12. Hadoop. HDFS, MapReduce

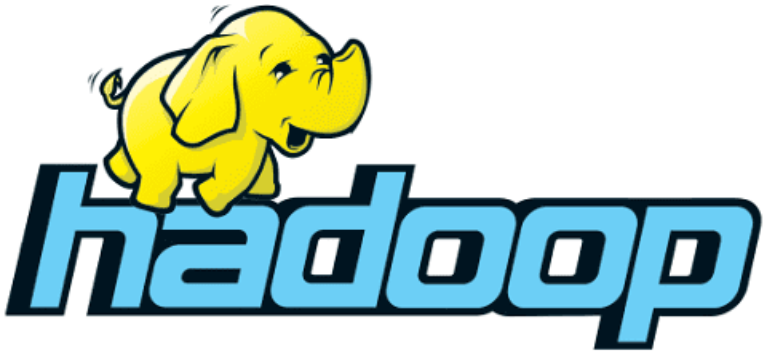
к.т.н., доцент кафедры ИиСП

Лучинин
Захар Сергеевич

Hadoop

Hadoop – это свободно распространяемый набор утилит, библиотек и фреймворк для разработки и выполнения распределённых программ, работающих на кластерах из сотен и тысяч узлов.

Основополагающая технология хранения и обработки больших данных (Big Data).



Компоненты Hadoop

- Hadoop Common – набор инфраструктурных программных библиотек и утилит
- HDFS – распределённая файловая система
- YARN – система планирования заданий
- Hadoop MapReduce – платформа выполнения распределённых MapReduce-вычислений

Что такое HDFS?

- HDFS (Hadoop Distributed File System) — файловая система, предназначенная для хранения файлов больших размеров.

- HDFS основан на Google File System (GFS) — распределенная файловая система, созданная компанией Google в 2000 году. Реализация является коммерческой тайной, но общие принципы построения системы опубликованы в 2003 году

Задача Google

Проблема

- Необходима система хранения данных при индексации интернета (Google Search)
- Хранение информации о пользователях и их поведении (Google Analytics)

Требования к системе хранения

- Масштабируемость
- Отказоустойчивость

Применение HDFS

- Хранение больших файлов
  - Терабайты или петабайты
  - Файлы с размером от 64 Мб
  - Миллионы, но не миллиарды
- Работа на “обычных” серверах, не суперкомпьютеры
- Может использоваться для MapReduce задач

Ограничения HDFS

- Не подходит для хранения большого количества маленьких файлов из-за накладных расходов на ресурсы сервера и неудовлетворительной скорости доступа к файлам
- Имеются ограничения по доступу к файлу
  - Нет “быстрого” способа изменять содержимое файлов
  - Возможно только дописывать информацию в конец файла

