Hadoop

Введение в Hadoop и MapReduce.

HDFS

Данные

- 2006 год – 0.18 зеттабайт
- 2011 год – 1.8 зеттабайт (10^21)

Нью-Йорская фондовая биржа 1 терабайт данных в день

Internet Archive более 2 петабайт и растет со скоростью 20 терабайт в месяц

Большой адронный коллайдер до 15 петабайт данных в год

Хранение и анализ данных

- 1990 год – 1370 Мбайт, скорость передачи до 4.4 Мбайт/с, 5 минут
- 2010 год – несколько терабайт, скорость передачи 100 Мбайт/с, 2.5 часа

Используем 100 дисков, на каждом из которых хранится 1/100 часть данных => при параллельной работе дисков данные будут прочитаны за 2 минуты

Hadoop

- Система надежного общего хранения и анализа данных
  - HDFS обеспечивает хранение
  - MapReduce - анализ

Hadoop и РСУБД

- Скорость позиционирования улучшается медленнее скорости передачи данных
- Позиционирование – процесс перемещения считывающей головки к определенному месту диска для чтения или записи данных.
- Скорость позиционирования определяет задержку при выполнении дисковых операций, тогда как скорость передачи данных определяют пропускную способность канала взаимодействия с диском
- Если в схеме обращения к данным преобладают операции позиционирования, то чтение и запись больших частей набора данных займут больше времени, чем при потоковых операциях, выполняемых со скоростью передачи данных

Hadoop и РСУБД

Параметр	Традиционная РСУБД	MapReduce
Размер данных	Гигабайты	Петабайты
Доступ	Интерактивный и пакетный	Пакетный
Обновления	Многократное чтение и запись	Однократная запись, многократное чтение
Структура	Статическая схема	Динамическая схема
Целостность	Высокая	Низкая
Масштабирование	Нелинейное	Линейное

Что такое Hadoop

- Инфраструктура (framework) для параллельной обработки больших объемов данных (терабайты)
- Особенности:
  - Функциональное программирование
  - Автоматическое распараллеливание
  - Перемещение вычислений к данным
- Open Source, http://hadoop.apache.org

Концепции

- Парадигмы программирования:
  - Императивное программирование (ИП)
  - Функциональное программирование (ФП)
- Работа с данными:
  - Перемещение данных к вычислительным ресурсам (ПДкВ)
  - Перемещение вычислений к данным (ПВкД)

Концепции

- Наиболее популярная сейчас технология:
  - императивное программирование + перемещение данных к вычислениям
- Примеры:
  - MPI
  - GPU

GPU nVidia

- Высокая производительность:
  - nVidia Tesla M2050/2070 – 0,5 TFlops double
- Шаги вычислений:
  - Копирование данных в память GPU
  - Обработка данных в GPU
  - Копирование данных в память хоста
- Программист полностью управляет процессом вычислений и перемещения данных

Недостатки ИП + ПДкВ

- MPI и GPU эффективны при:
  - Небольших объемах данных
  - Высокой сложности вычислений
  - Небольшом количестве узлов (сотни)
- Терабайты данных перемещать долго
- Управлять логикой передачи данных на тысячи узлов сложно

Проблемы разработки для крупных параллельных систем

- Масштабирование на тысячи узлов
- Эффективное распределение нагрузки
- Эффективный обмен данными в процессе вычислений
- Обработка отказов вычислительных узлов
- Императивное программирование:
  - Все эти задачи программист должен решать сам

Функциональное программирование

- Программист описывает функцию, которую надо вычислить, но не процесс вычисления
- Входные данные не изменяются, создаются новые
- Поток данных жестко встроен в программу

Задачи Hadoop/MapReduce

- Эффективная обработка терабайтов данных
- Автоматическое распараллеливание на тысячи узлов
- Автоматическое распределение нагрузки
- Автоматическая обработка отказов оборудования
- Простота использования

Примеры приложений

- Распределенный grep
- Распределенная сортировка
- Инвертированный индекс
- Подсчет количества запросов к URL
- Реверсивный web-link граф

История

- Google:
  - 2003 - The Google File System
  - 2004 - MapReduce: Simplified Data Processing on Large Clusters
- 2005 - Open Source поисковик Apache Nutch использует MapReduce
- 2006 – Open Source реализация MapReduce выделяется в отдельный проект Apache Hadoop

Кто использует Hadoop

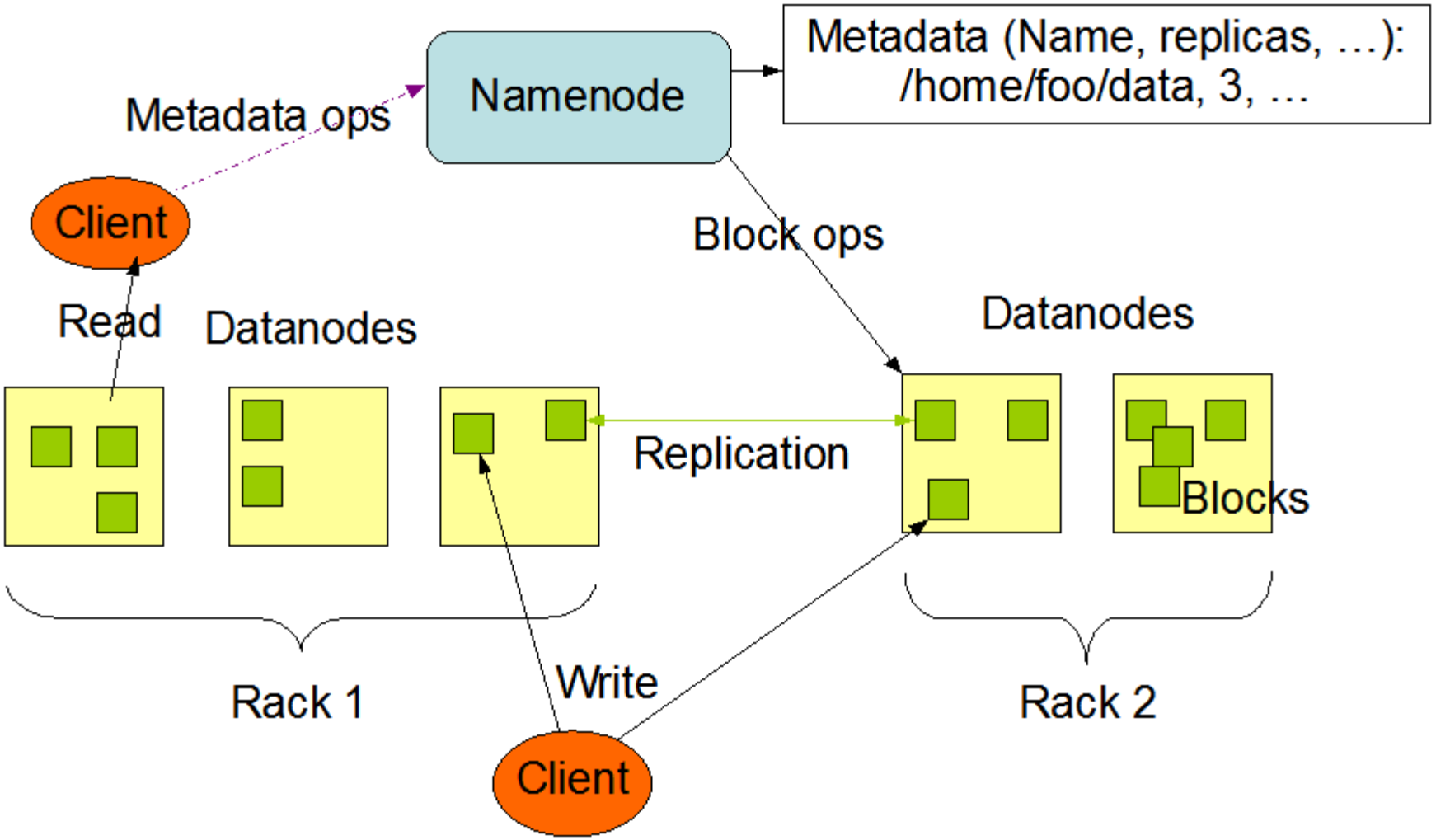
Состав Hadoop

- Hadoop Common – общие компоненты Hadoop
- Hadoop HDFS – распределенная файловая система
- Hadoop MapReduce – реализация MapReduce на Java

