

СИСТЕМЫ ОБРАБОТКИ БОЛЬШИХ ДАННЫХ

Разработка приложений



К.Т.Н.
Папулин Сергей Юрьевич
papulin_bmstu@mail.ru

Лекция 2. Распределенная файловая система HDFS



- Виртуальная Java Машина (JVM)
- Назначение и отличительные особенности HDFS
- Основные компоненты HDFS
- Операции чтение/запись в HDFS
- HDFS HA
- Hadoop 3.x – HDFS

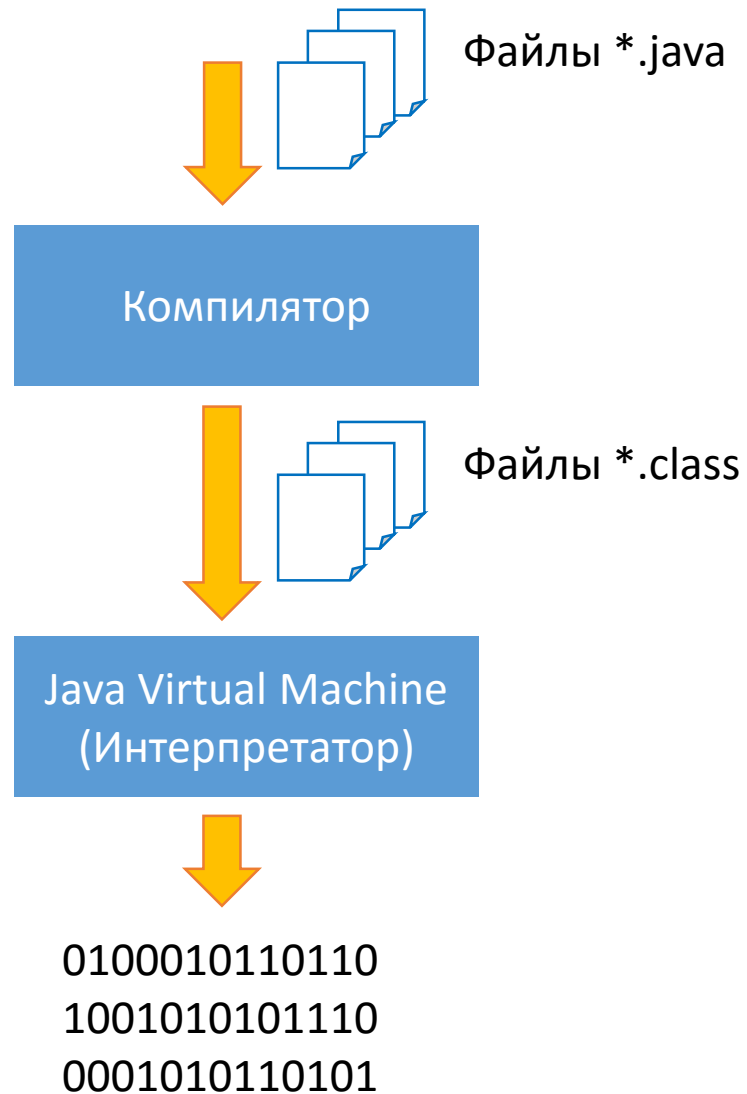
Виртуальная Java Машина (JVM)

Java Virtual Machine (JVM) – спецификация для реализации выполняемой в реальном времени среды, в которой Java байт-код транслируется в машинный код

Реализация JVM – Java Runtime Environment (JRE)

Java Development Kit = JRE + дополнительные пакеты/инструменты для разработки

Этапы трансляции программного кода в машинный



Компиляция исходного кода программы

```
void spin() {  
    int i;  
    for (i = 0; i < 100; i++) {  
        ; // Loop body is empty  
    }  
}
```