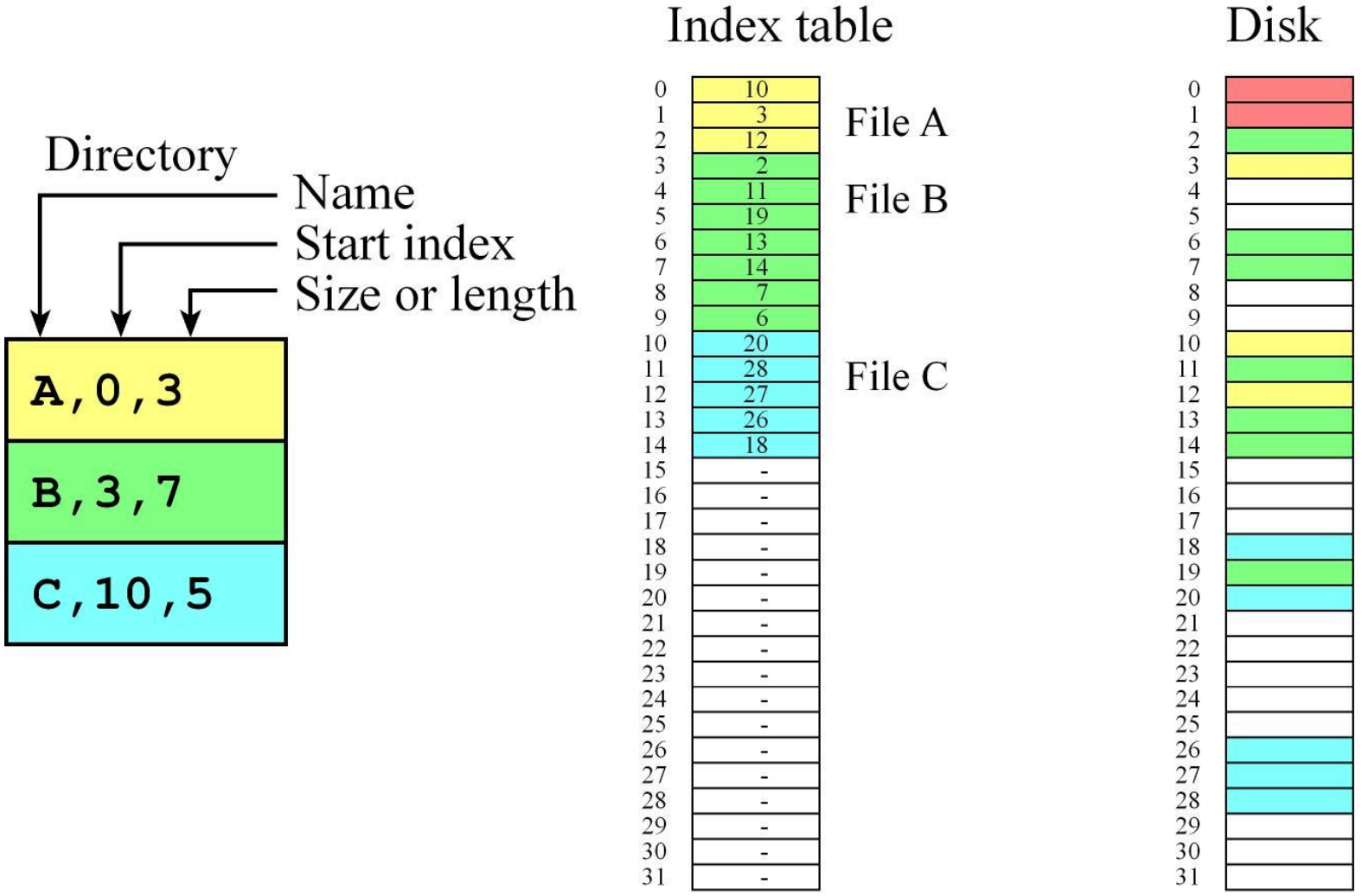
Задачи файловой системы

File system is a method and data structure that the operating system uses to control how data is stored and retrieved.

Задачи файловой системы:

