Hadoop

Лекция 8.

Основы администрирования Hadoop

План

- Установка Hadoop
- Администрирование MapReduce
- Администрирование HDFS

Установка Hadoop

- Операционные системы:
  - Linux (продуктивные системы и разработка)
  - Windows (только для разработки)
  - Unix (официально не поддерживается)
- Необходимое ПО:
  - Java
  - SSH клиент и сервер
  - Cygwin (для Windows)

Дистрибутивы Hadoop

- Дистрибутивы:
  - Apache http://hadoop.apache.org/
  - Yahoo! http://developer.yahoo.com/hadoop/distribution/
  - Cloudera http://www.cloudera.com/hadoop/
- В курсе используется дистрибутив Apache
- Дистрибутивы Yahoo! и Cloudera включают патчи и дополнительные компоненты (Pig, HBase, безопасность и т.п.)

Режимы работы Hadoop

- Локальный
  - Все задачи выполняются на одной машине, данные хранятся в локальной файловой системе
- Псевдо-распределенный
  - Все задачи выполняются на одной машине, но используется HDFS и JobTracker
- Распределенный
  - Работа на кластере из нескольких машин, используется HDFS и JobTracker

Локальная установка

- Установить Java 1.6 (желательно от Sun)
- Скачать дистрибутив Hadoop
- Распаковать дистрибутив
- В конфигурационном файле conf/hadoop-env.sh прописать путь к Java в переменную JAVA_HOME
- Проверить работоспособность:
  - `$ bin/hadoop`

Karmasphere Studio

- Альтернативный вариант локальной установки для рабочего места разработчика:
  - Установка Karmasphere Studio
- Karmasphere Studio включает преконфигурированные дистрибутивы Hadoop разных версий
- Установка Karmasphere Studio – см. лекцию 4.
- Для Windows нужен cygwin

Псевдо-распределенный режим

- Особенности локального режима:
  - Задачи Hadoop запускаются в рамках одного процесса Java
  - Данные берутся и записываются в локальную файловую систему
- Псевдо-распределенный режим:
  - Для запуска задач используется JobTracker и TaskTracker, для Map и Reduce созлаются отдельные процессы
  - Доступна HDFS

Установка в псевдо-распределенном режиме

- Выполнить все действия локальной установки
- Настроить доступ на локальную машину по SSH без пароля:

```
$ ssh-keygen -t dsa -P '' -f ~/.ssh/id_dsa
$ cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys
$ ssh localhost
```

- Отредактировать конфигурационные файлы Hadoop

Конфигурационные файлы

- Дистрибутив Hadoop состоит из трех компонентов:
  - Core (или Common) – общие компоненты
  - HDFS
  - MapReduce
- Конфигурационных файлов тоже три, для каждого компонента:
  - `conf/core-site.xml`
  - `conf/hdfs-site.xml`
  - `conf/mapred-site.xml`

Конфигурационные файлы

- В конфигурационных файлах прописываются только параметры, специфичные для данной установки
- Значения по умолчанию содержатся в файлах:
  - `src/core/core-default.xml`
  - `src/core/hdfs-default.xml`
  - `src/core/mapred-default.xml`

Формат конфигурационных файлов

- Конфигурационный файл включает свойства, состоящие из имени и значения
- Используется формат xml
- Пример:

```
<configuration>
    <property>
        <name>fs.default.name</name>
        <value>hdfs://localhost:9000</value>
    </property>
</configuration>
```

Конфигурационные файлы для псевдо-распределенного режима

Имя свойства	Значение свойства
conf/core-site.xml	
`fs.default.name`	`hdfs://localhost:9000`
conf/hdfs-site.xml	
`dfs.replication`	`1`
conf/mapred-site.xml	
`mapred.job.tracker`	`localhost:9001`

Назначение свойств

- fs.default.name
  - Адрес узла имен файловой системы HDFS по умолчанию
- dfs.replication
  - Фактор репликации, количество копий блоков в HDFS
- mapred.job.tracker
  - Адрес узла запуска задач