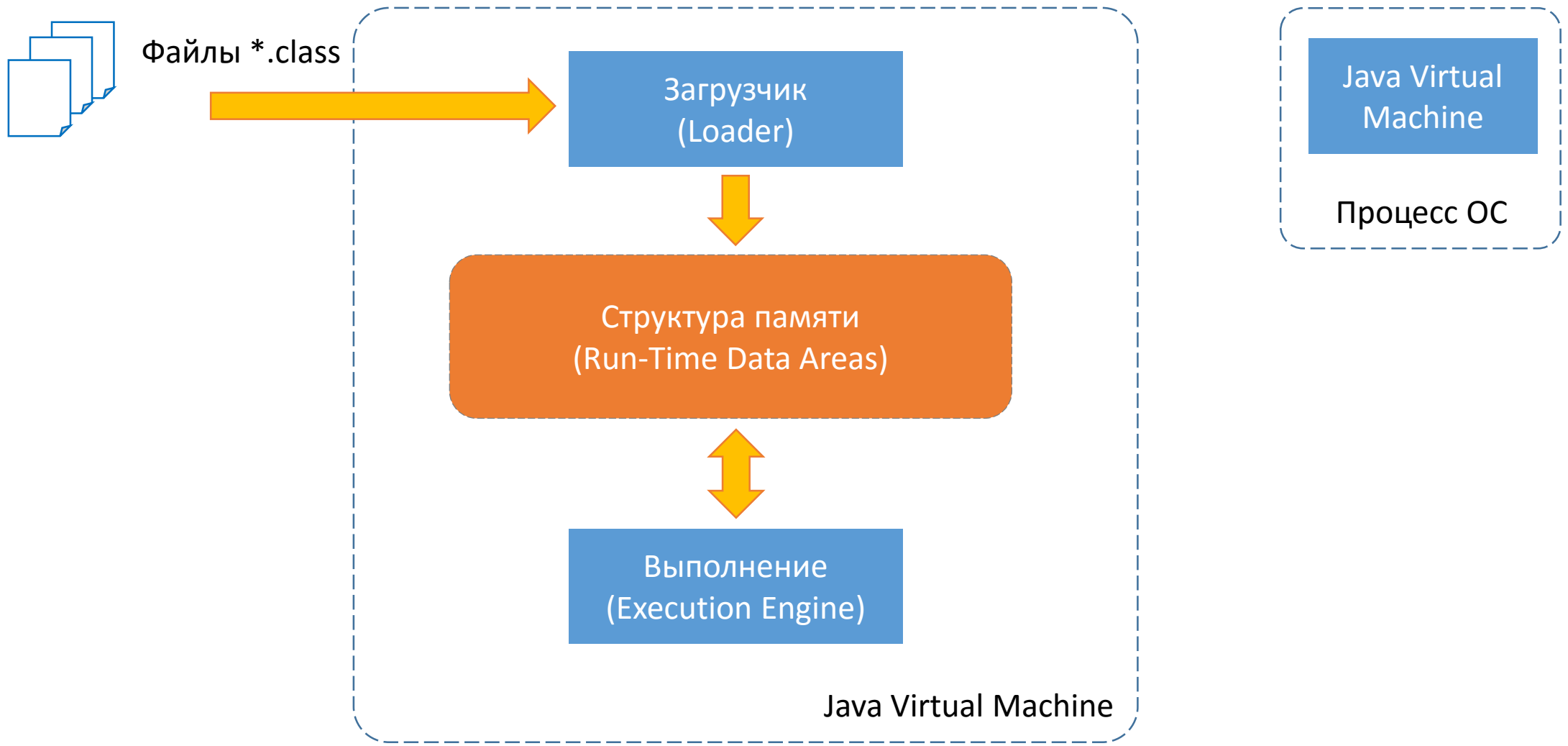


Компилятор

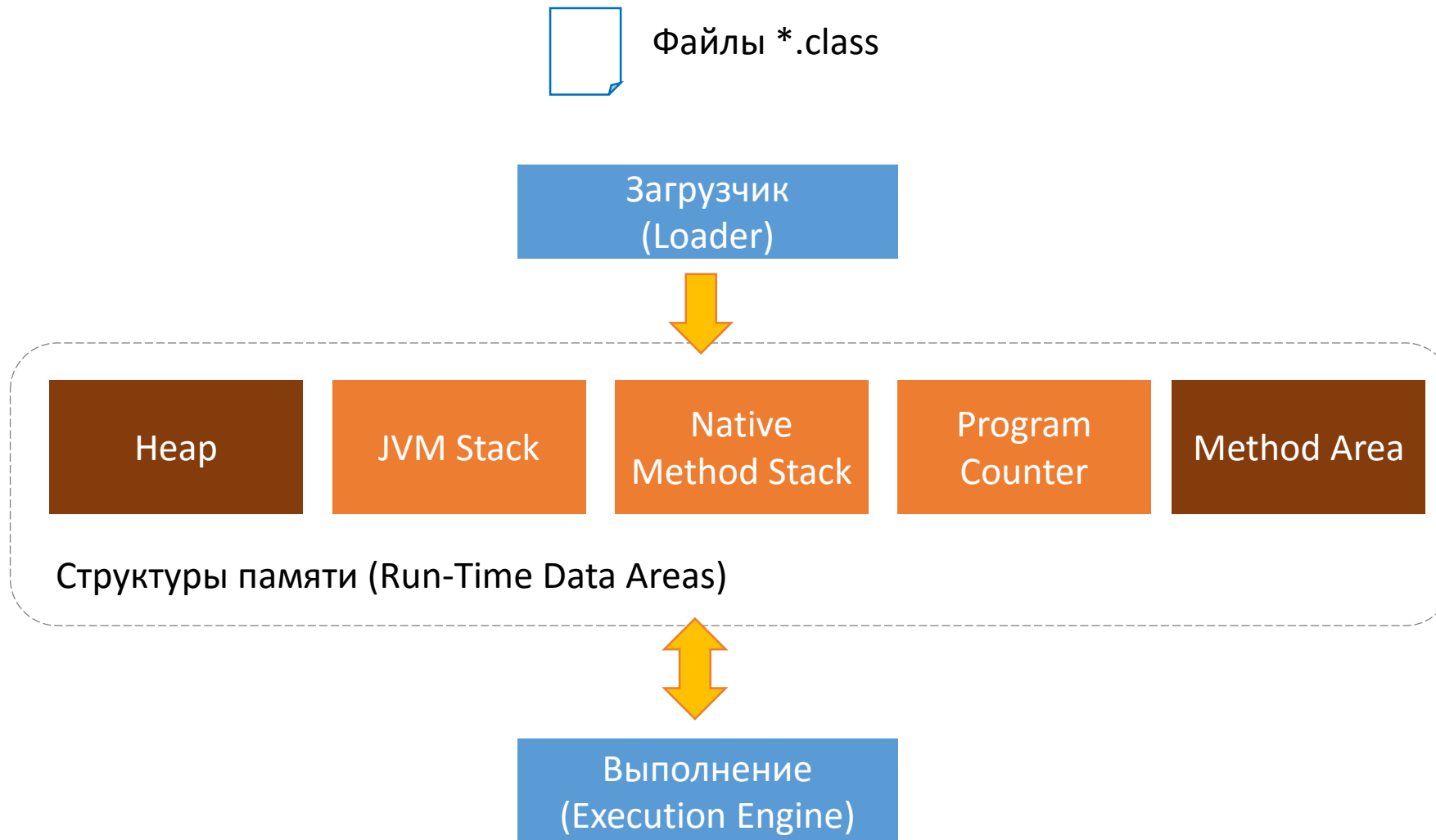


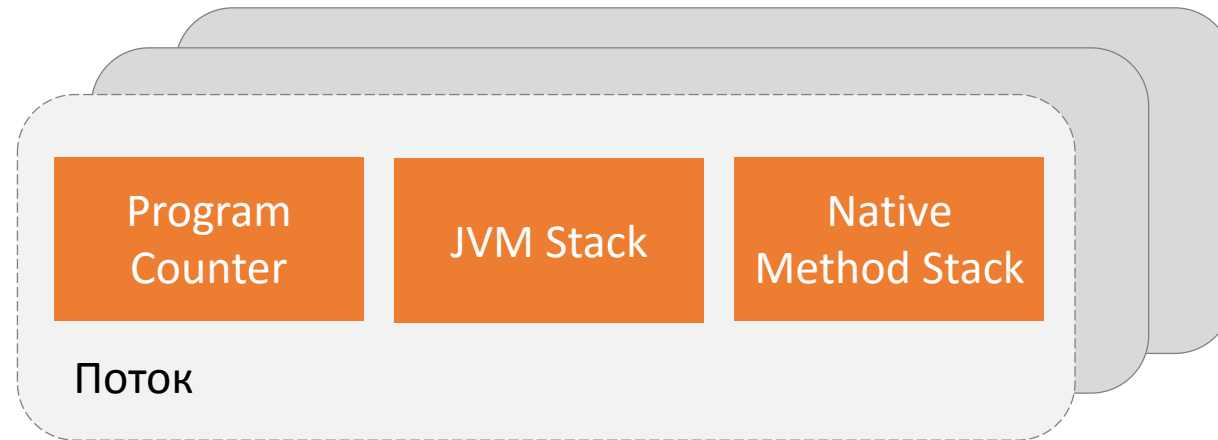
```
0  iconst_0    // Push int constant 0  
1  istore_1    // Store into local variable 1 (i=0)  
2  goto 8      // First time through don't increment  
5  iinc 1 1     // Increment local variable 1 by 1 (i++)  
8  iload_1     // Push local variable 1 (i)  
9  bipush 100   // Push int constant 100  
11 if_icmplt 5  // Compare and loop if less than (i < 100)  
14 return      // Return void when done
```

Структура кучи (heap)

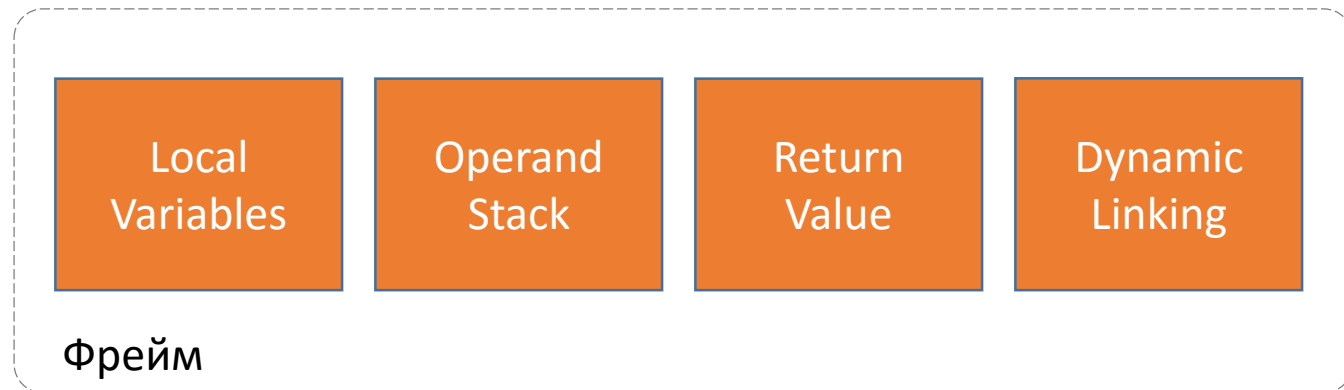
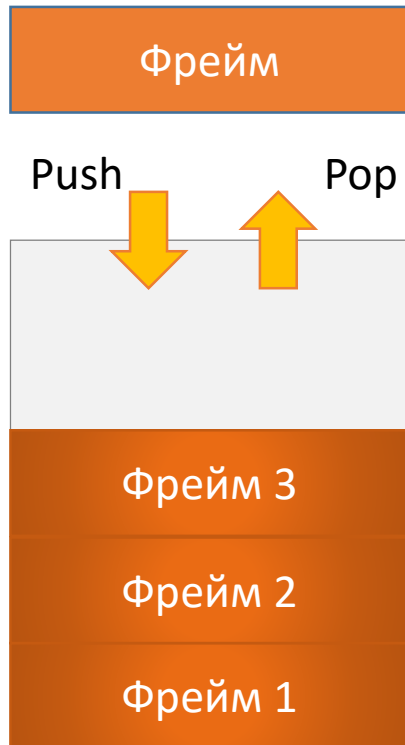


Структуры памяти в JVM

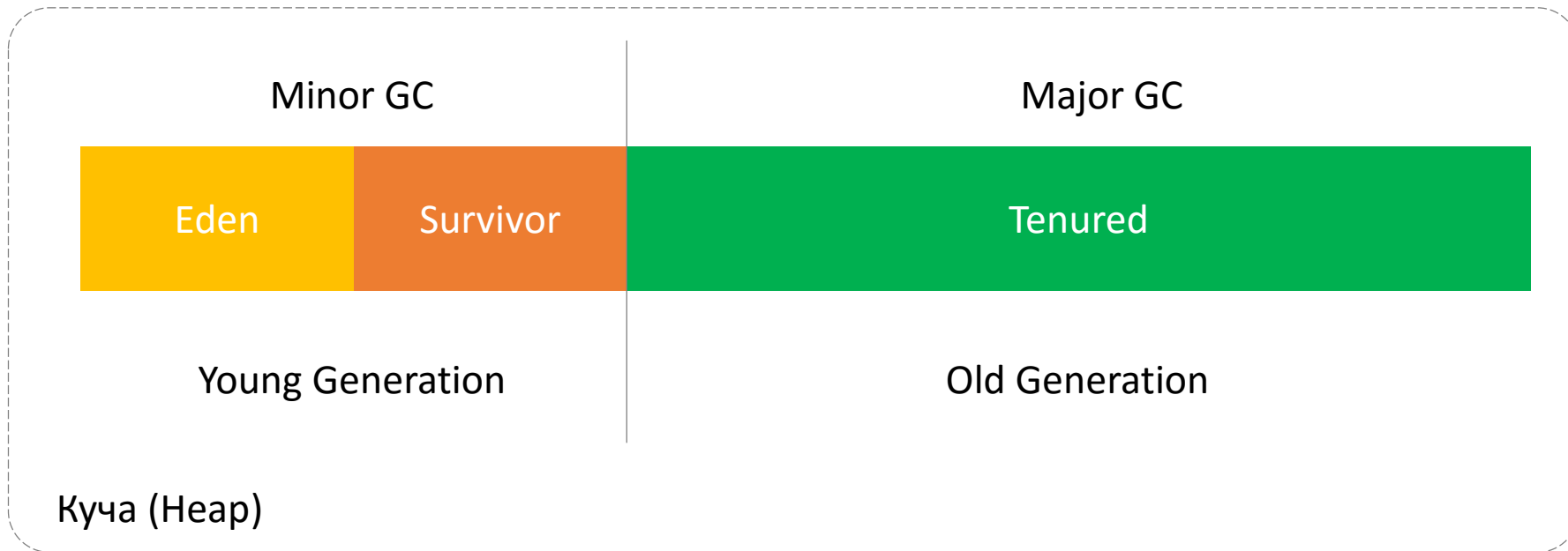




Структура кучи (heap)