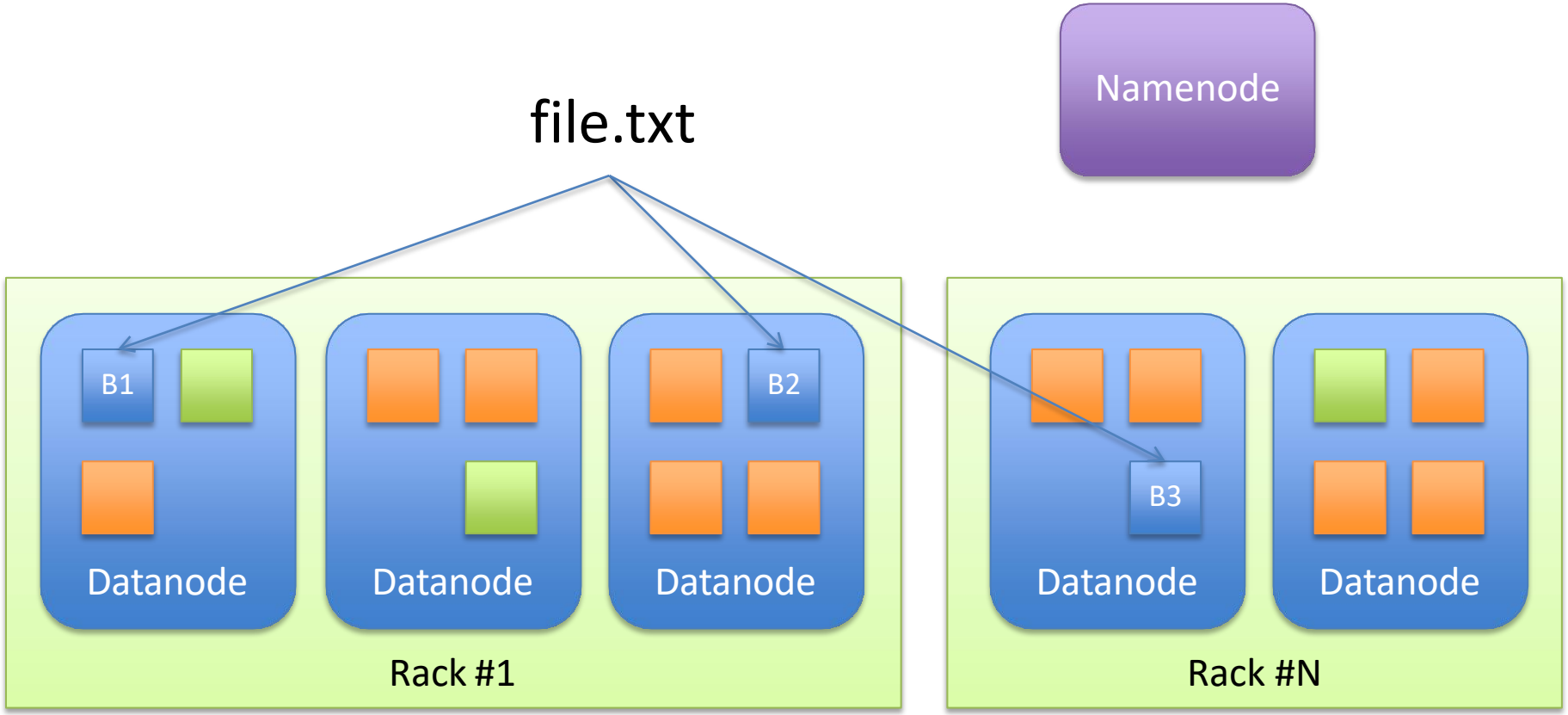
- Управление данными на низком – физическом уровне (хранение на диске блоков файл)
- Управление метаинформацией (путь к файлу, доступ к файлу)