Hadoop HDFS

- Специализированная распределенная файловая система для хранения Терабайтов данных
- Цели разработки:
  - Надежное хранение данных на дешевом ненадежном оборудовании
  - Высокая пропускная способность ввода-вывода
  - Потоковый доступ к данным
  - Упрощенная модель согласованности: WORM
- Архитектура аналогична Google File System

Архитектура Hadoop HDFS



Архитектура HDFS

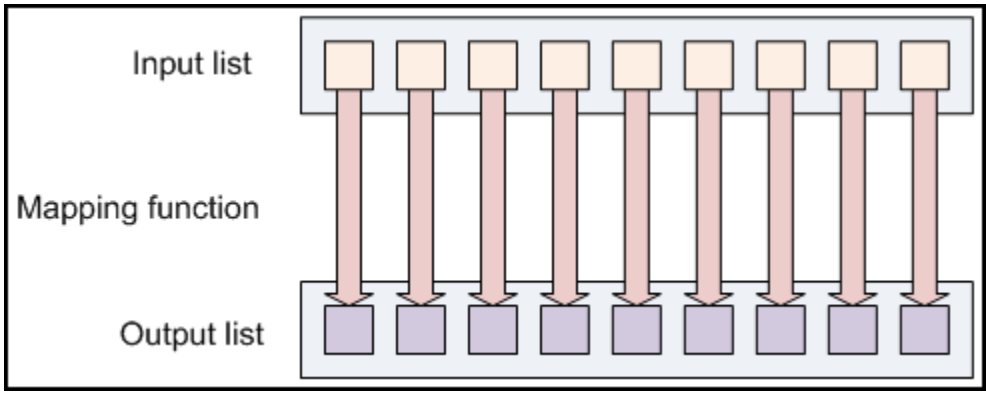
- Узлы хранения – серверы стандартной архитектуры
- Данные хранятся на внутренних дисках серверов
- Единое адресное пространство
- Параллельное чтение и запись на узлы – высокая пропускная способность

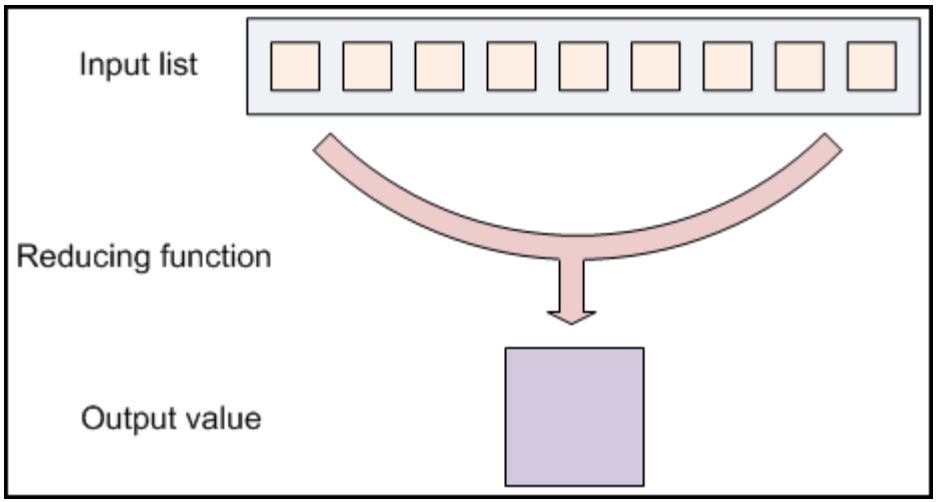
MapReduce

- Программная модель параллельной обработки **больших объемов данных** за путем разделения на **независимые** задачи
- MapReduce разработан в Google для поисковой системы
- Использует функциональное программирование, обработку списков

