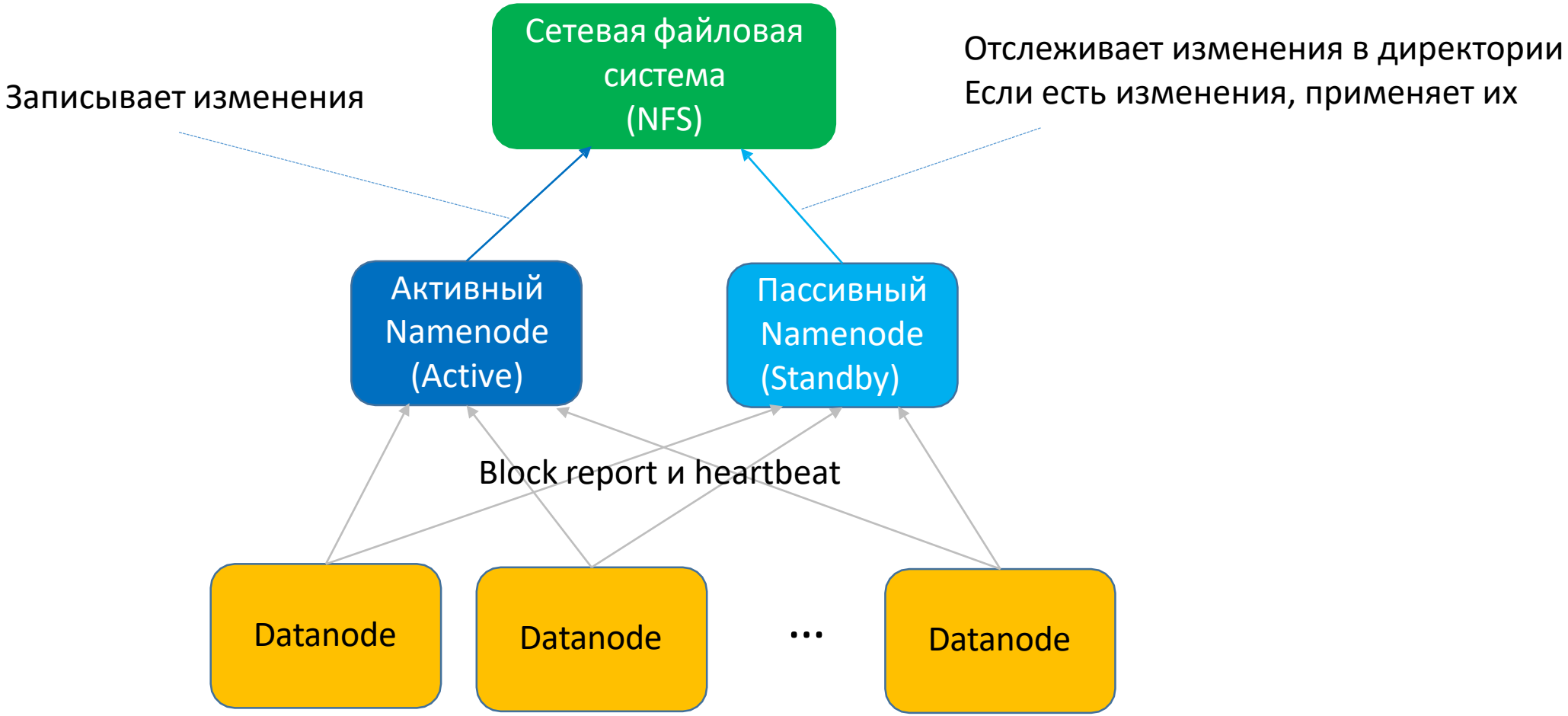
HDFS High Availability – механизм HDFS, поддерживающий работу нескольких Namenode'ов (2 и более) в конфигурации **активный/пассивные**.