Размещение файлов в ФС



Файл в HDFS

file.txt = block 1 + block 2 + block 3



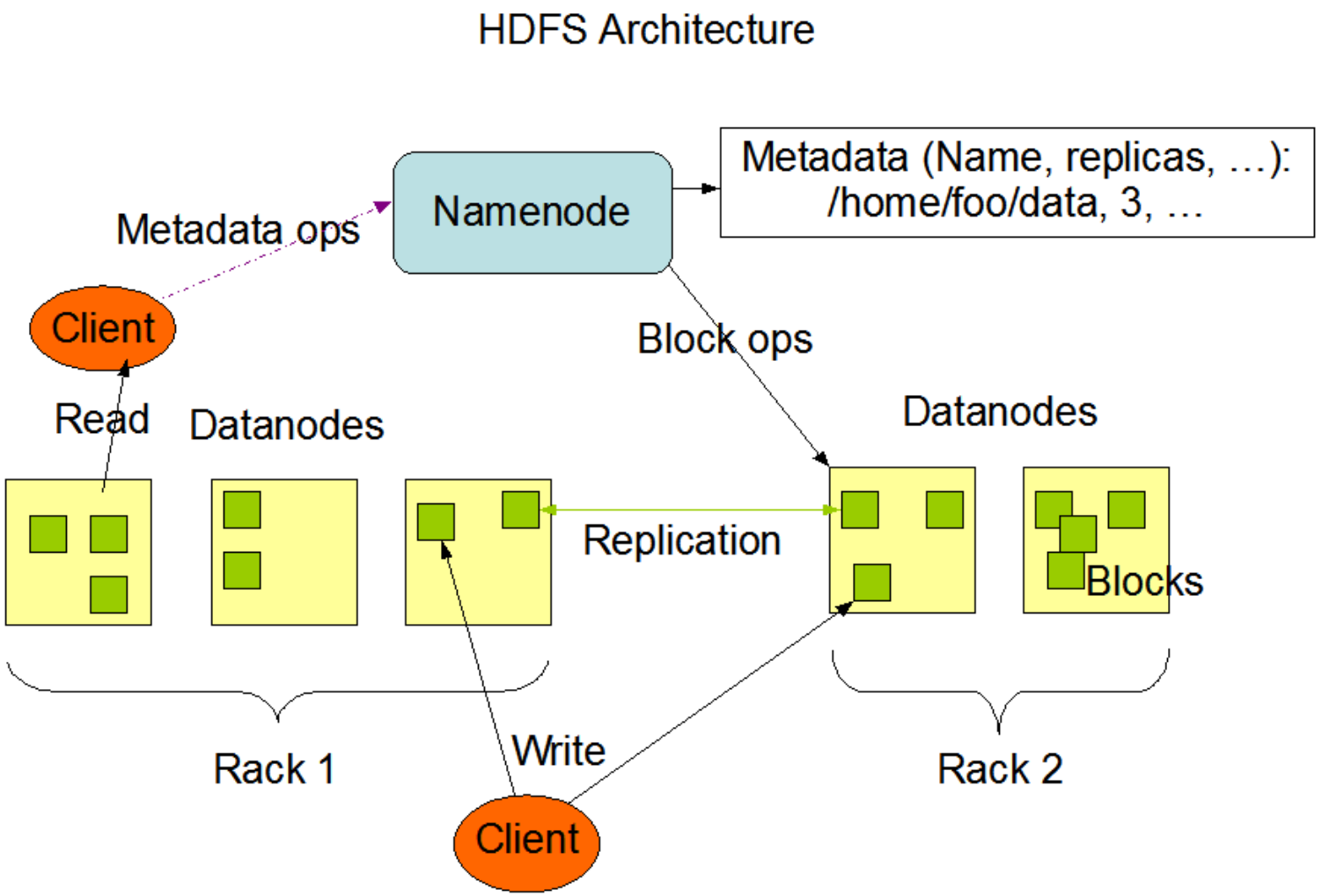
Блоки в HDFS

- Стандартный размер блоков 64Мб или 128Мб (Даже если файл несколько Кб, то он будет занимать размер блока)

- Цель снизить стоимость seek time по сравнению со скоростью передачи данных (transfer rate)

- При seek time = 10ms и transfer rate = 100 MB/s Для получения seek time равного 1% от transfer rate размер блока должен быть 100Мб

Архитектура HDFS



NameNode

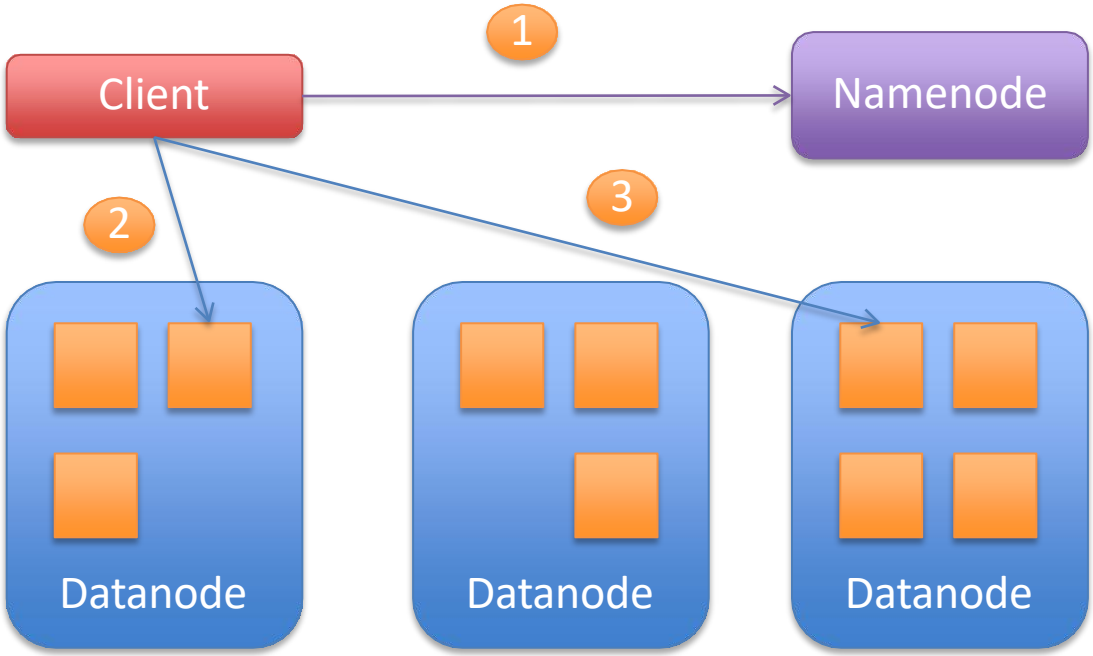
- Запускается на 1й (выделенной) машине
- Выполняет операции по обслуживанию namespace - открытие, закрытие, переименование файлов и директорий
- Назначает права доступа на файлы и директории
- Определяет маппинг блоков данных на DataNodes
- Для быстрого доступа вся мета-информация о блоках хранится в ОЗУ