Функции MAP и Reduce

- Названия заимствованы из функциональных языков (LISP, ML)
- Обработка списков





MapReduce в Hadoop

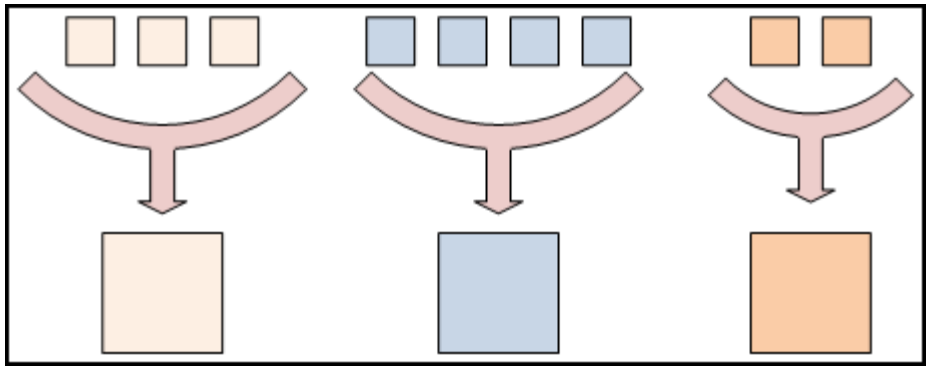
- Списки пар: ключ-значение

  AAA-123 65mph, 12:00pm

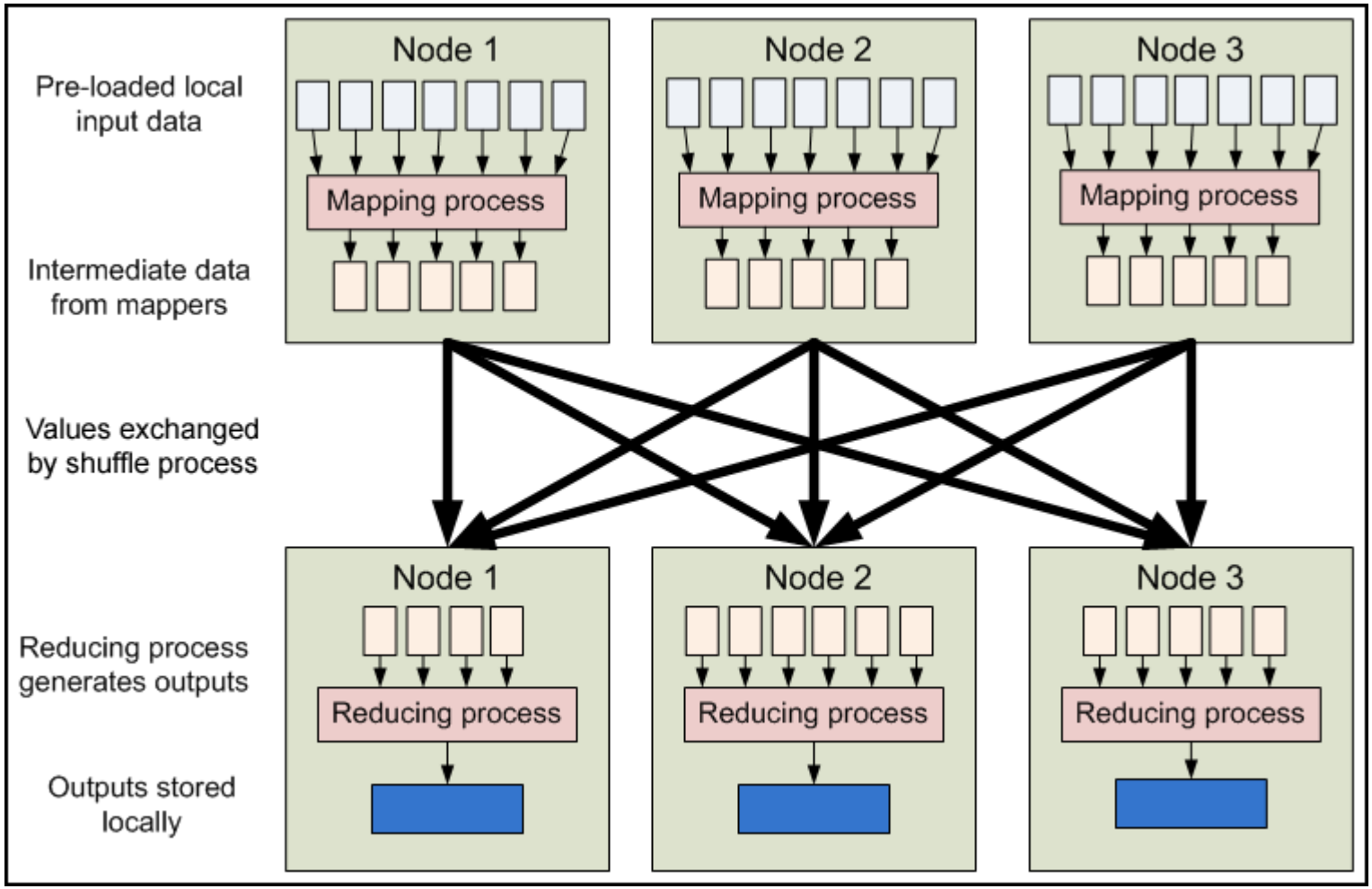
  ZZZ-789 50mph, 12:02pm

  AAA-123 40mph, 12:05pm

- Reduce выполняется отдельно для разных ключей

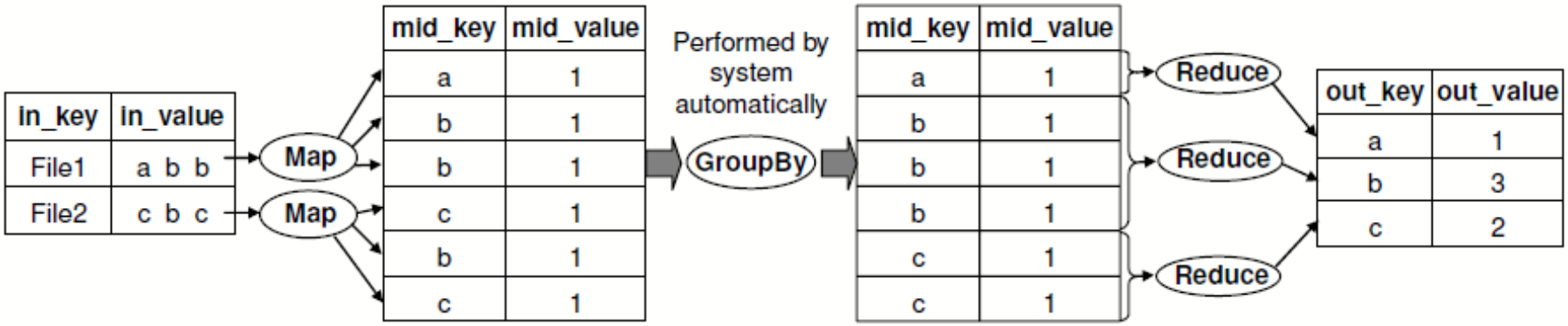


Поток данных MapReduce

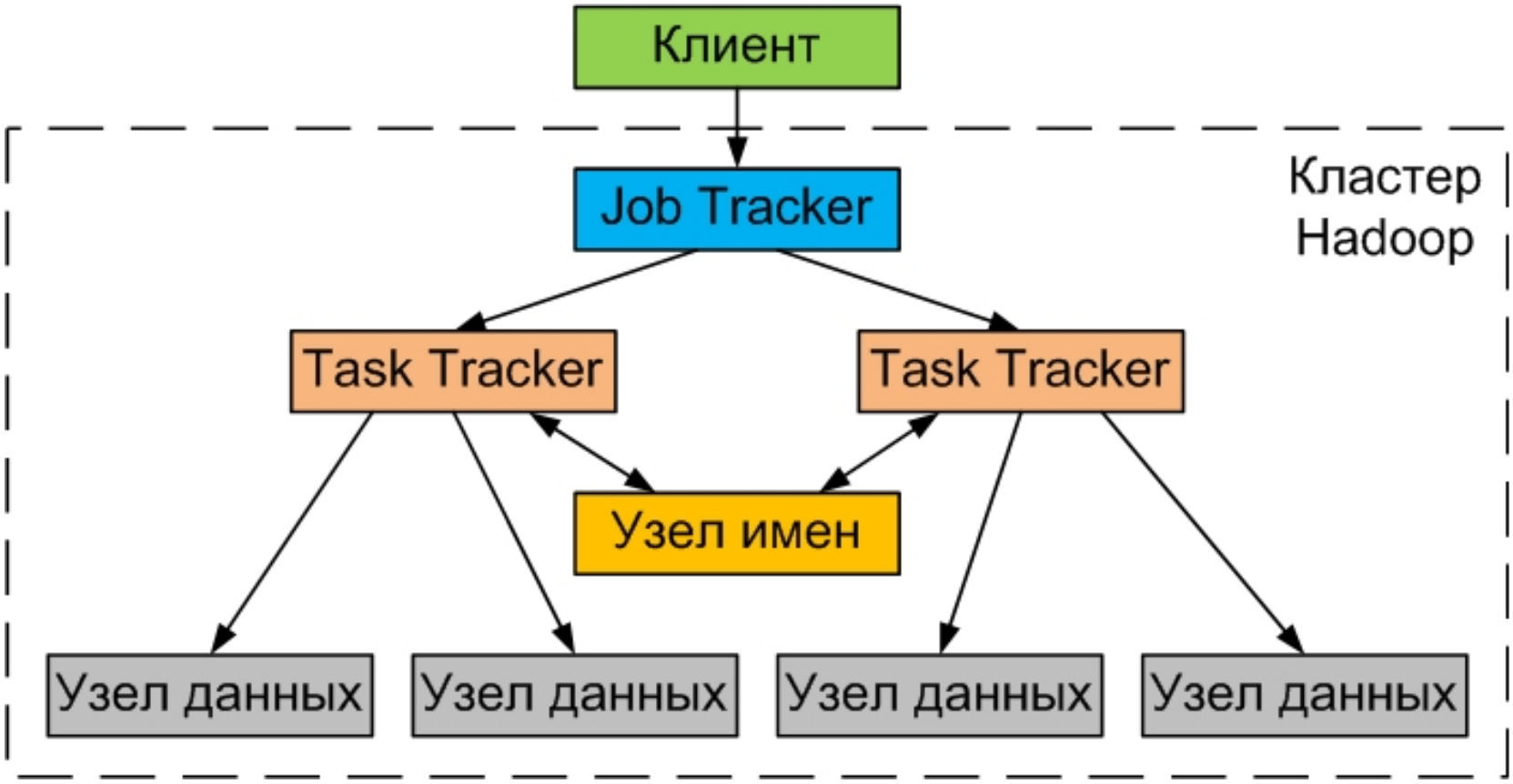


Пример WordCount

- Подсчет количества слов в файлах



Архитектура Hadoop



Перемещение вычислений к данным

- Задача запускается на том узле хранения, который содержит данные для обработки (фаза MAP)
- Перемещаются только входные списки для Reduce, их объем мал (как правило)

Результаты Hadoop в TeraSort

Байты	Узлы	Maps	Reduces	Время
$5*10^{11}$	1 406	8 000	2 600	59 секунд
10^{12}	1 460	8 000	2 700	62 секунды
10^{14}	3 452	190 000	10 000	173 минуты
10^{15}	3 658	80 000	20 000	975 минут

Источник: Owen O'Malley and Arun C. Murth. Winning a 60 Second Dash with a Yellow Elephant.