- Позволяет быстро восстановить данные на **пассивном Namenode**.
- **Используется** при 1) выходе из строя активного Namenode, 2) при плановой поддержке (например, при обновлении ПО на узле, где работает активный Namenode)
- **Активный Namenode** отвечает за все операции клиента в кластере
- Datanode'ы отправляют block report и heartbeat'ы всем Namenode'ам
- Пассивные Namenode также выполняют checkpoint, поэтому нет необходимости в Secondary Namenode, CheckpointNode или BackupNode

HDFS High Availability. Виды

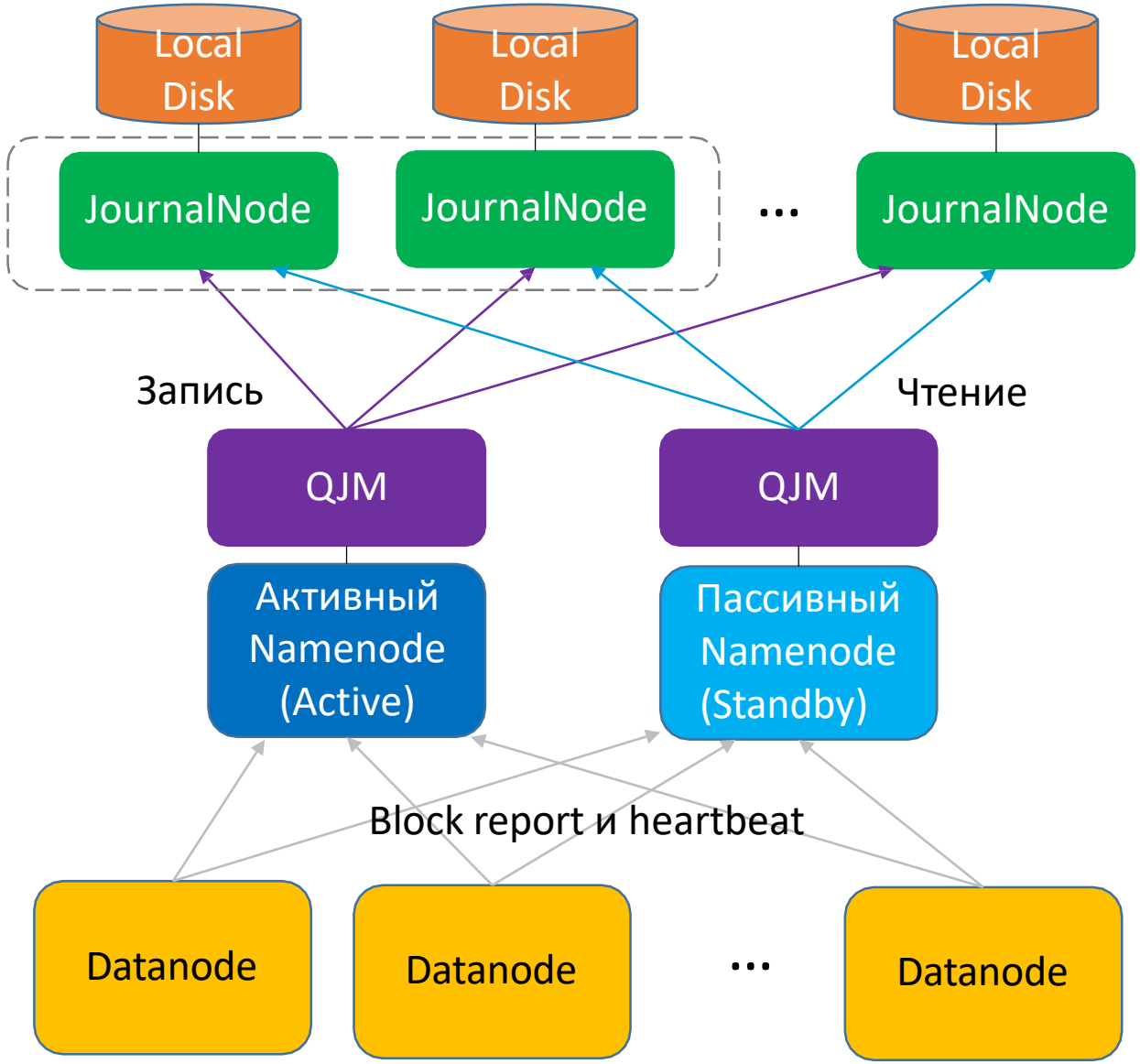
- Общая сетевая файловая система (NFS)
- Quorum Journal Manager (QJM)