DataNodes

- Отвечают за хранение данных
- Отвечает за чтение и запись запросов от клиентов
- Создает, удаляет и реплицирует блоки данных под командой NameNode
- DataNode ничего не знает о HDFS файлах. Она хранит каждый блок данных в разных файлах в локальной файловой системе.

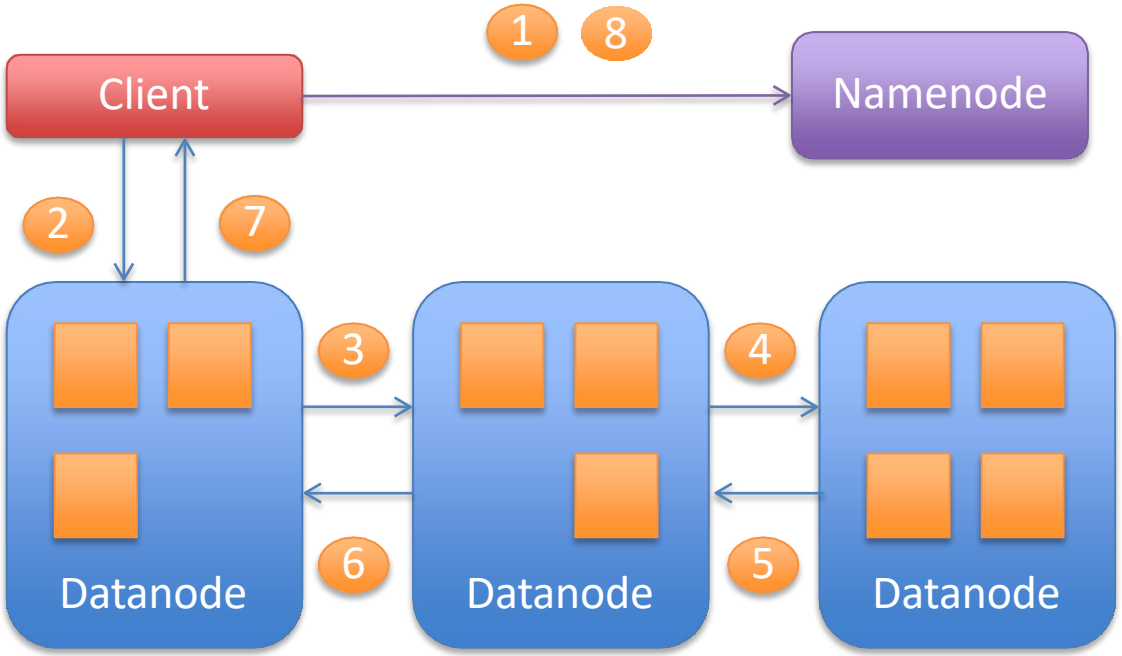
Чтение файла с HDSF

1. Получить расположение блоков
2. Прочитать 1й блок файла
3. Прочитать 2й блок файла