ОС и режимы работы

- Java 6
- Поддерживаемые ОС:
  - Linux (продуктив)
  - Windows (только тестирование)
  - Любой UNIX (не гарантируется)
- Режимы работы:
  - Локальный
  - Псевдо-распределенный
  - Распределенный

Программирование Hadoop

- Java API
- Hadoop Plugin для Eclipse
- Hadoop Streaming - другие языки:
  - Shell
  - Python
  - Ruby
  - и др.

Системы на основе Hadoop

- Pig – высокоуровневый язык потоков данных
- HBase – распределенная база данных
- Cassandra – multi-master база данных без единой точки отказа
- Hive – хранилище данных (warehouse)
- Mahout – машинное обучение и извлечение знаний

Распределенная файловая система HDFS

- Мотивация использования распределенных файловых систем
- Архитектура HDFS
- Команды работы с HDFS
- Права доступа в HDFS
- Работа с HDFS из Java программ

Мотивация

- Что нужно для эффективной обработки терабайтов данных?
  - Большая емкость
  - Высокая производительная
  - Надежность

Традиционное решение

- Системы хранения данных
  - Емкость: сотни и тысячи дисков
  - Производительность: сотни ГБ/с
  - Надежность: RAID, дублирование компонентов, репликация
- Примеры: EMC Symmetrix VMAX, Hitachi VSP, HP XP20000
- Недостаток: высокая стоимость (миллионы долларов)

Распределенные файловые системы

- Можно ли получить емкость, производительность и надежность дешево?
- Да, можно. Google:
  - “The Google File System”, Sanjay Ghemawat, Howard Gobioff, Shun-Tak Leung. *Proceedings of the 19th ACM Symposium on Operating Systems Principles*, 2003, pp. 20-43.
  - Для хранения данных используются диски недорогих обычных серверов
  - Независимые диски объединяются в распределенную файловую систему GFS

Преимущества распределенных файловых систем

- Высокая емкость:
  - Много серверов с внутренними дисками
- Высокая производительность:
  - Параллельная запись на диски, много сетевых интерфейсов
- Высокая надежность:
  - Репликация данных на разные серверы
- Низкая стоимость:
  - Серверы стандартной архитектуры с Linux

HDFS

- Hadoop Distributed File System (HDFS) – распределенная файловая система, входящая в состав Hadoop
- Основывается на архитектуре Google File System
- HDFS - специализированная файловая система для приложений Hadoop

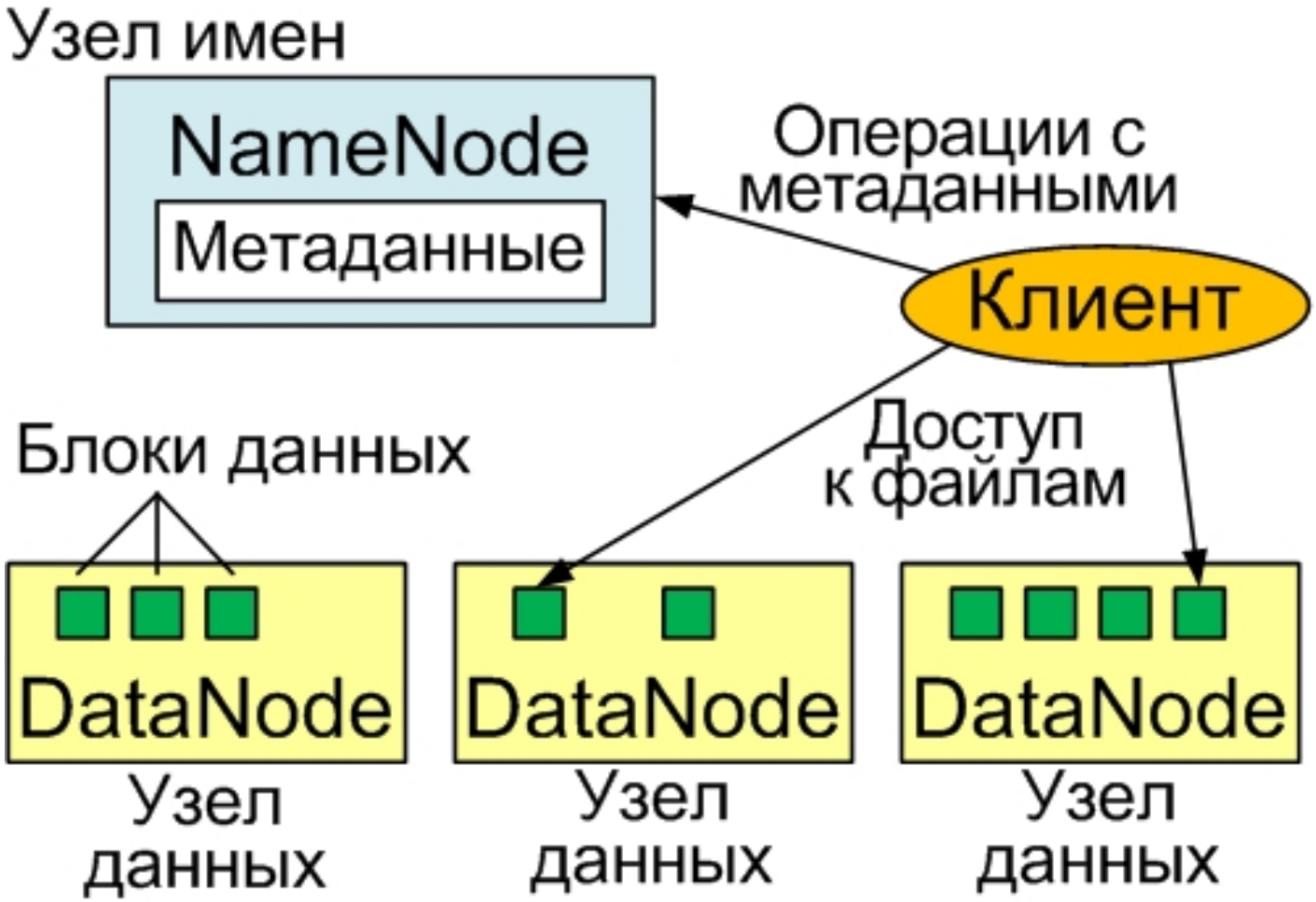
Потребности приложений Hadoop

- Типовое приложение – поисковый робот
  - Файлы индексов содержимого Web больших размеров
  - Файлы индексов записываются один раз, а затем только читаются (без изменений)
  - Потоковые операции ввода-вывода
  - Пакетная обработка