HDFS HA. NFS. Синхронизация Active NN и Standby NN



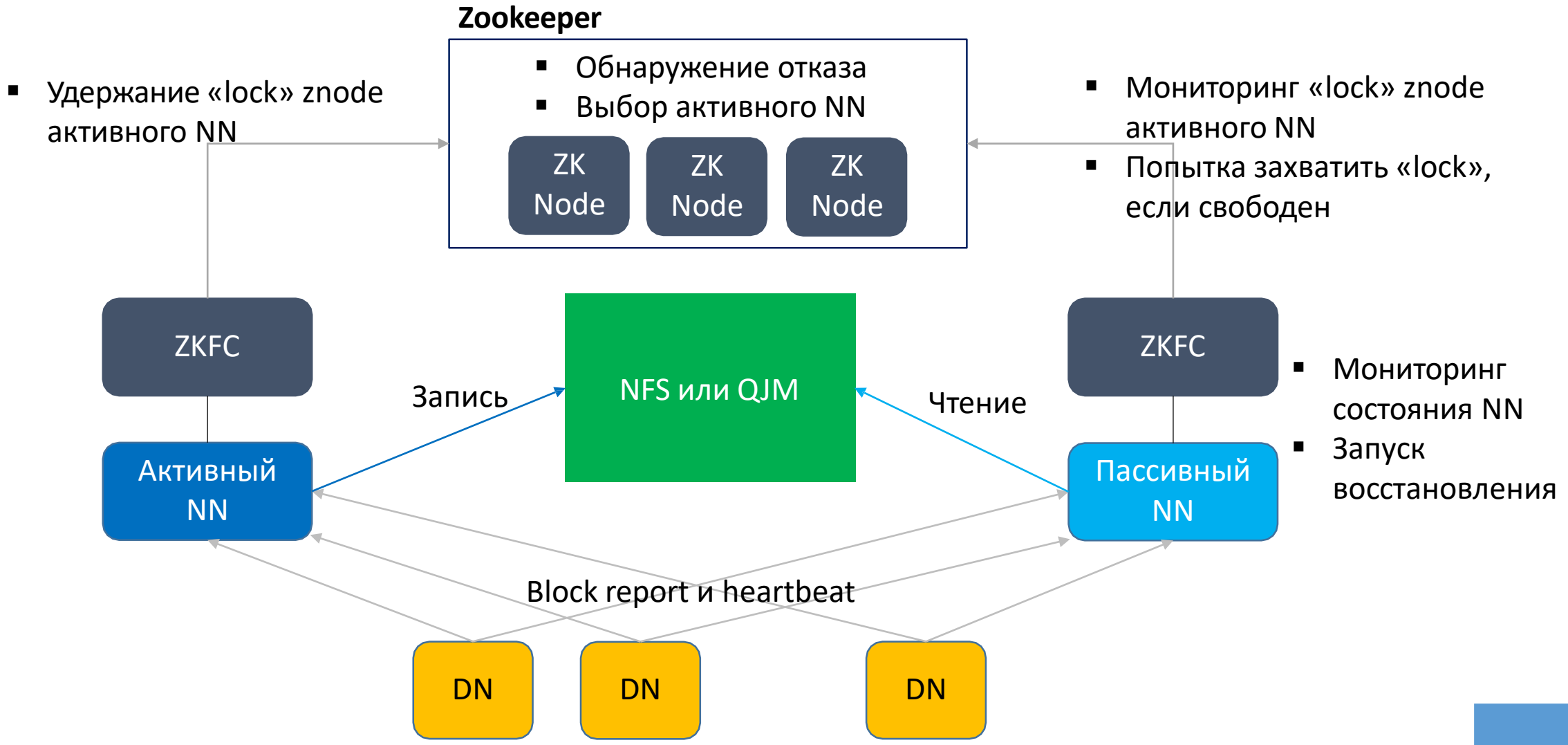
HDFS HA. QJM. Синхронизация Active NN и Standby NN