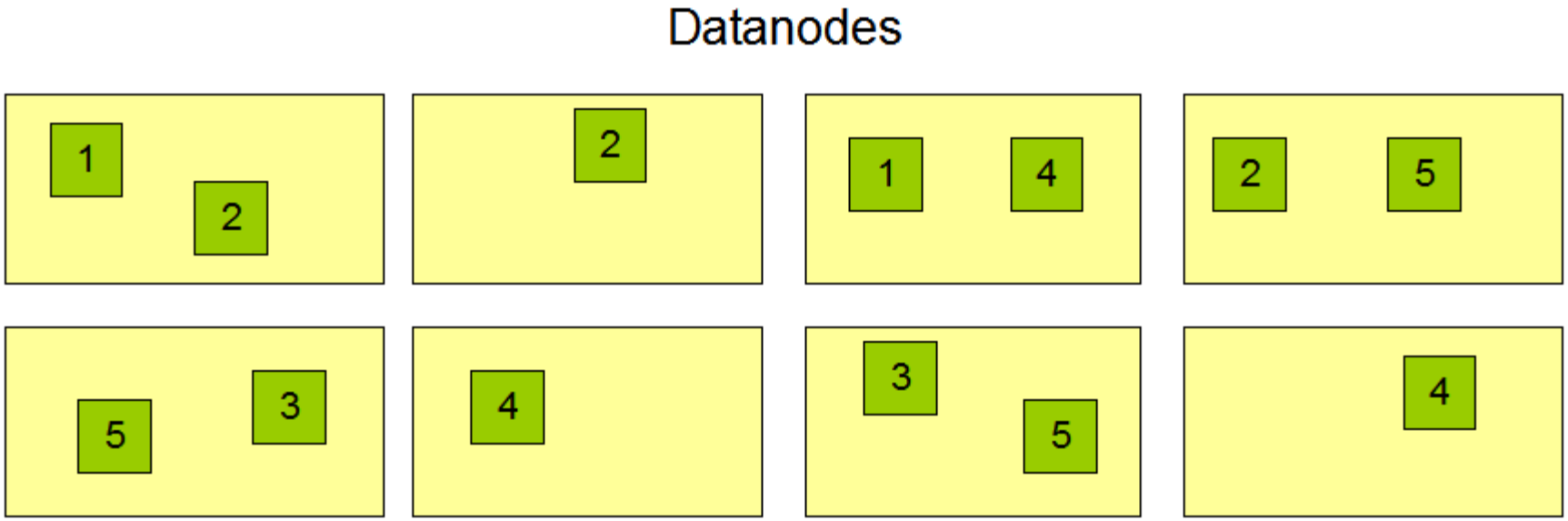
Запись файла в HDFS

1. Создать новый файл в namespace на NN и определить топологию блоков
2. Отправить данные на 1-ю DN
3. Отправить данные на 2-ю DN
4. Отправить данные на 3-ю DN
5. Подтверждение Success/Failure
6. Подтверждение Success/Failure
7. Подтверждение Success/Failure
8. Фиксация изменений



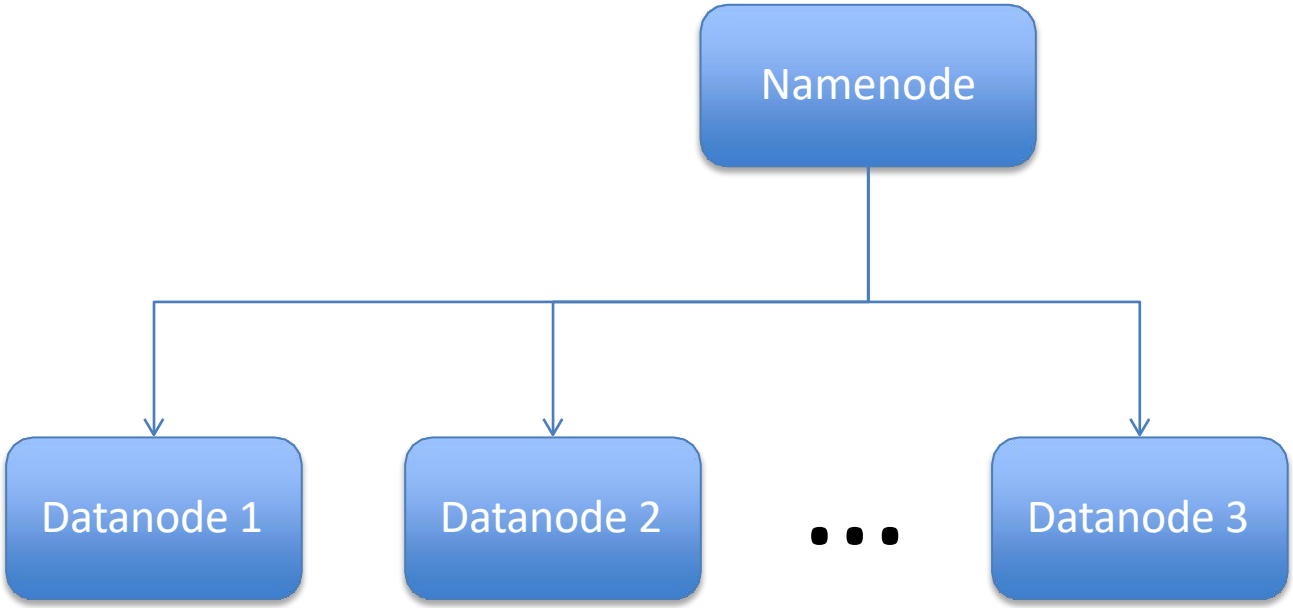
Репликация

- Каждый блок хранится в нескольких экземплярах (на разных серверах)
- Если сервер выходит из строя, то блок доступен для чтения