Ограничения HDFS

- Оптимизация для потоковых операций с большими файлами
  - Случайный доступ работает медленно
- Модель доступа к файлам WORM (Write-Once-Read-Many)
  - Запись в файл производиться только один раз, потом только чтение
- Не поддерживается POSIX
  - Нельзя подмонтировать, не работают стандартные Linux команды ls, cp, mkdir и т.п.
- Кэширование не используется
  - Накладные расходы слишком велики

Архитектура HDFS



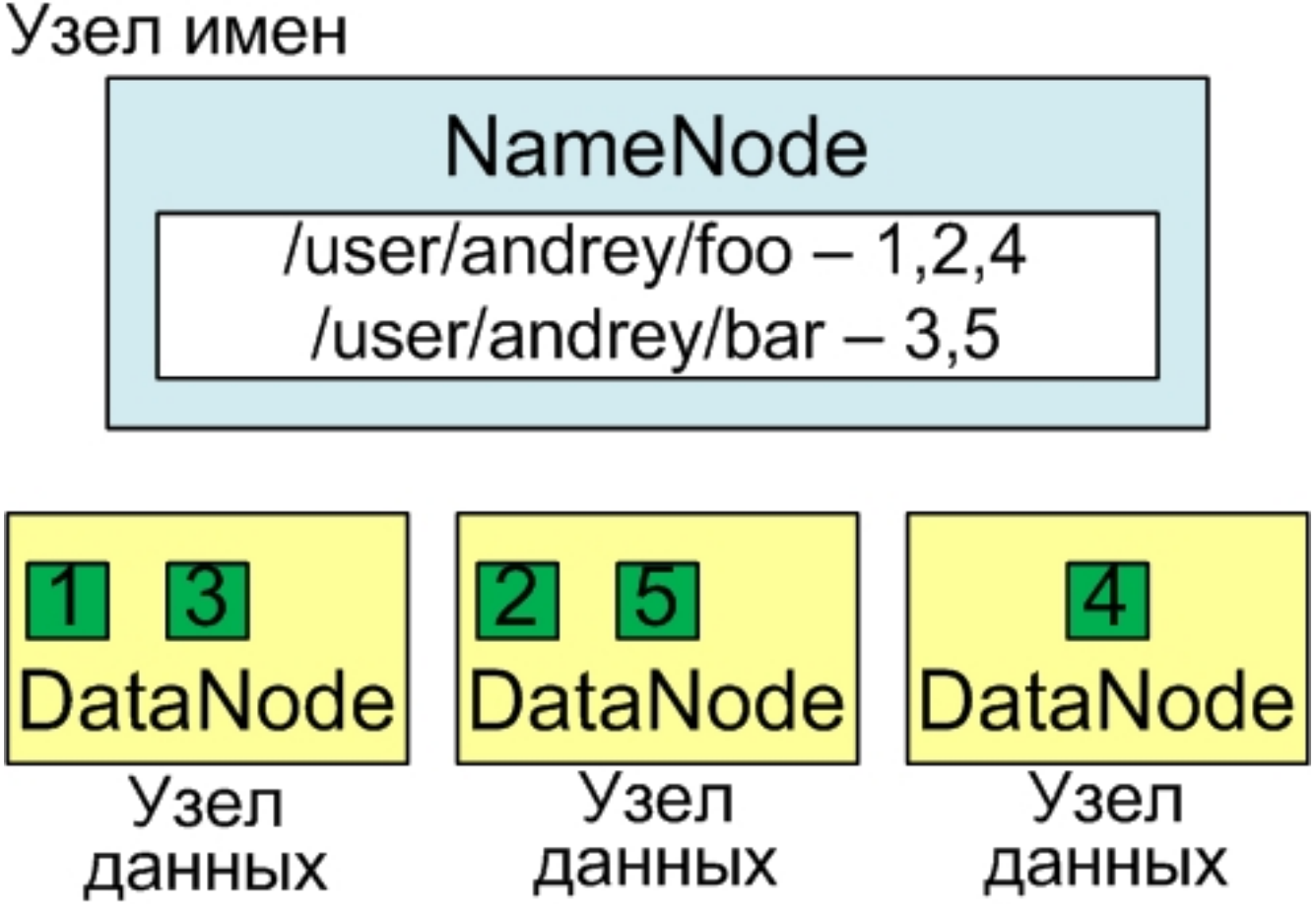
Архитектура HDFS

- Namenode (узел имен):
  - Управляющий узел
  - Обеспечивает единое пространство имен
  - Регулирует доступ клиентов
  - Хранит метаданные
- Datanode (узел данных)
  - Хранит данные
- Узлы имен и данных – серверы Linux (как правило)

Хранение файлов в HDFS

- Блочная структура:
  - Файл разбивается на блоки одинакового размера (64МБ по умолчанию)
  - Блоки хранятся на одном или нескольких узлах хранения
  - Возможна репликация блоков
- Узел имен хранит метаданные о распределении блоков по узлам хранения