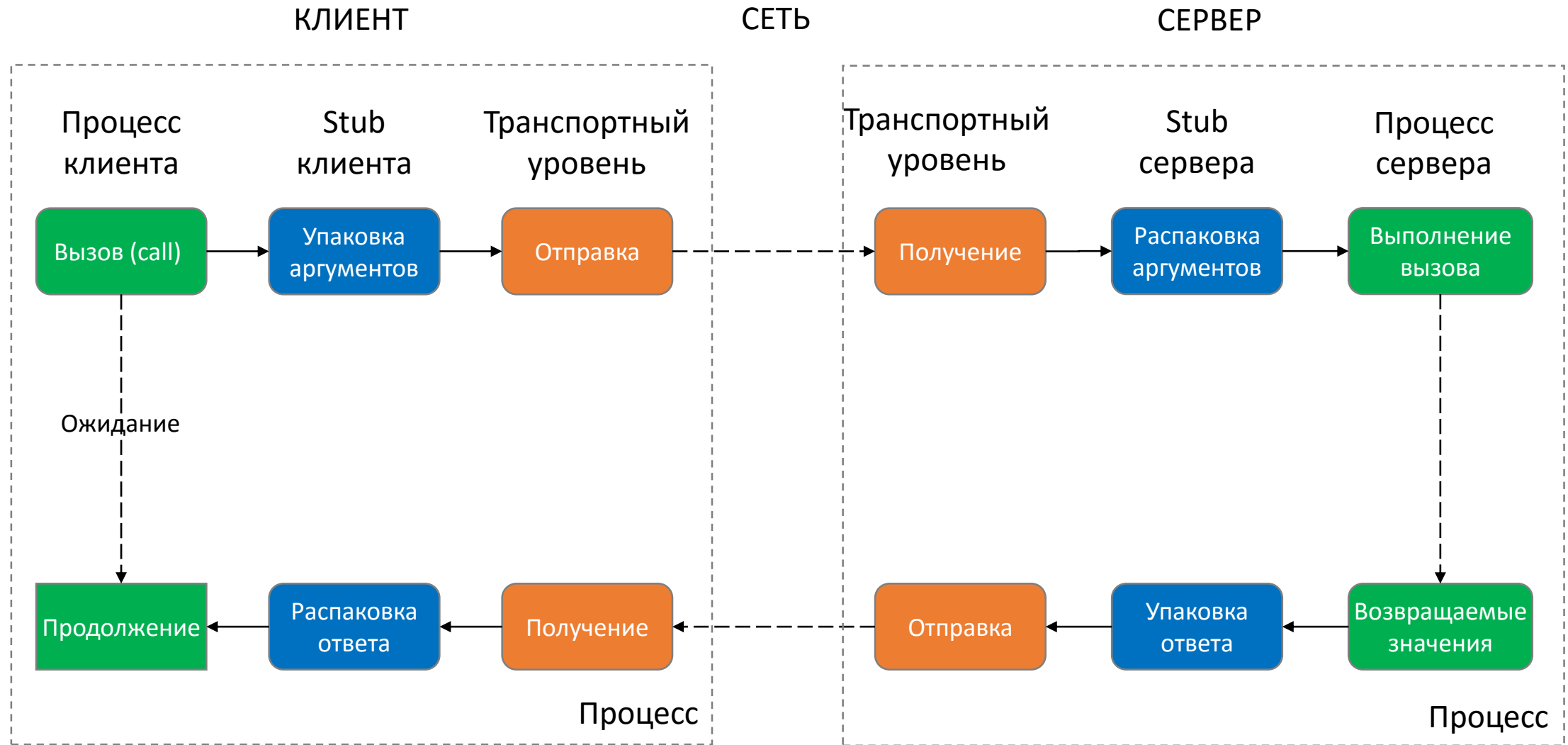
Поколения и куча



- Объем heap и non-heap памяти
- Количество потоков
- Количество классов, загружаемых в память
- Параметры GC

Удаленный вызов процедур RPC

Remote Procedure Call (RPC)



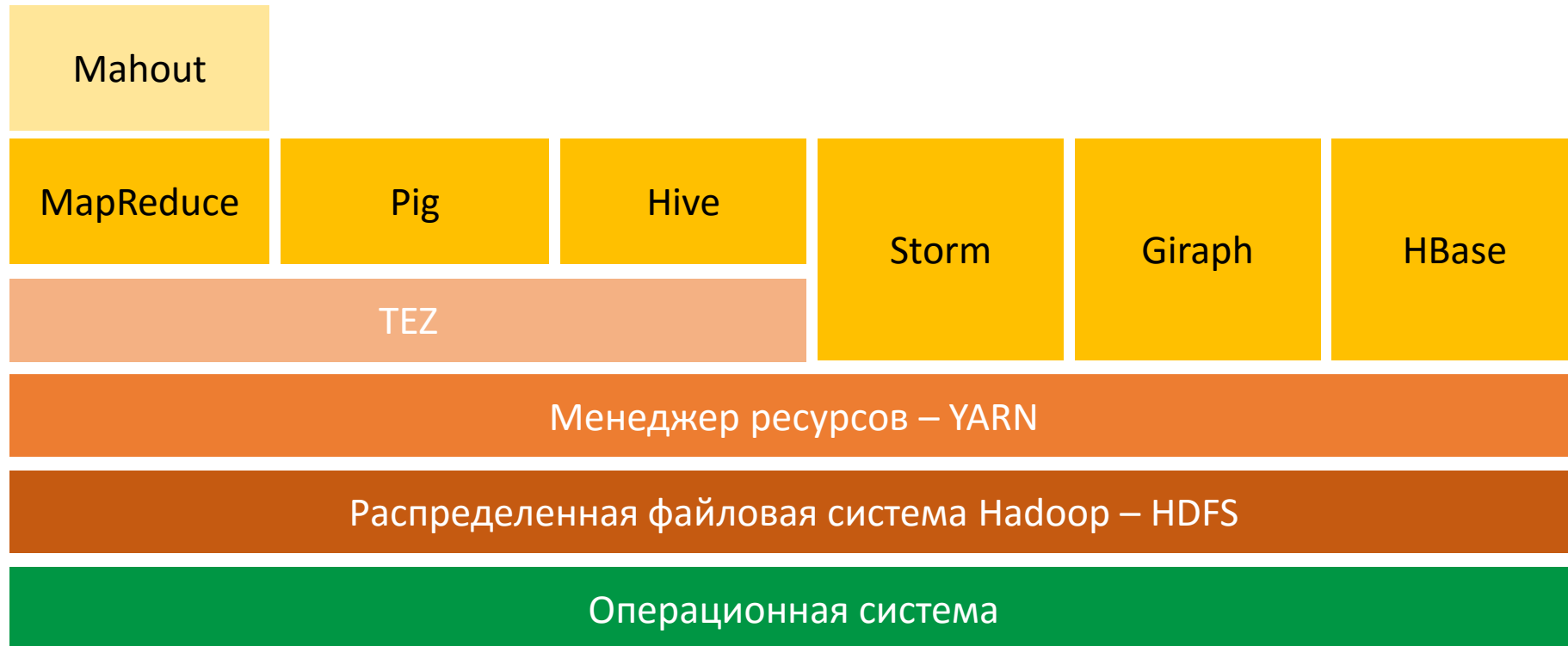
Назначение и отличительные особенности Hadoop

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

Сравнение с Hadoop

	Реляционная СУБД	Hadoop
Размещение данных	Централизованное	Распределены по вычислительным узлам
Размер данных	Гигабайты	Петабайты
Доступ	Интерактивный и batch	Batch
Изменения	Чтение и множественная запись	Запись один раз, чтение много раз
Транзакции	ACID	нет
Структура	Схема на запись	Схема на чтение
Целостность	Высокая	Низкая
Масштабируемость	Нелинейная	Линейная

Стек Hadoop



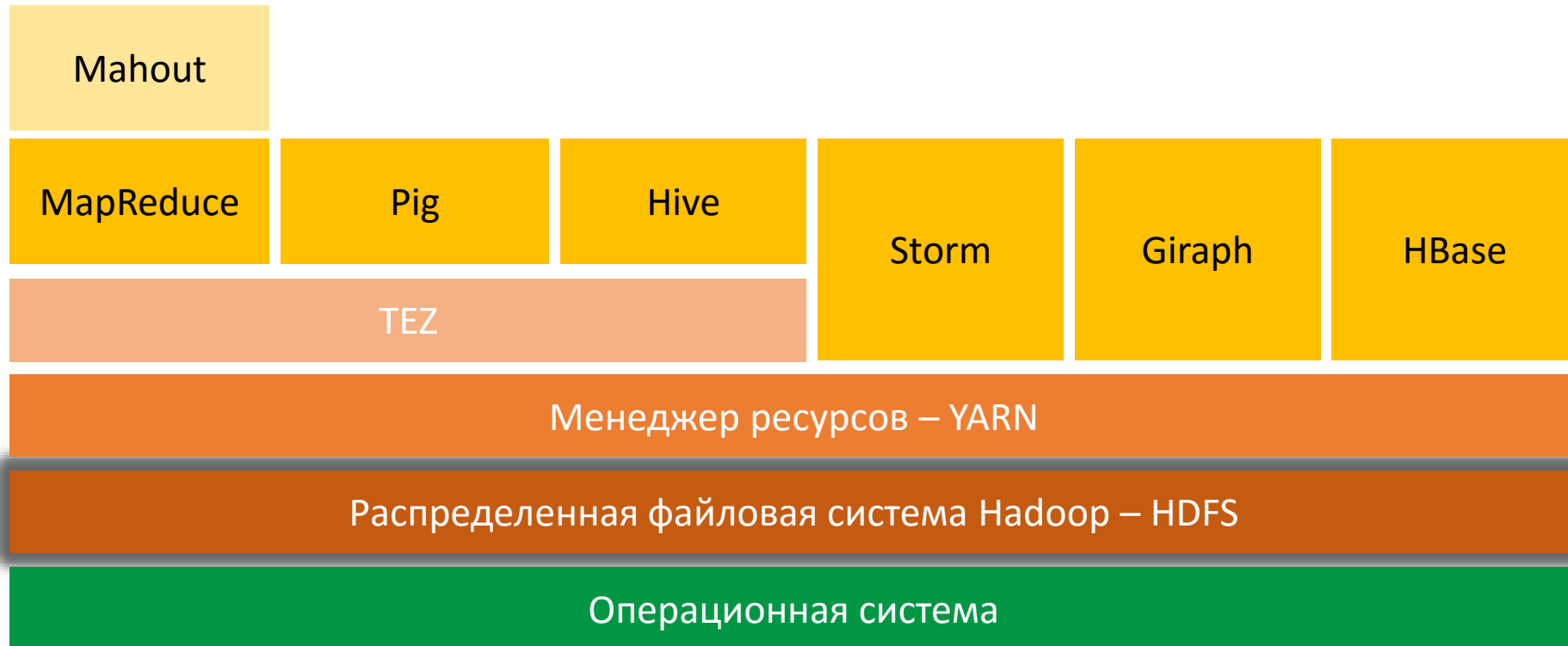
Распределенная файловая система Hadoop (HDFS)

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)

- + Большие файлы
- + Поточковый доступ к данным
- + Серверы общего назначения

- Низкая задержка доступа к данным
- Множество маленьких файлов
- Произвольный доступ и изменение данных