Кворум – большинство (majority) JN



HDFS HA. Автоматический запуск восстановления

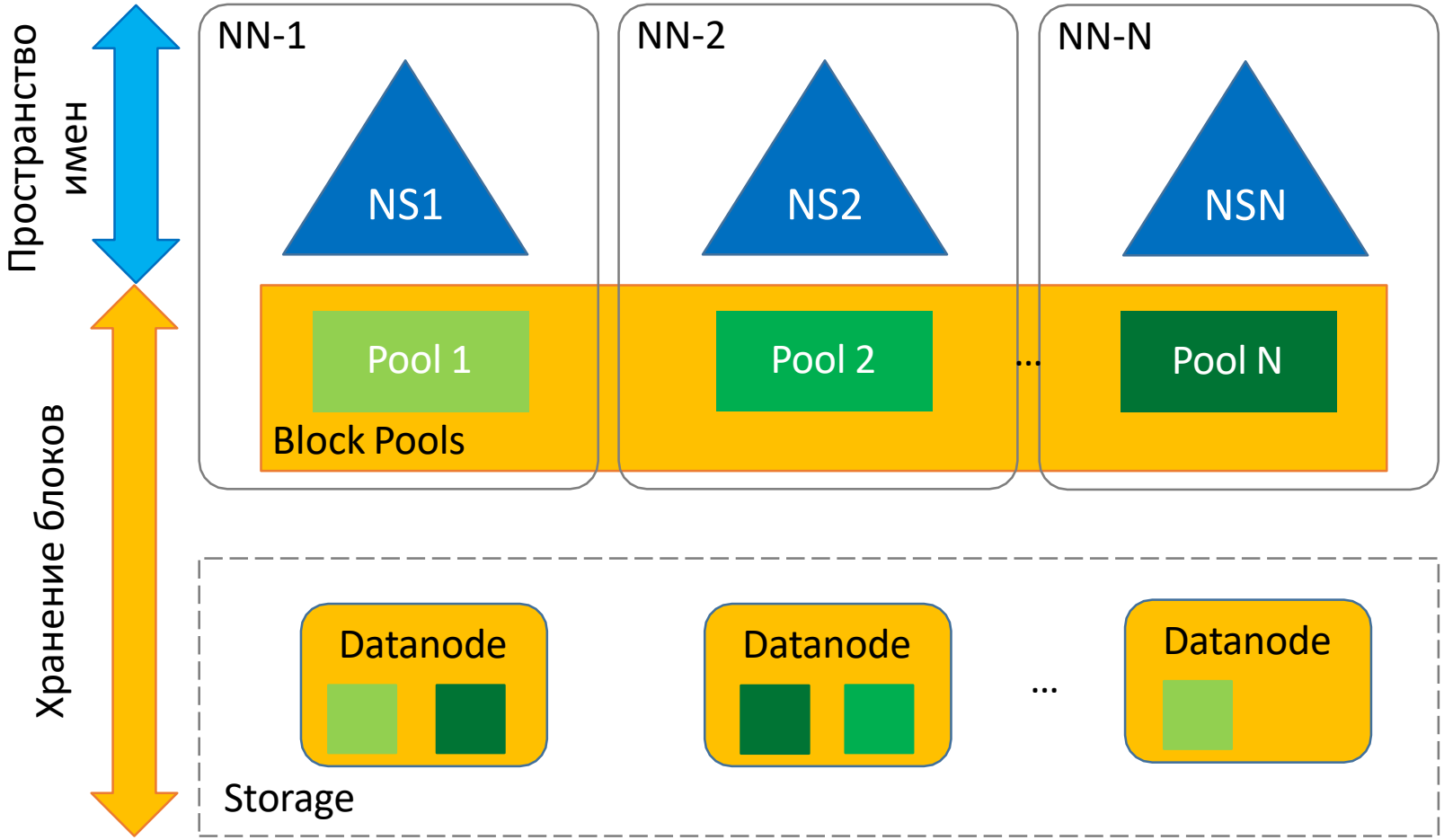
По умолчанию механизм восстановления запускается вручную