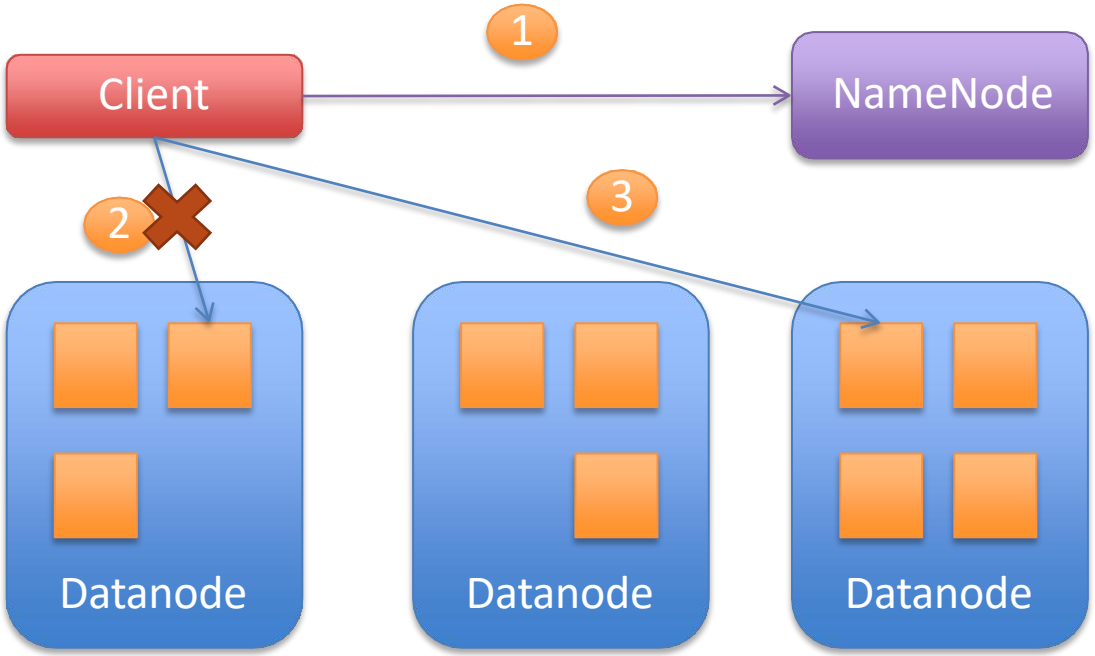
Репликация детали реализации

- NameNode периодически опрашивает (собирает HeartBeat) и blockreport с каждой DataNode. Blockreport содержит список всех блоков со всех DataNode.
- NameNode управляет “дорепликацией” если продадает один из серверов



Чтение файла при потере сервера

1. Получить расположение блоков
2. Прочитать 1й блок файла с host1
3. Прочитать 1й блок файла с host3



Ответ от NameNode:

file -> block[]
block -> [host]

MapReduce

MapReduce – это модель распределённых вычислений от компании Google, используемая в технологиях Big Data для параллельных вычислений над очень большими (до нескольких петабайт) наборами данных

Именование происходит от двух функций (Map и Reduce), которые необходимо определить для выполнения расчета

Задача Google

- Рассчитать объем трафика (количество посещений) каждой страницы.
- Подсчитать сколько страниц посетил каждый пользователь.