Стек Hadoop



Основные компоненты HDFS

➤ Данные

➤ Блок данных

➤ Клиент

➤ NameNode

➤ Secondary NameNode

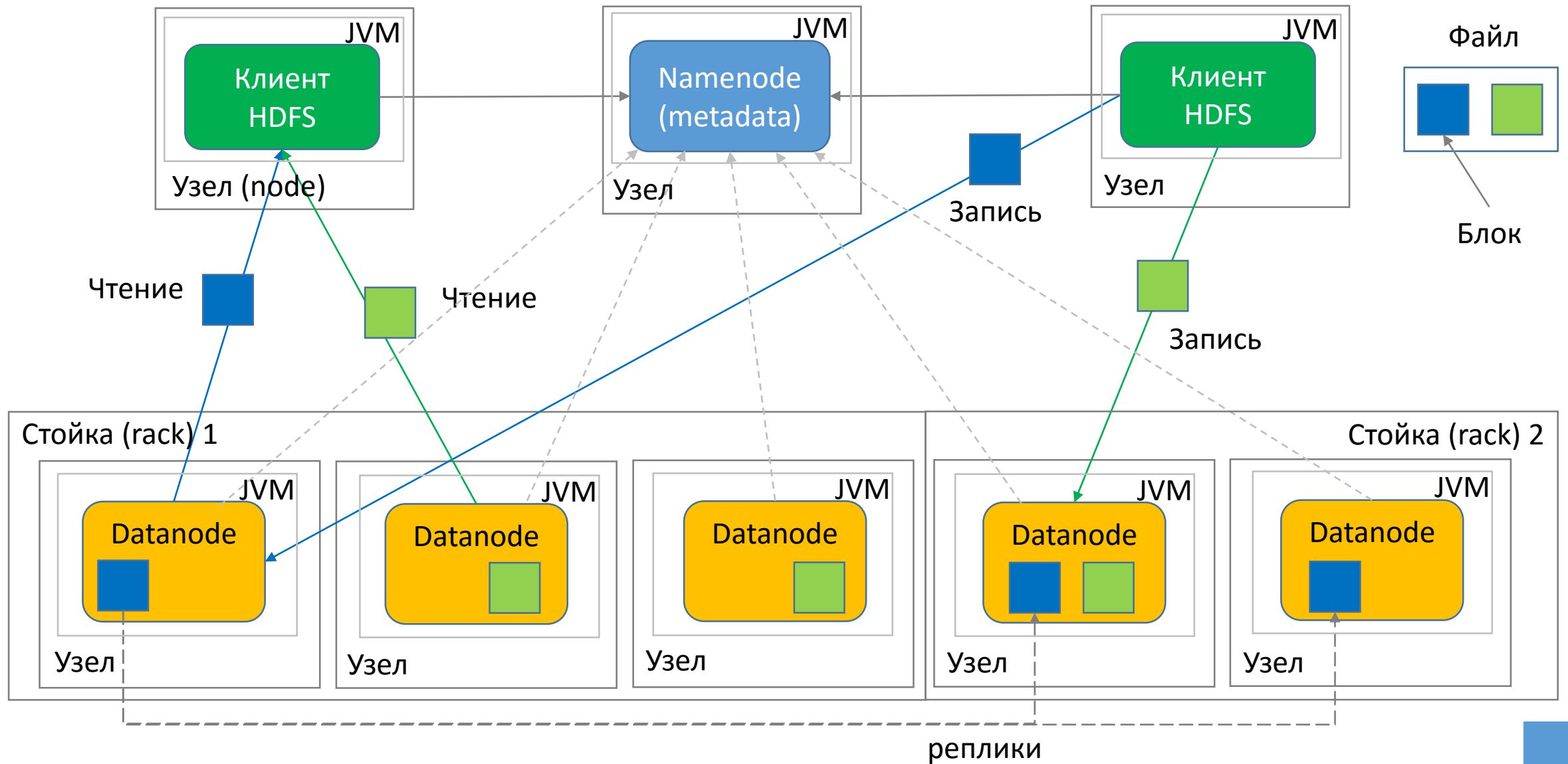
➤ DataNode

➤ Реплика

➤ JournalNode

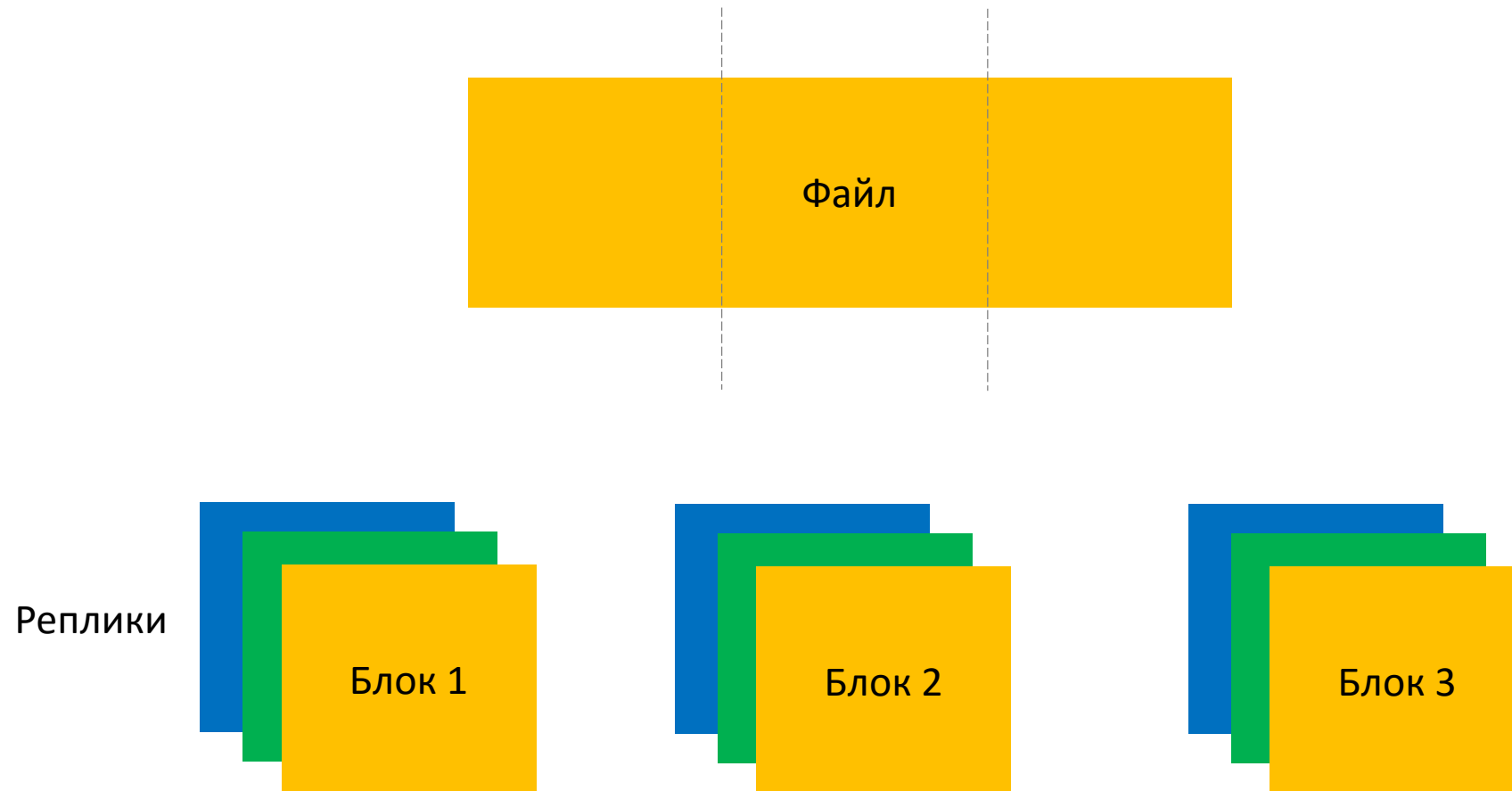
➤ Federation

Архитектура HDFS



Блоки данных – объем данных для чтения/записи

64/128/256МБ



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 – 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 задач приложений параллельно



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

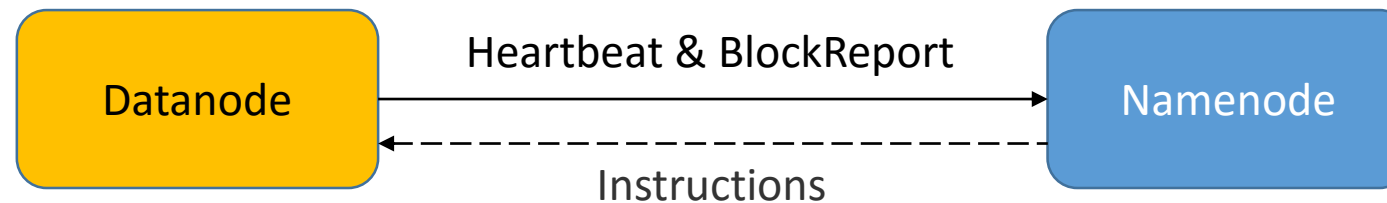
- **Block report** от DN к NN (~ часы – 1) содержит информацию о хранящихся репликах блоков данных:
 - Id блока,
 - Метка поколения
 - Размер каждого блока

NN может обрабатывать тысячи heartbeat'ов в секунду без воздействия на другие операции NN

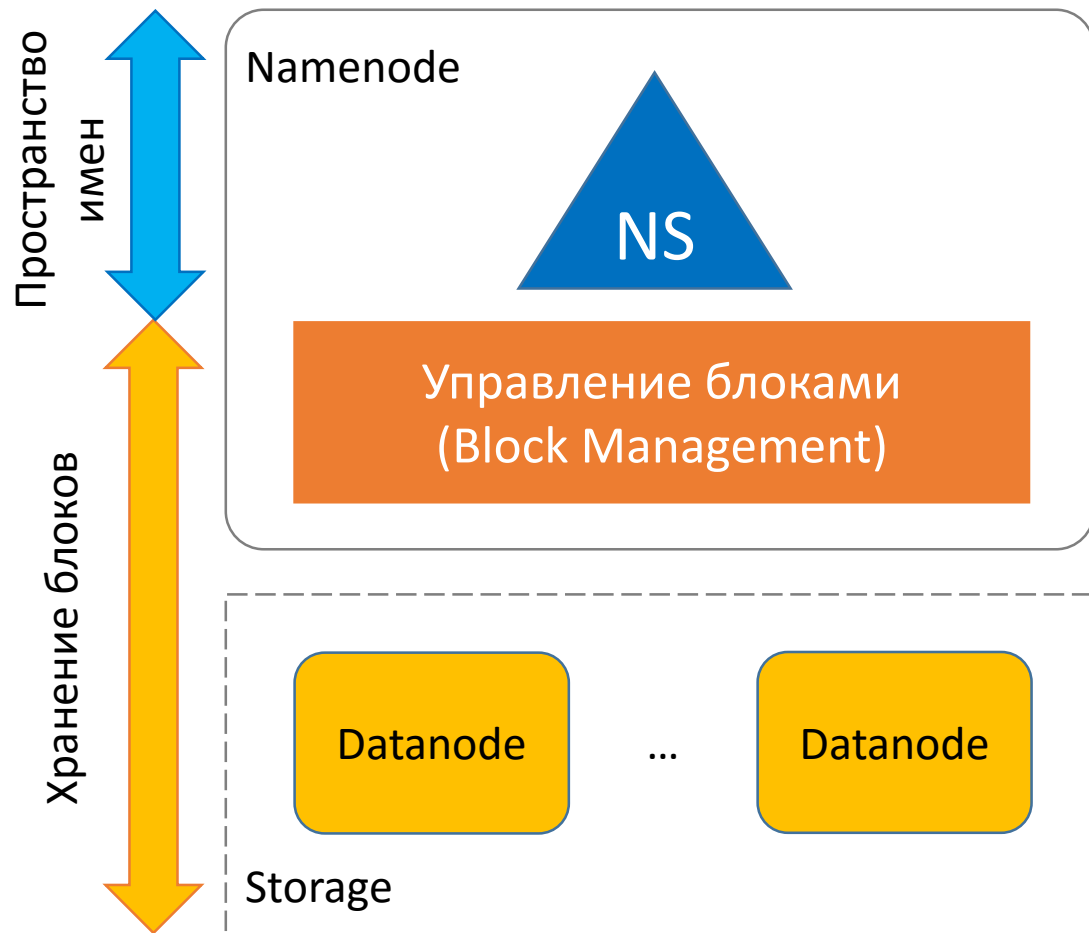
NameNode использует ответ на heartbeat'ы, чтобы отправлять **инструкции** на Datanode'ы.