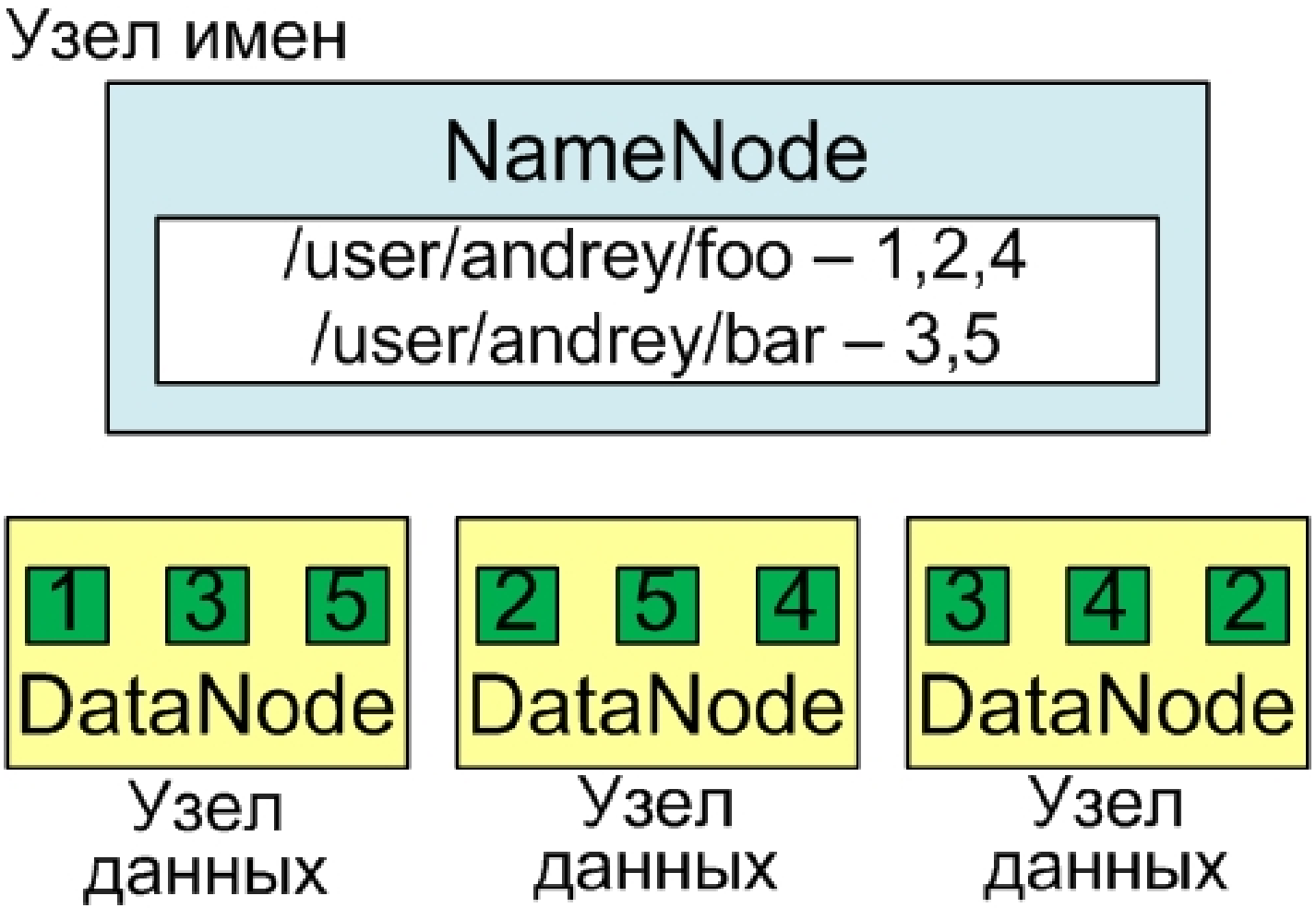
Хранение файлов в HDFS



Репликация

- В большом кластере всегда будут неисправные узлы
- Для защиты от сбоев HDFS использует репликацию – хранение нескольких копий блока
- Фактор репликации – количество копий блока (3 шт. по умолчанию)
- Отказ сервера снижает производительность, но не приводит к потере данных

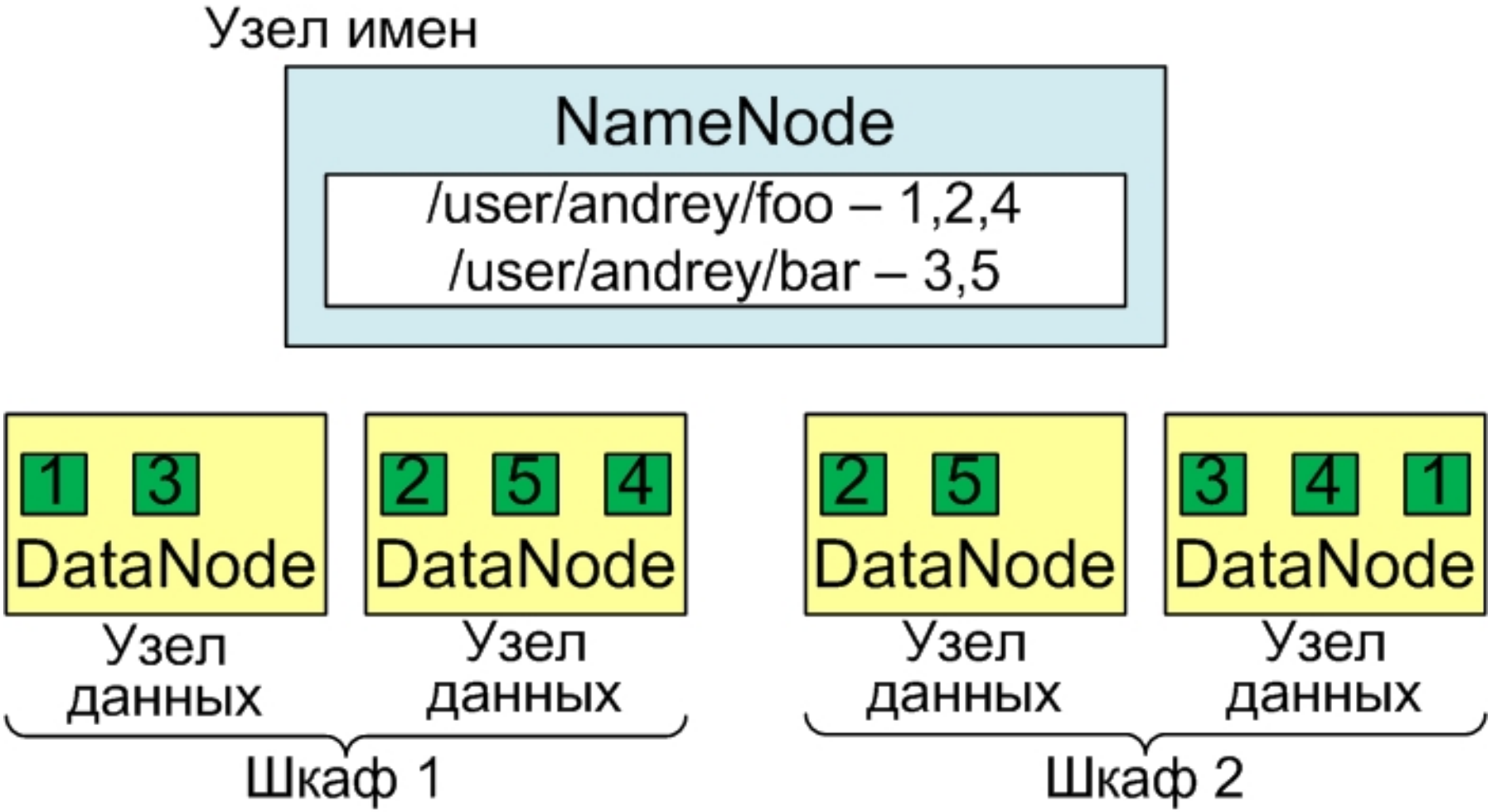
Репликация


Узел имен
NameNode
/user/andrey/foo – 1,2,4
/user/andrey/bar – 3,5
1 3 5
DataNode
Узел данных
2 5 4
DataNode
Узел данных
3 4 2
DataNode
Узел данных

Rack Awareness

- Дополнительный механизм защиты от сбоя группы серверов
- Rack – серверный шкаф:
  - Отключение питания всего шкафа
  - Потеря сетевого соединения со шкафом
- HDFS умеет распределять реплики между разными шкафами
- Отказ всего шкафа не приводит к потере данных

Rack Awareness



Доступ к HDFS

- Блоки HDFS распределены по разным серверам:
  - HDFS нельзя подмонтировать
  - Не работают стандартные Linux команды: ls, cp, mv и .т.
- Для работы с HDFS используются специальные команды:
  - `$bin/hadoop dfs -cmd`

Структура HDFS

- Корневой каталог HDFS - /
- Домашние каталоги пользователей - /user/$USER
- Временный каталог - /tmp
- Нет понятия текущий каталог
  - Нет команд cd, pwd
- Пути:
  - Полные – начиная с /
  - Относительные – из домашнего каталога пользователя

Просмотр файлов в каталоге

- Домашний каталог:

```
hadoop@hadoop:~/hadoop$ bin/hadoop dfs -ls
Found 3 items
-rw-r--r--   1 hadoop supergroup    0 2011-06-22 13:58 /user/hadoop/file1
-rw-r--r--   1 hadoop supergroup    0 2011-06-22 13:58 /user/hadoop/file2
-rw-r--r--   1 hadoop supergroup    0 2011-06-22 13:58 /user/hadoop/file3
```

- Корневой каталог:

```
hadoop@hadoop:~/hadoop$ bin/hadoop dfs -ls /
Found 2 items
drwxr-xr-x   - hadoop supergroup          0 2011-05-17 18:32 /tmp
drwxr-xr-x   - hadoop supergroup          0 2011-05-18 14:35 /user
```

Просмотр файла

- Список файлов:

```
hadoop@hadoop:~/hadoop$ bin/hadoop dfs -ls
Found 3 items
-rw-r--r--   1 hadoop supergroup    0 2011-06-22 13:58 /user/hadoop/file1
-rw-r--r--   1 hadoop supergroup    0 2011-06-22 13:58 /user/hadoop/file2
-rw-r--r--   1 hadoop supergroup    0 2011-06-22 13:58 /user/hadoop/file3
```