Шаг Map

- Map принимает на вход список значений и некую функцию, которую затем применяет к каждому элементу списка и возвращает новый список

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

Функция map превращает входной документ в набор пар (слово, 1)

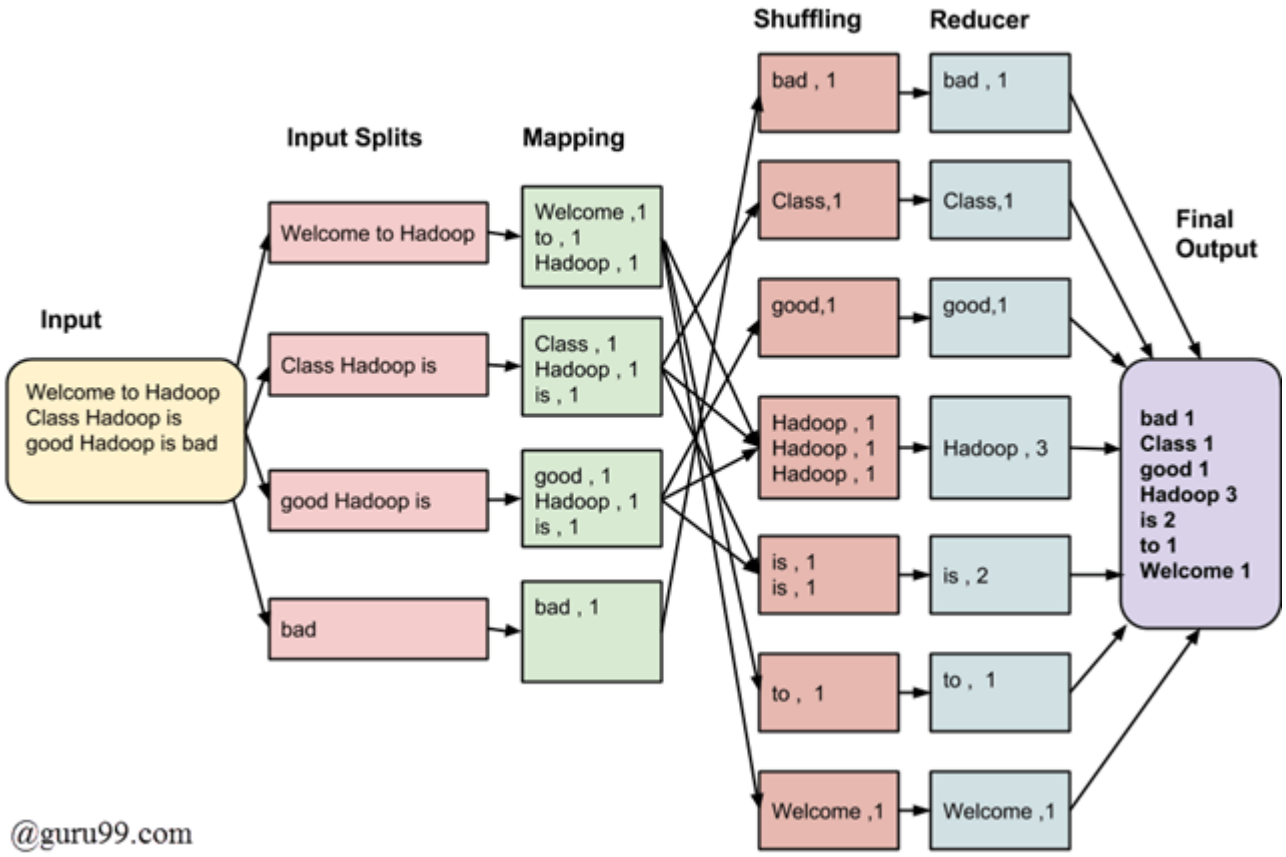
Шаг Reduce

- Reduce (свёртка) преобразует список к единственному атомарному значению при помощи заданной функции, которой на каждой итерации передаются новый элемент списка и промежуточный результат

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

Reduce суммирует эти единицы, возвращая финальный ответ для слова

Иллюстрация работы MapReduce



Пример

Задача: имеется csv-лог рекламной системы вида:

<user_id>,<country>,<city>,<campaign_id>,<creative_id>,<payment>
11111,RU,Moscow,2,4,0.3
22222,RU,Voronezh,2,3,0.2

Необходимо: рассчитать среднюю стоимость показа рекламы по городам России.

```
def map(record):
    user_id, country, city, campaign_id, creative_id, payment = record.split(",")
    payment=float(payment)
    if country == "RU":
        yield city, payment


def reduce(city, payments):
    yield city, sum(payments)/len(payments)
```

MapReduce в Hadoop