Инструкции включают **команды** на:

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



Block report и heartbeat



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

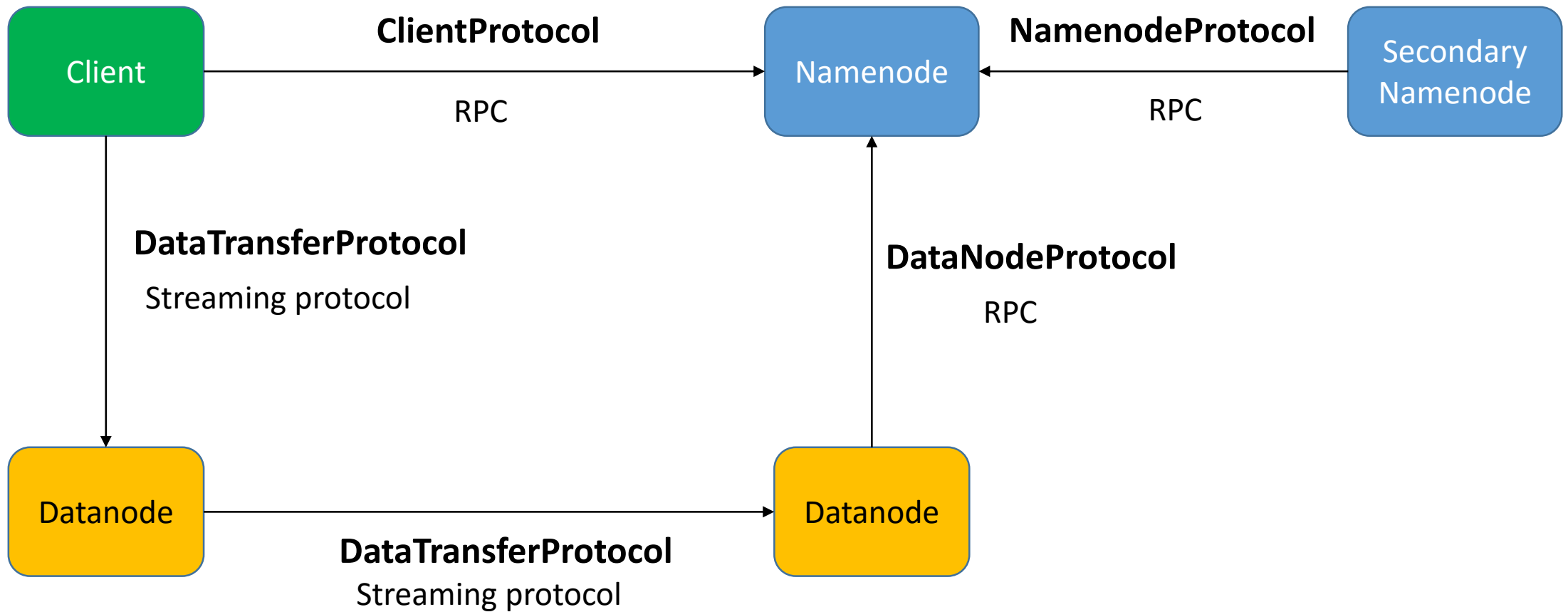
Daemon	Environment Variable
NameNode	HADOOP_NAMENODE_OPTS
DataNode	HADOOP_DATANODE_OPTS
Secondary NameNode	HADOOP_SECONDARYNAMENODE_OPTS

Client

Namenode

Secondary
Namenode

Datanode



Основные команды в HDFS

“hadoop fs” vs “hdfs dfs”

hadoop fs

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

```
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 ...]
```

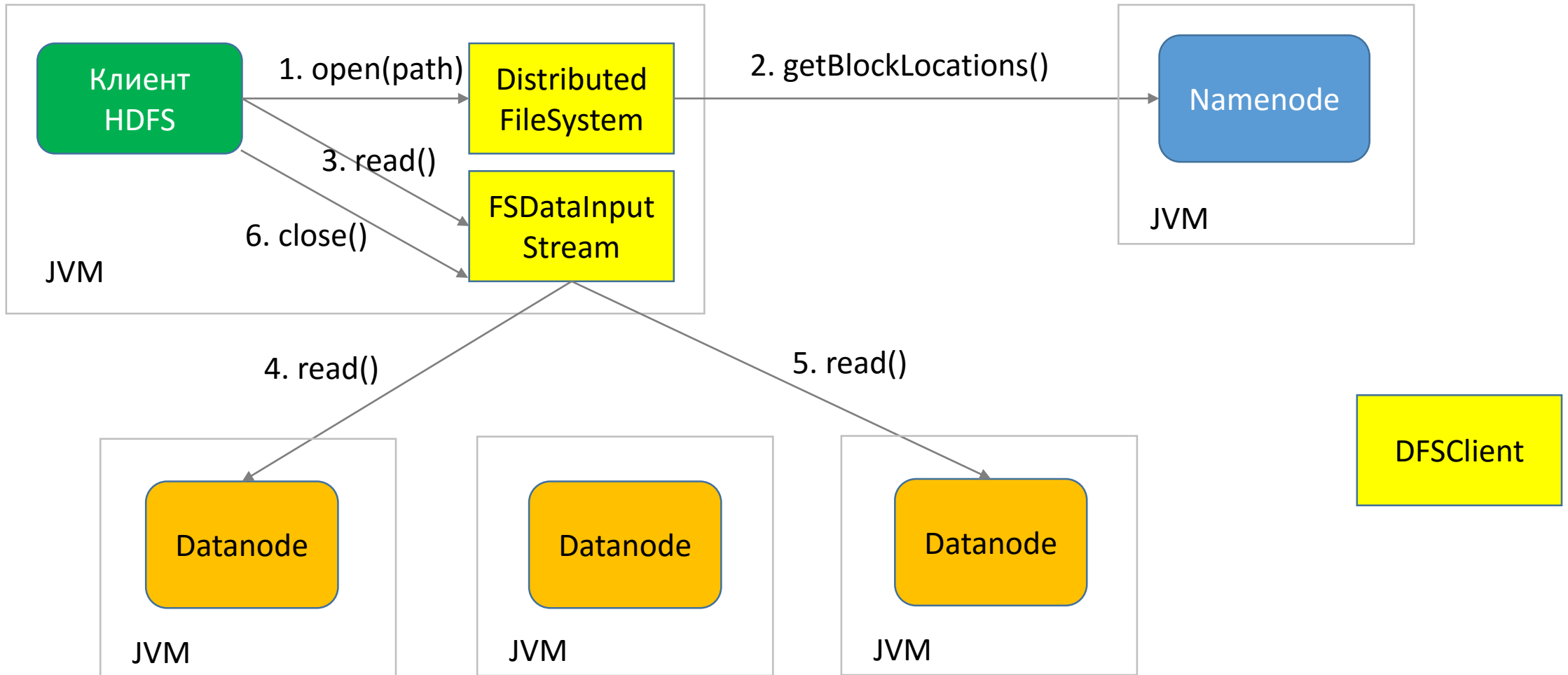
```
hadoop fs -du [-s] [-h] URI [URI ...]
```

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

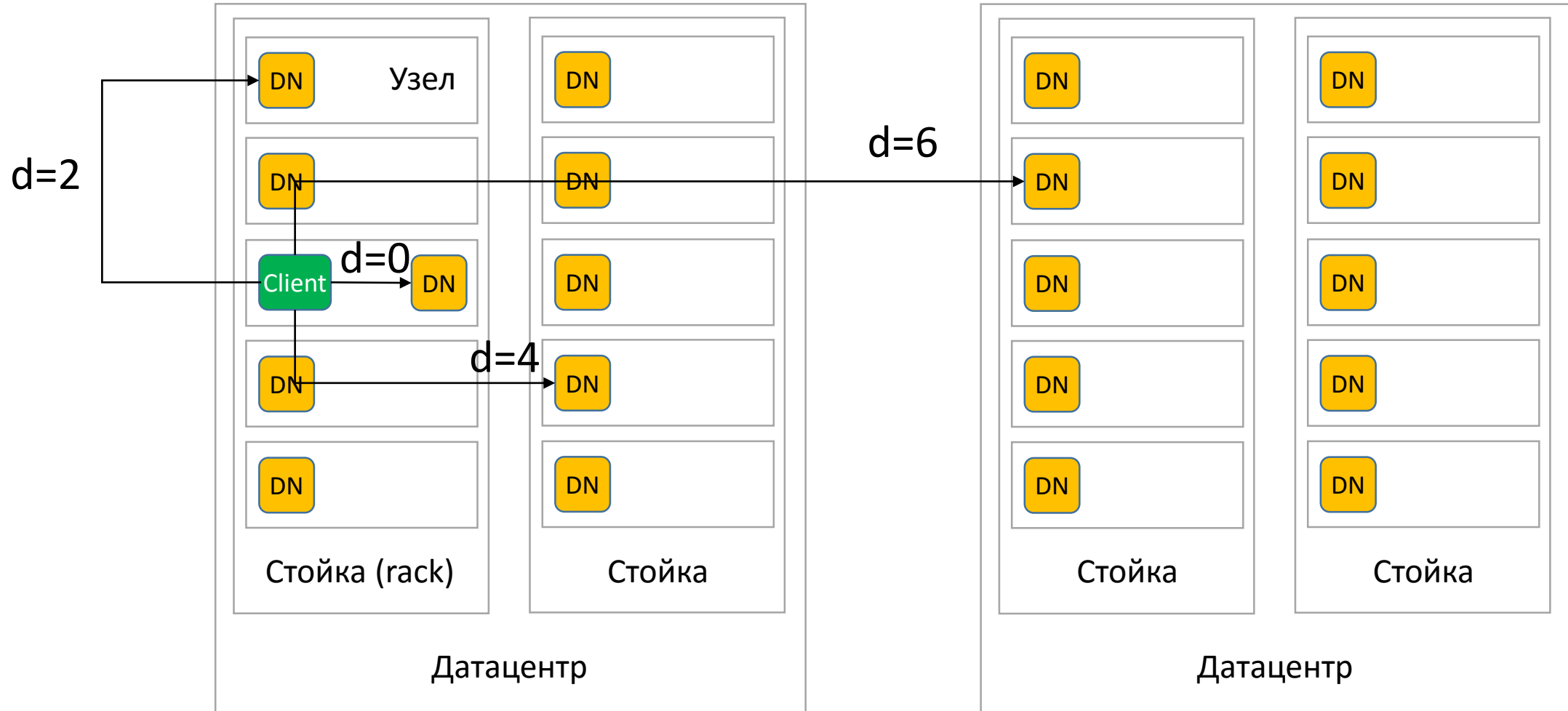
<https://hadoop.apache.org/docs/r2.7.0/hadoop-project-dist/hadoop-hdfs/HDFSCommands.html>