Federation

Block Pool – множество блоков, которые принадлежат одному пространству имен

Пространство имен и его **block pool** вместе называются **Namespace Volume**



Преимущества Federation

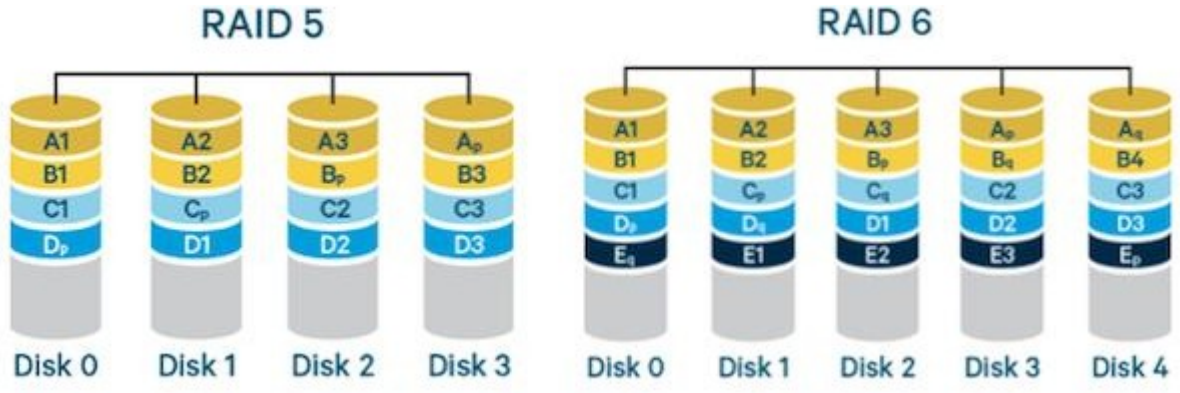
- Масштабируемость пространства имен
- Производительность (увеличивается пропускная способность)
- Изолированность

Hadoop 3.x - HDFS

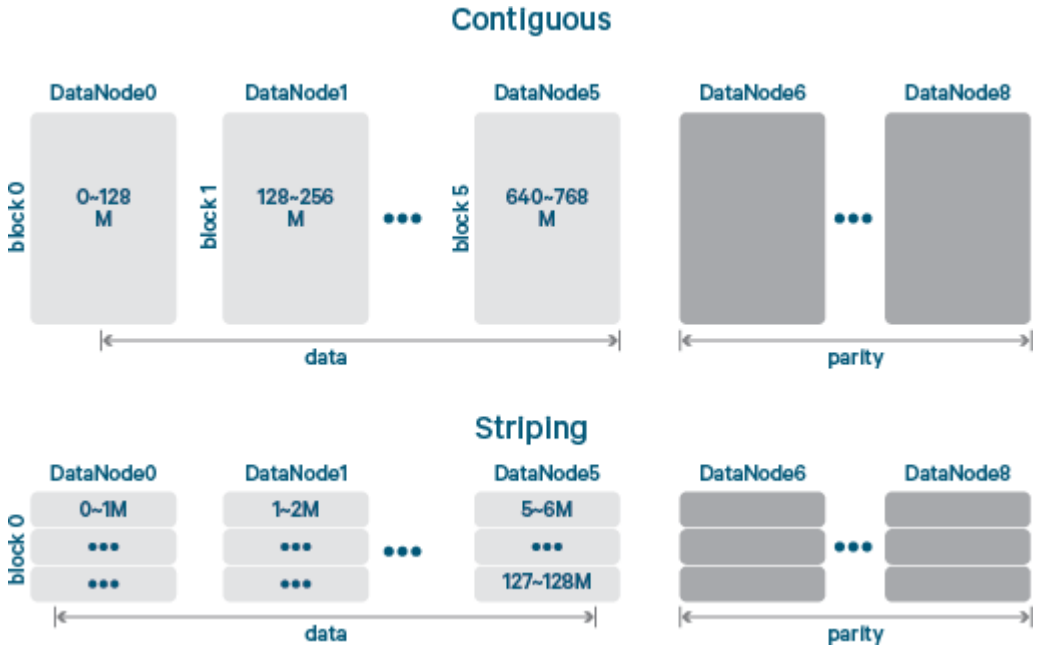
Изменения

- Intra-DataNode Balancer
- Erasure Encoding

Erasure Encoding



	Data Durability	Storage Efficiency
Single replica	0	100%
Three-way replication	2	33%
XOR with six data cells	1	86%
RS(6,3)	3	67%
RS(10,4)	4	71%



typically 64KB or 1MB

Источники

Hadoop (github source code)

Hadoop: The Definitive Guide, 4th Edition (book)

The Hadoop Distributed File System (book)

Check point (e-book)

Configuring Environment of Hadoop Daemons (doc)

HDFS High Availability Using the Quorum Journal Manager (doc)

HDFS Federation (doc)

Hadoop HDFS High Availability (blog)

A Guide to Checkpointing in Hadoop (blog)

Quorum-based Journaling in CDH4.1 (blog)

Introduction to HDFS Erasure Coding in Apache Hadoop (blog)