- Просмотр файла:

```
hadoop@hadoop:~/hadoop$ bin/hadoop dfs -cat file1
Hello, world!
Hello, Hadoop!
```

- Просмотр файла, полный путь:

```
hadoop@hadoop:~/hadoop$ bin/hadoop dfs -cat /user/hadoop/file1
Hello, world!
Hello, Hadoop!
```

Запись файлов в HDFS

- Команда:
  - `$bin/hadoop dfs –put` *`localSrc hdfsDest`*
- Копирует из локальной файловой системы в HDFS
- Работает как с файлами, так и с каталогами
- Если файл уже существует, выдает ошибку
- Синоним -copyFromLocal

Запись файлов в HDFS

- Запись файла:

```
hadoop@hadoop:~/hadoop$ bin/hadoop dfs -put /tmp/file1 /user/hadoop
```

- Запись каталога:

```
hadoop@hadoop:~/hadoop$ bin/hadoop dfs -put /tmp/dir1 /user/hadoop
hadoop@hadoop:~/hadoop$ bin/hadoop dfs -ls dir1
Found 3 items
-rw-r--r--   1 hadoop supergroup  0 2011-06-22 14:49 /user/hadoop/dir1/filea
-rw-r--r--   1 hadoop supergroup  0 2011-06-22 14:49 /user/hadoop/dir1/fileb
-rw-r--r--   1 hadoop supergroup  0 2011-06-22 14:49 /user/hadoop/dir1/filec
```

- Файл существует:

```
hadoop@hadoop:~/hadoop$ bin/hadoop dfs -put /tmp/file1 file1
put: Target file1 already exists
```

Получение файлов из HDFS

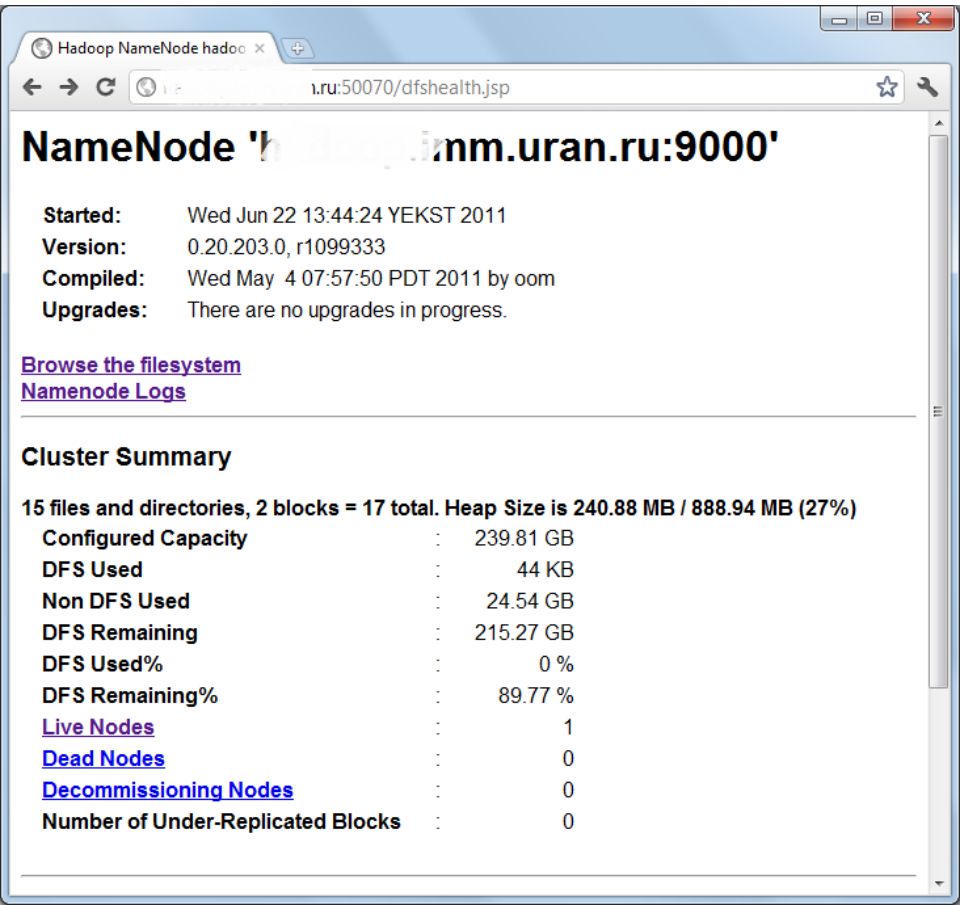
- Команда:
  - `$bin/hadoop dfs –get` *`hdfsSrc localDest`*
- Копирует из HDFS в локальную файловую систему
- Работает как с файлами, так и с каталогами
- Синоним -copyToLocal

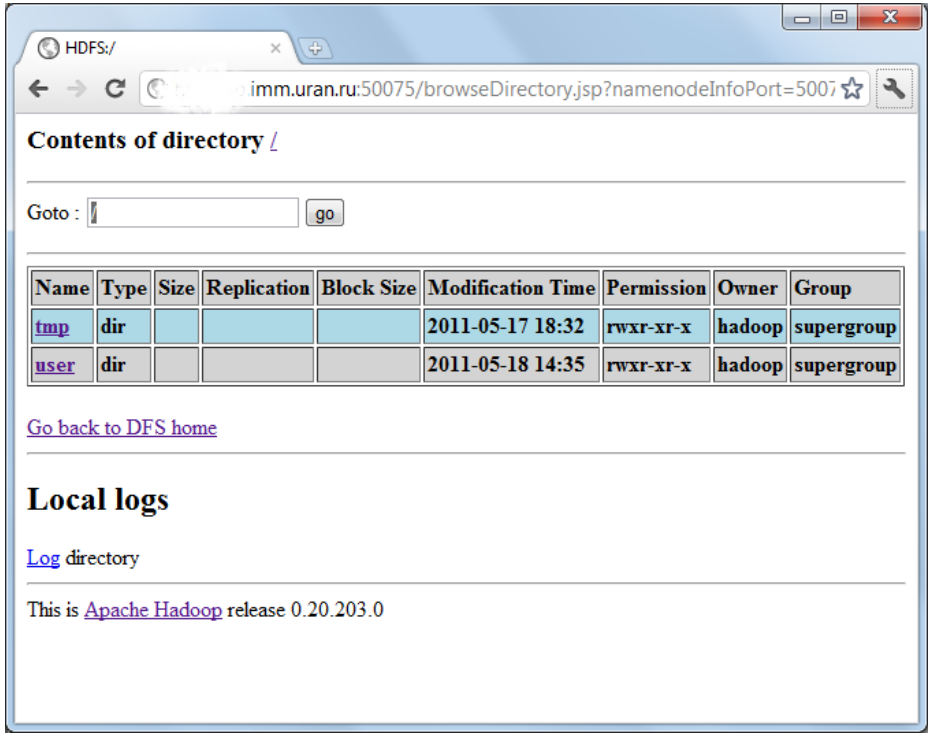
Команды для работы с HDFS

Команда	Назначение
`-mv` *`src dest`*	Перемещение файлов внутри HDFS
`-cp` *`src dest`*	Копирование файлов внутри HDFS
`-rm` *`path`*	Удаление файла или пустого каталога
`-rmr` *`path`*	Удаление файла или каталога рекурсивно
`-mkdir` *`path`*	Создание каталога (работает как mkdir –p в Linux)
`-stat` *`path`*	Выводит информацию по файлу или каталогу
`-tail [-f]` *`path`*	Вывод последнего килобайта файла (с -f выводит добавляемые данные)
`-help`	Перечень команд работы с HDFS