Операции чтение/запись в HDFS

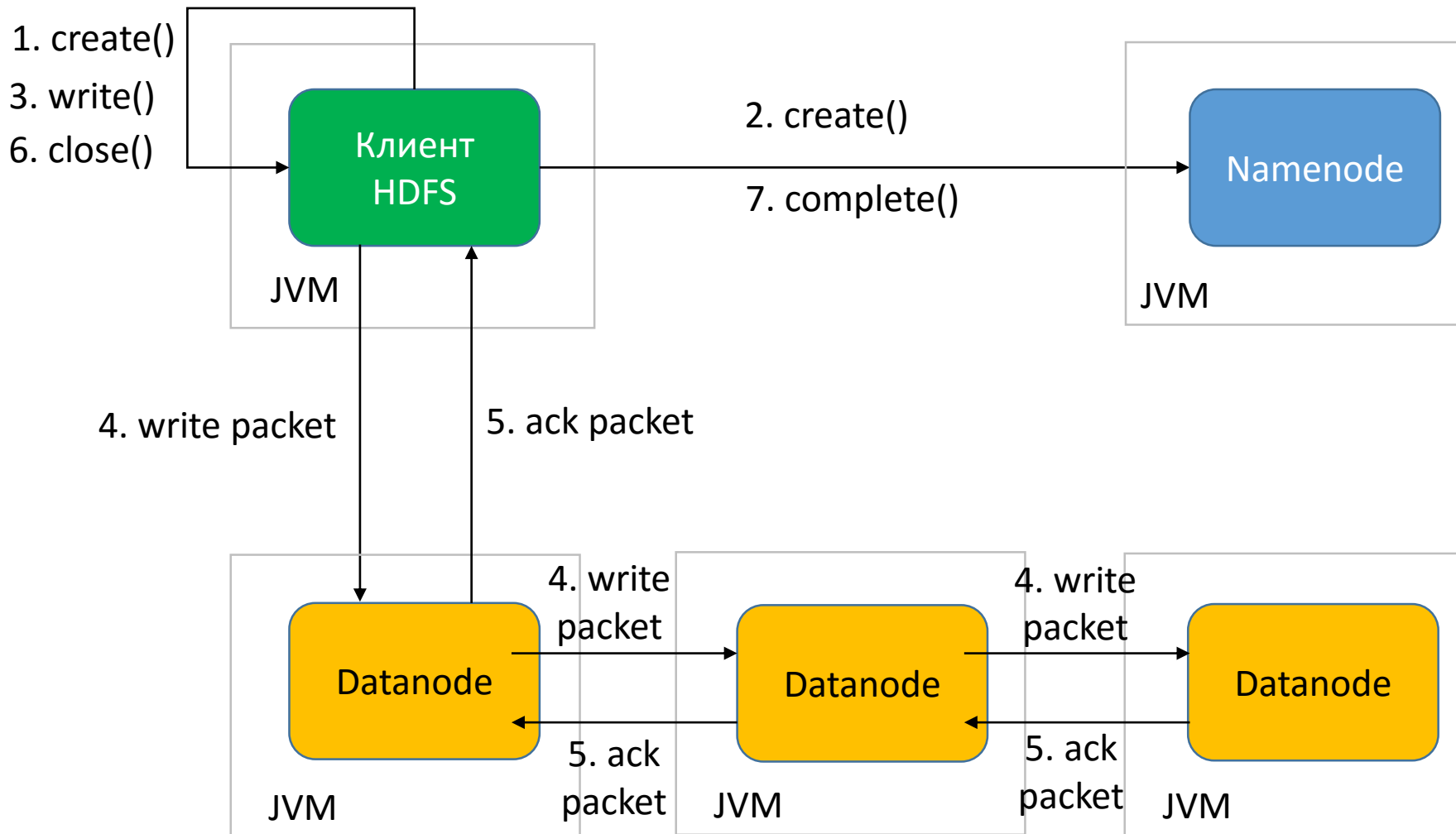
Чтение данных



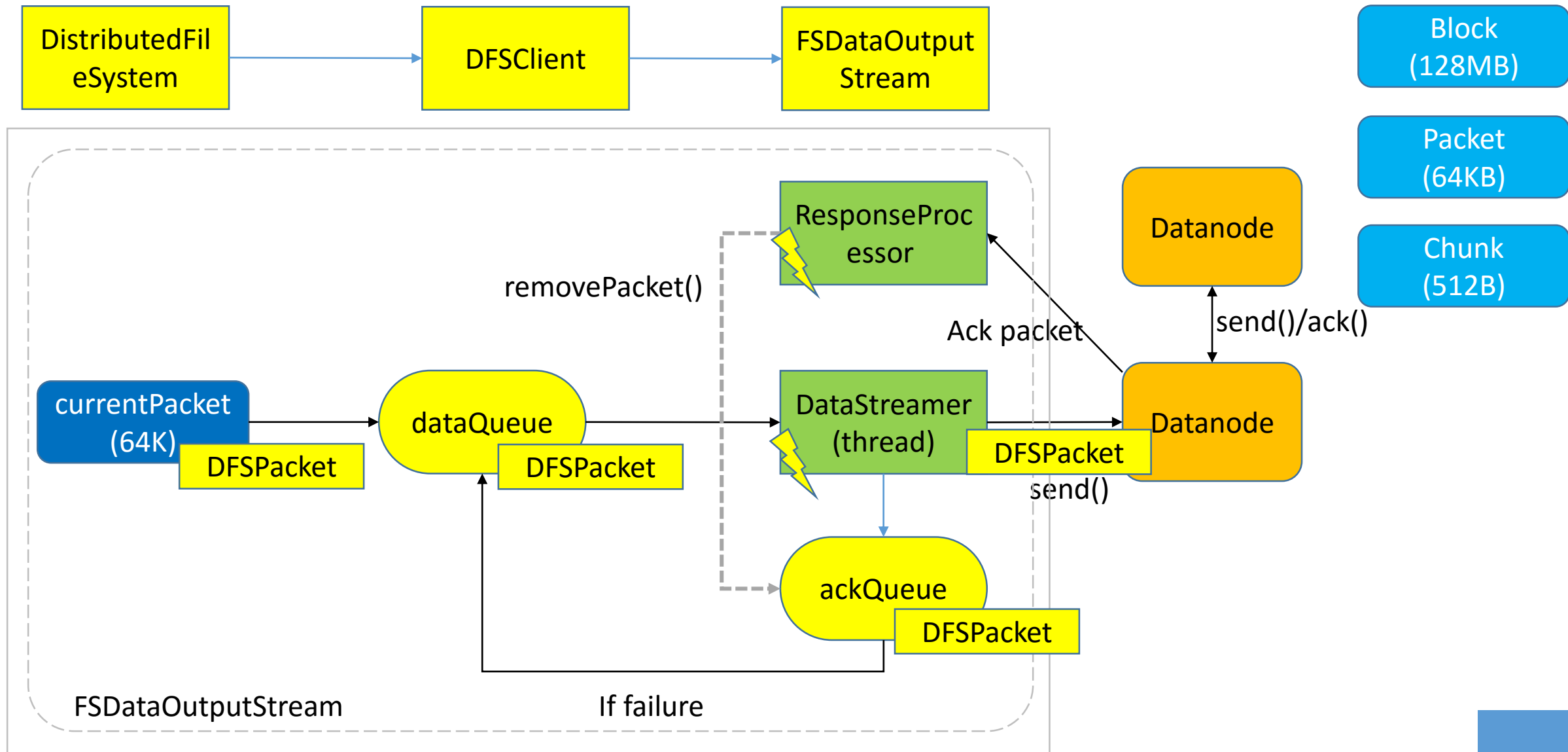
Определение расстояния



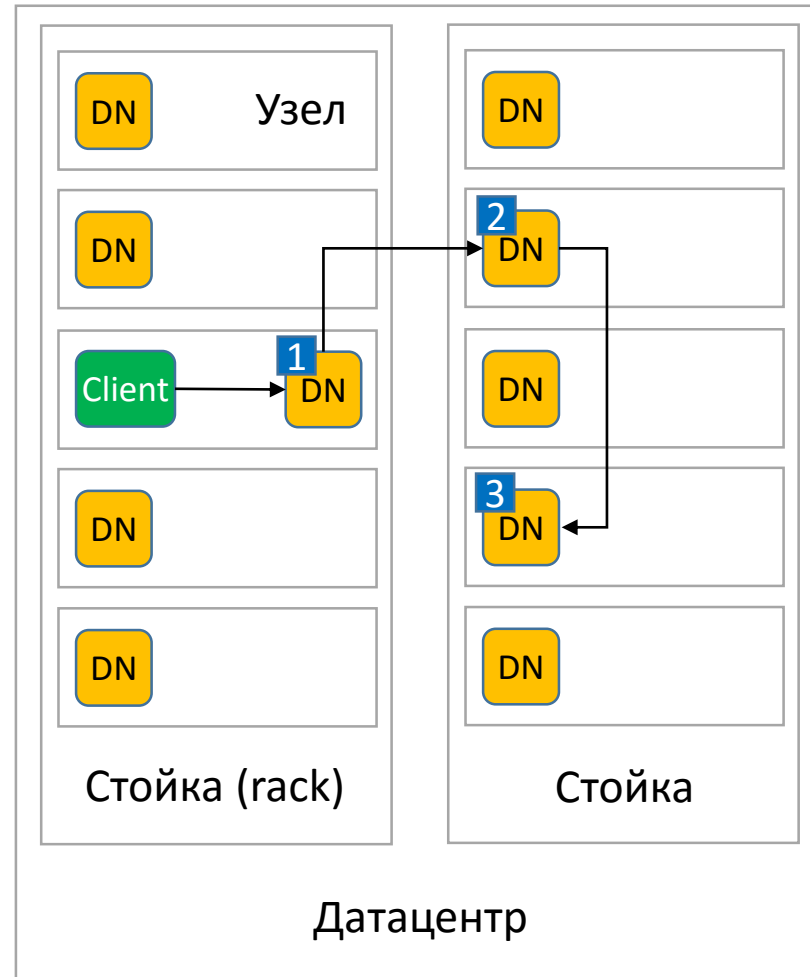
Запись данных



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



Расположение реплик



Что теряем, если выйдет из строя Datanode/Namenode?

- 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 загружает его обратно в **Namenode**

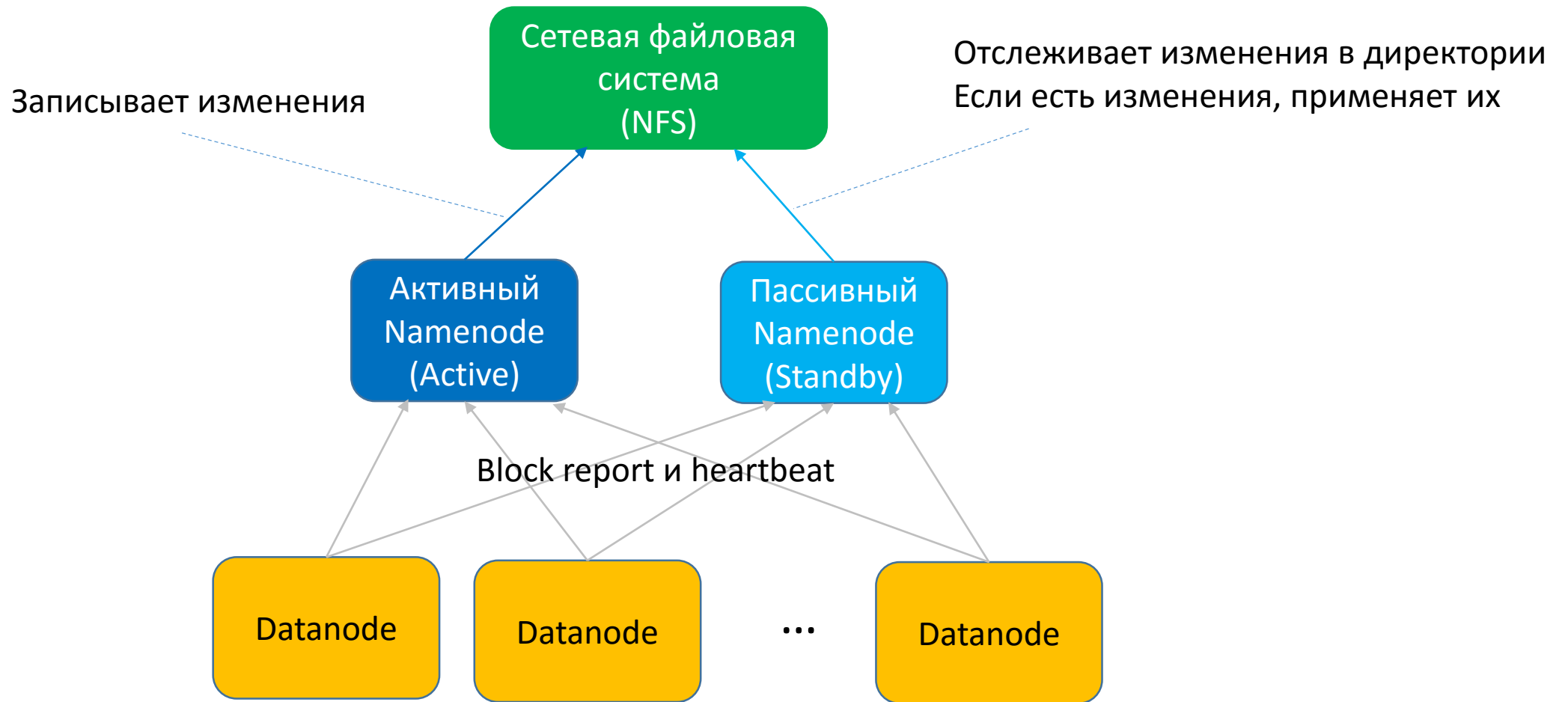
По умолчанию интервал 3600 сек., размер edits для старта 64MB

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

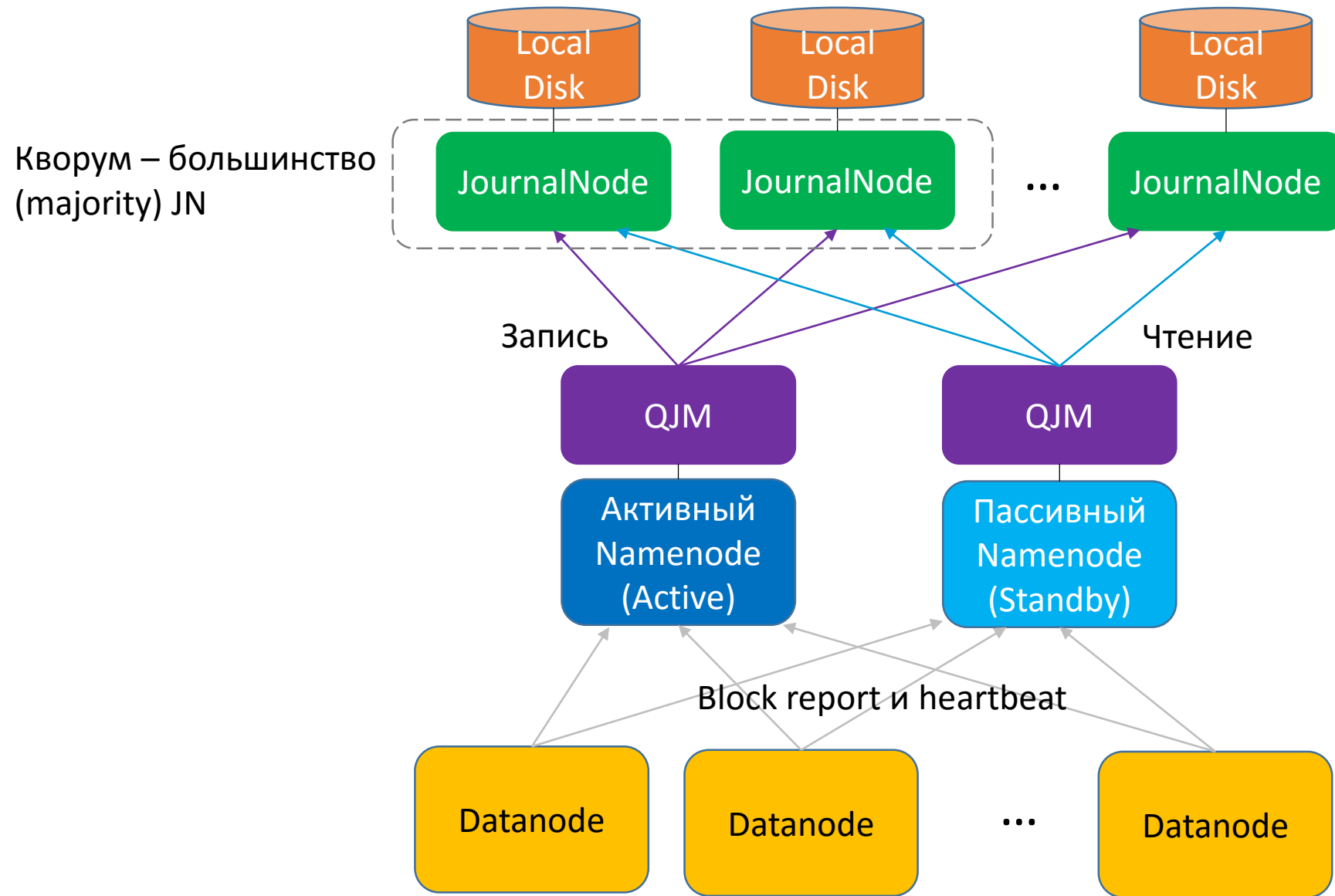
- Позволяет быстро восстановить данные на **пассивном Namenode**.
- **Используется** при 1) выходе из строя активного Namenode, 2) при плановой поддержке (например, при обновлении ПО на узле, где работает активный Namenode)
- **Активный Namenode** отвечает за все операции клиента в кластере
- Datanode'ы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