Web-интерфейс к HDFS

- http://namenode-hostname:50070





Права доступа в HDFS

- Модель прав доступа HDFS похожа на POSIX:
  - Файл имеет владельца (owner) и группу (group)
  - Права задаются отдельно для владельца, группы и всех остальных
  - Права доступа rwx
  - Нет sticky bit, setuid or setgid

Семантика прав доступа

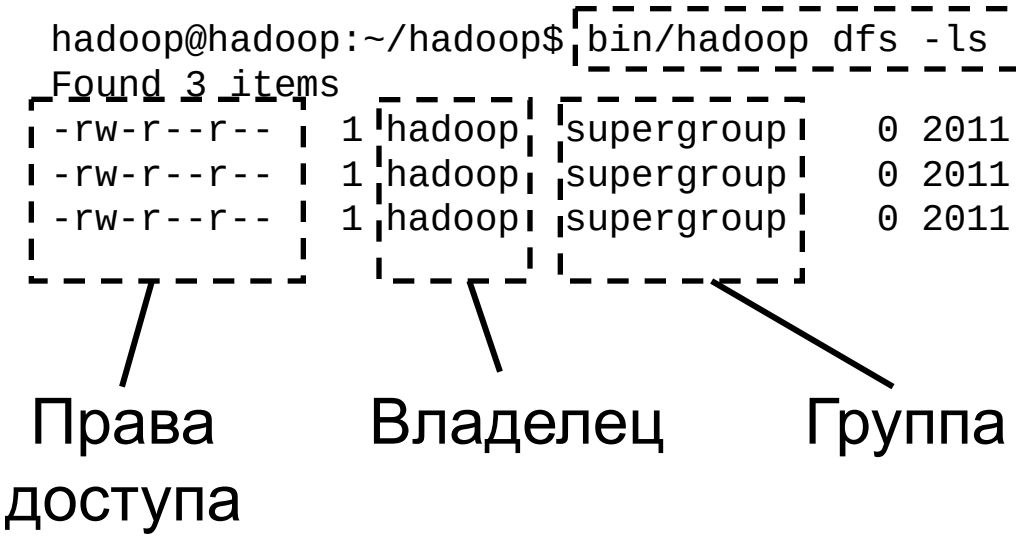
- Для файлов:
  - r – чтение
  - w – запись
  - x – не используется
- Для каталогов:
  - r – просмотр содержимого каталога
  - w – создание файлов или каталогов
  - x – доступ к файлам и подкаталогам

Пользователи HDFS

- Пользователи HDFS соответствуют пользователям Linux:
  - Пользователь: `whoami`
  - Список групп: `bash -c groups`
- Суперпользователь
  - Не действуют ограничения прав доступа
  - Пользователь, который запустил Hadoop
  - Нет постоянного суперпользователя

Просмотр прав доступа

```
hadoop@hadoop:~/hadoop$ bin/hadoop dfs -ls
Found 3 items
-rw-r--r--  1 hadoop supergroup   0 2011-06-22 13:58 /user/hadoop/file1
-rw-r--r--  1 hadoop supergroup   0 2011-06-22 13:58 /user/hadoop/file2
-rw-r--r--  1 hadoop supergroup   0 2011-06-22 13:58 /user/hadoop/file3
```



Управление правами доступа

- Команды

Команда	Назначение
`-chmod [-R] mode` *`path`*	Изменение прав доступа
`-chown [-R] owner[:group]` *`path`*	Изменение владельца (и группы)
`-chgroup [-R] grm` *`path`*	Изменение группы

- Опция *–R* – рекурсивные изменения
- Права доступа записываются как в Linux

Изменение прав доступа

- Исходные права доступа

```
$ bin/hadoop dfs -ls /user/Andrey/file1
-rw-r--r--   1 Andrey supergroup   0 2011-06-23 11:20 /user/Andrey/file1
```

- Цифровой режим

```
$ bin/hadoop dfs -chmod 600 /user/Andrey/file1

$ bin/hadoop dfs -ls /user/Andrey/file1
-rw-------   1 Andrey supergroup  0 2011-06-23 11:20 /user/Andrey/file1
```

- Символьный режим

```
$ bin/hadoop dfs -chmod g+rw /user/Andrey/file1

$ bin/hadoop dfs -ls /user/Andrey/file1
-rw-rw----   1 Andrey supergroup  0 2011-06-23 11:20 /user/Andrey/file1
```

Изменение владельца и группы

- Исходное состояние

```
$ bin/hadoop dfs -ls /user/Andrey/file1
-rw-rw----   1 Andrey supergroup          0 2011-06-23 11:20 /user/Andrey/file1
```

- Изменение владельца

```
$ bin/hadoop dfs -chown anton /user/Andrey/file1

$ bin/hadoop dfs -ls /user/Andrey/file1
-rw-rw----   1 anton supergroup          0 2011-06-23 11:20 /user/Andrey/file1
```

- Изменение группы

```
$ bin/hadoop dfs -chgrp project1 /user/Andrey/file1
$ bin/hadoop dfs -ls /user/Andrey/file1
-rw-rw----   1 anton project1          0 2011-06-23 11:20 /user/Andrey/file1
```

Работа с HDFS из Java

```
// Настройка путей
Configuration conf = new Configuration();
FileSystem fs = FileSystem.get(conf);
Path filenamePath = new Path("hello.txt");
// Запись в файл
FSDataOutputStream out = fs.create(filenamePath);
out.writeUTF(«Hello, world");
out.close();
// Чтение файла
FSDataInputStream in = fs.open(filenamePath);
String messageIn = in.readUTF();
System.out.print(messageIn);
in.close();
```

Подключение к файловой системе

- org.apache.hadoop.fs.FileSystem – интерфейс для работы с DFS и другими файловыми системами
- org.apache.hadoop.conf.Configuration – конфигурация Hadoop и HDFS
- Подключение к файловой системе:

```
Configuration conf = new Configuration();
FileSystem fs = FileSystem.get(conf);
```

Структура имени файла

- Файл в HDFS:
  - `hdfs://namenode:port/path/file`
  - `hdfs://localhost:9000/user/hadoop/file1`
  - `hdfs://namenode:port` – можно не указывать, тогда используется namenode из текущего конфигурационного файла Hadoop
- Файл на локальном диске:
  - `file://path/file`

Запуск программы

- FileSystem может работать как с HDFS, так и с локальной файловой системой, в зависимости от способа запуска
- Локальный запуск:
  - `java HDFSHelloWorld`
- Запуск через Hadoop (запись в HDFS):
  - `$bin/hadoop HDFSHelloWorld`

Методы FileSystem

Метод	Назначение
`copyFromLocalFile`	Копирование из локальной файловой системы в HDFS
`copyToLocalFile`	Копирование из HDFS в локальную файловую систему
`create`	Создание файла
`mkdirs`	Создание каталога
`delete`	Удаление файла или каталога
`rename`	Переименование файла
`setOwner`	Установка владельца и группы файла
`setPermissions`	Установка прав доступа к файлу
`getFileBlockLocations`	Возвращает список серверов, хранящих блоки файла

Дополнительные материалы

- The Google File System
  - http://labs.google.com/papers/gfs.html
- HDFS Architecture Guide
  - http://hadoop.apache.org/common/docs/current/hdfs_design.html
- HDFS Permissions Guide
  - http://hadoop.apache.org/common/docs/current/hdfs_permissions_guide.html
- HDFS Users Guide
  - http://hadoop.apache.org/common/docs/current/hdfs_user_guide.html

Вопросы?