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

HDFS HA. NFS. Синхронизация Active NN и Standby NN

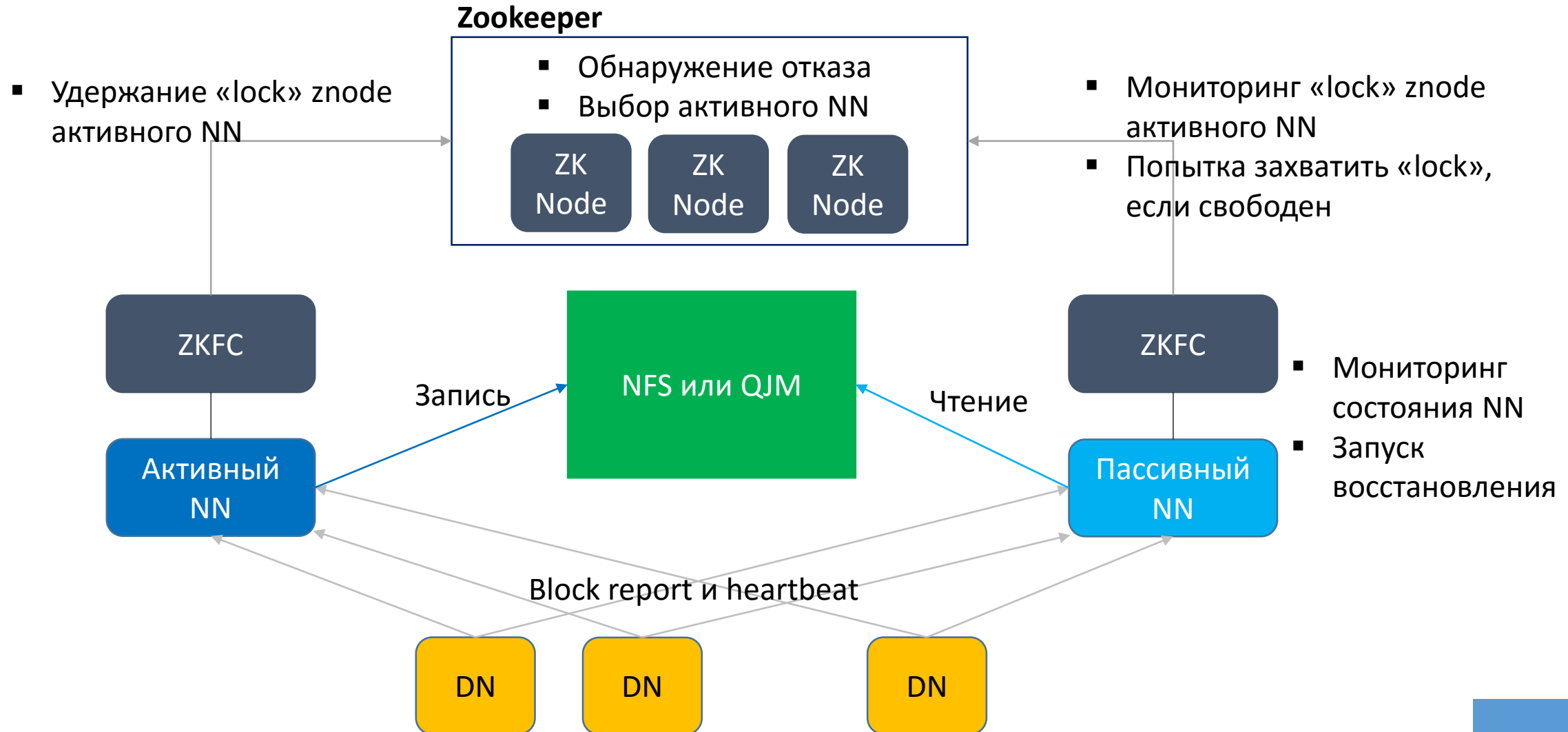


HDFS HA. QJM. Синхронизация Active NN и Standby NN



HDFS HA. Автоматический запуск восстановления

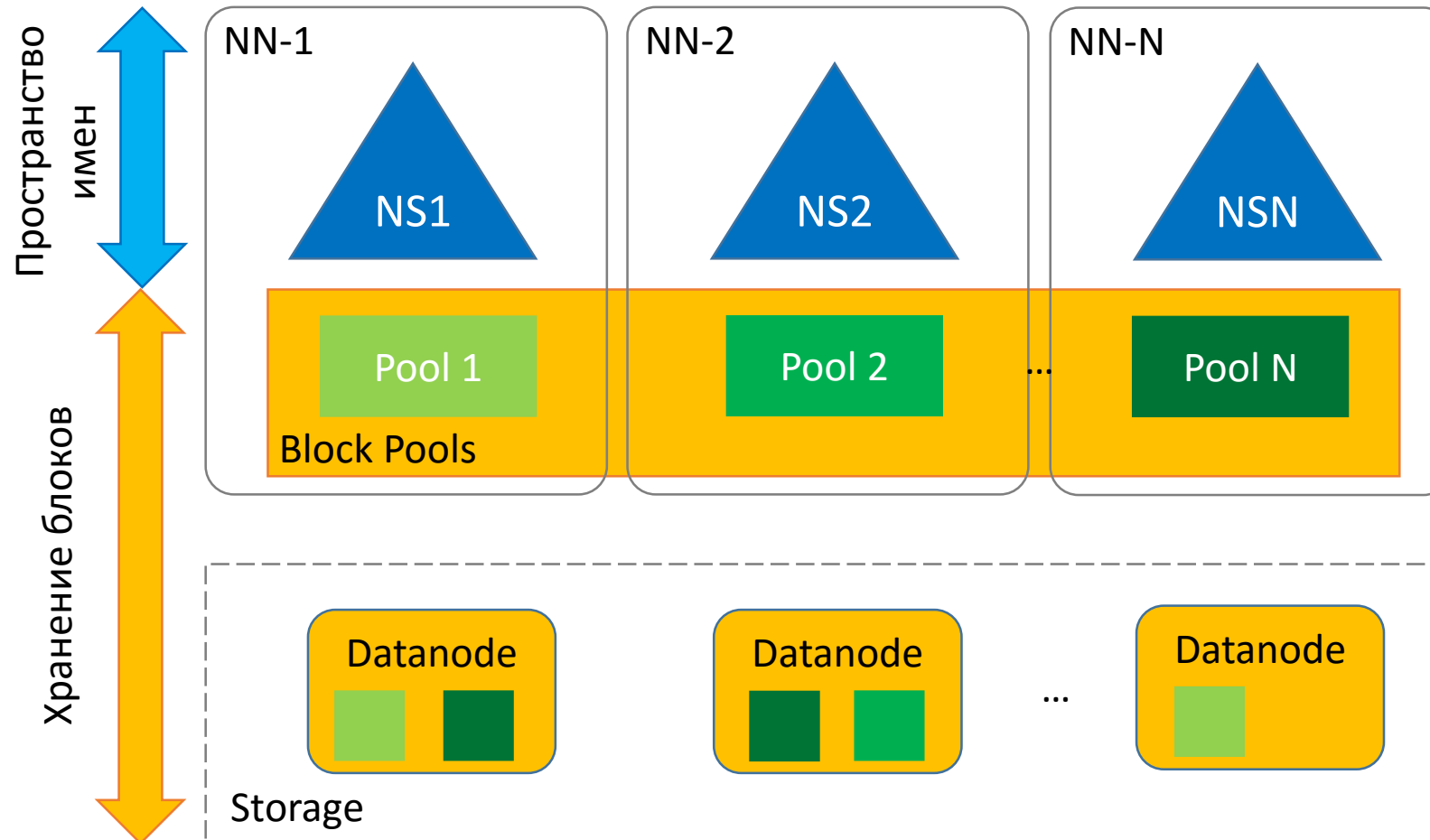
По умолчанию механизм восстановления запускается вручную



Federation

Block Pool – множество блоков, которые принадлежат одному пространству имен

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

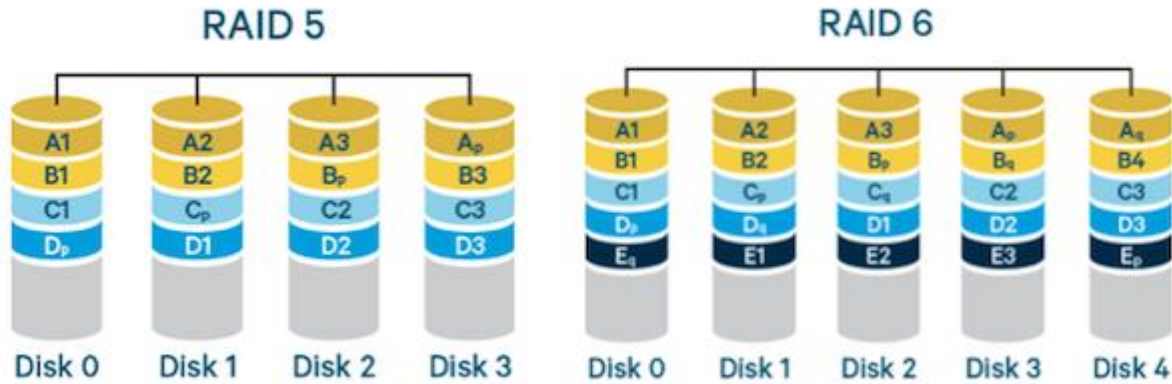


- Масштабируемость пространства имен
- Производительность (увеличивается пропускная способность)
- Изолированность

Hadoop 3.x - HDFS

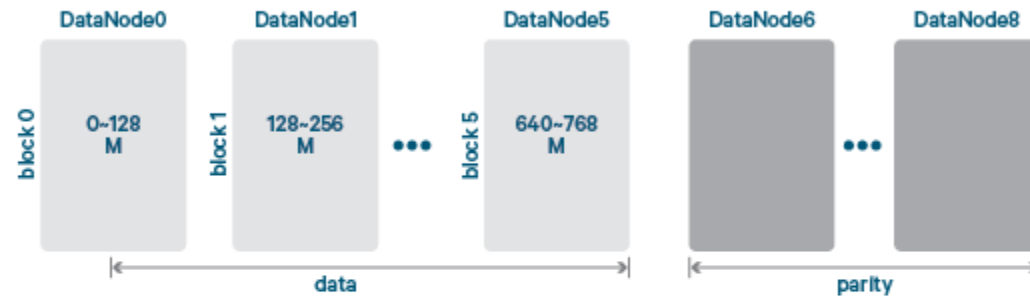
- Intra-DataNode Balancer
- Erasure Encoding

Erasure Encoding

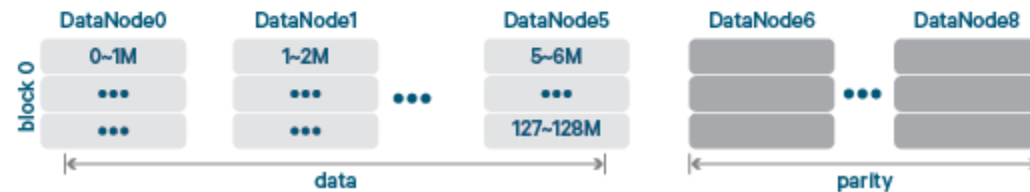


	Data Durability	Storage Efficiency
Single replica	0	100%
Three-way replication	2	33%
XOR with six data cells	1	86%
RS(6,3)	3	67%
RS(10,4)	4	71%

Contiguous



Striping



typically 64KB or 1MB

[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)