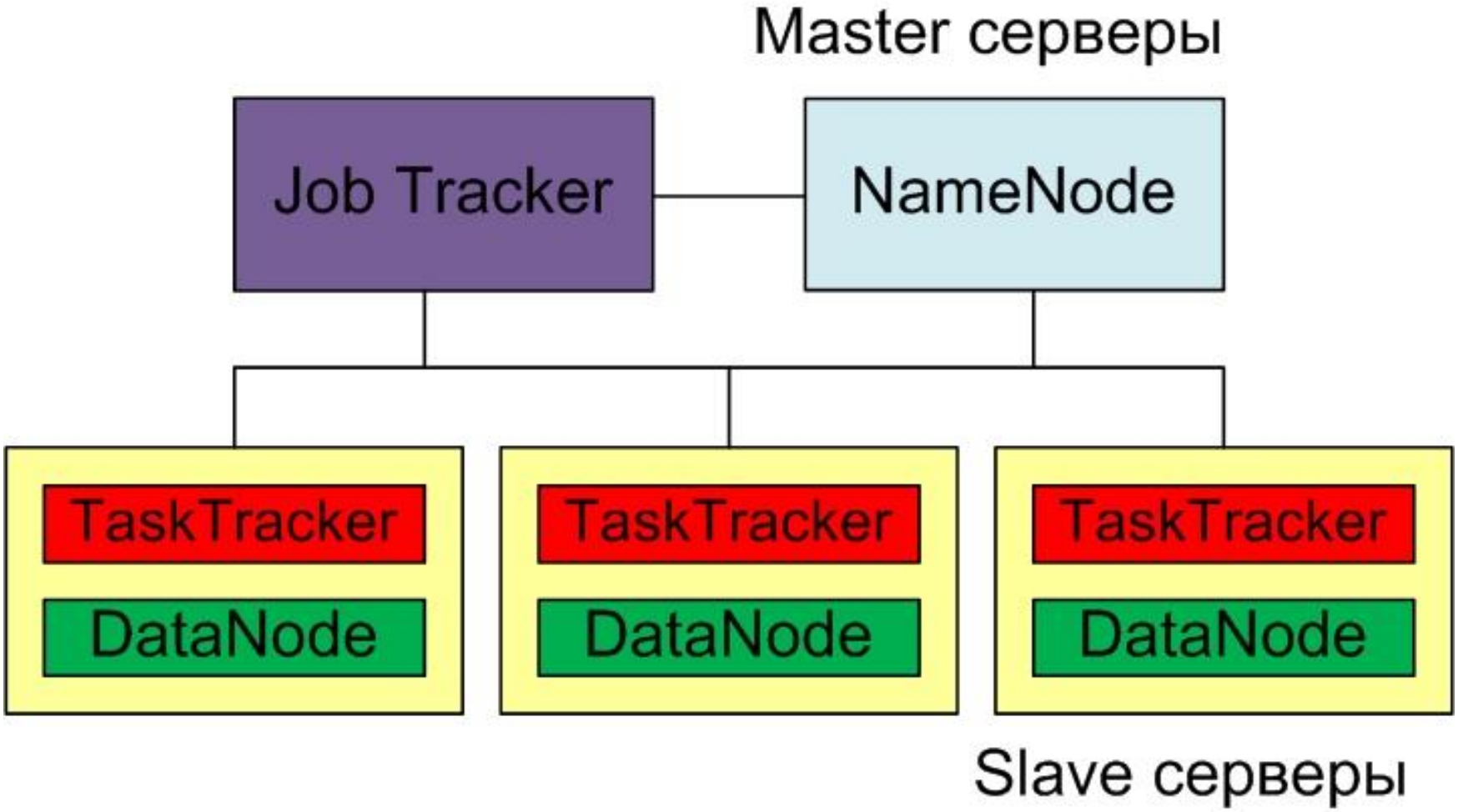
Установка в псевдо-распределенном режиме

- Форматирование HDFS:
  - `bin/hadoop namenode -format`
- Запуск Hadoop:
  - `bin/start-all.sh`
- Проверка работоспособности через Web-интерфейс:
  - HDFS: http://localhost:50070/
  - JobTracker: http://localhost:50030/

Установка кластера

- Небольшой кластер, до 10 узлов
- Более крупным кластерам нужна дополнительная настройка

Кластер Hadoop



Кластер Hadoop

- Узлы кластера Hadoop:
  - Сервер имен (NameNode), Master
  - Сервер запуска задач (JobTracker), Master
  - Рабочие серверы (DataNode и TaskTracker), slave
- В небольших кластерах допускается совмещать сервера имен и запуска задач

Установка кластера

- На всех узлах:
  - Синхронизировать время, например по NTP
  - Установить Java
  - Скачать и распаковать дистрибутив Hadoop, прописать путь к Java
  - Настроить доступ без пароля по SSH с каждого узла на каждый
- В файле conf/slaves прописать имена всех slave серверов, по одному в строке
  - В conf/masters ничего писать не нужно

Запуск команд на узлах

- Часто требуется запустить одинаковые команды на всех узлах кластера
- Hadoop для этой цели включает специальный скрипт:
  - `$ bin/slaves.sh command`
- Скрипт выполнит команду command на всех узлах кластера, указанных в файле conf/slaves

Создание каталогов для HDFS

- HDFS по умолчанию хранит данные во временном каталоге (/tmp в Linux)
  - Данные могут быть потеряны
  - Недопустимо для продуктивных кластеров!!!
- Необходимо создать каталоги:
  - Для хранения метаданных на узле имен
    - `$ mkdir /home/hadoop/dfs/name`
  - Для хранения данных на slave узлах
    - `$ bin/slaves.sh mkdir /home/hadoop/dfs/data`

Конфигурационные файлы

Имя свойства	Значение свойства
conf/core-site.xml	
fs.default.name	hdfs://dfs-mater:9000
hadoop.tmp.dir	/tmp/hadoop
conf/hdfs-site.xml	
dfs.replication	2
dfs.name.dir	/home/hadoop/dfs/name
dfs.data.dir	/home/hadoop/dfs/data
conf/mapred-site.xml	
mapred.job.tracker	jobtracker-master:9001
mapred.system.dir	/hadoop/mapred/system

Назначение свойств

- hadoop.tmp.dir
  - Адрес временного каталога Hadoop
- dfs.name.dir
  - Каталог для хранения метаданных HDFS
- dfs.data.dir
  - Каталог для хранения данных HDFS
- mapred.system.dir
  - Системный каталог MapReduce, должен находиться в HDFS

Установка кластера

- Заполненные конфигурационные файлы необходимо скопировать на все серверы кластера
- Форматирование HDFS:
  - `$ bin/hadoop namenode -format`
- Запуск кластера:
  - `$ bin/start-all.sh`
- Проверка работоспособности через Web:
  - http://dfs-master:50070
  - http://jobtracker-master:50030

Запуск и остановка кластера

- Запуск кластера Hadoop:
  - `$ bin/start-all.sh`
  - Запускается NameNode, JobTracker и на каждом узле в файле conf/slaves DataNode и TaskTracker
- Запуск отдельных компонентов Hadoop:
  - `$ bin/start-dfs.sh`
  - `$ bin/start-mapred.sh`
- Остановка Hadoop:
  - `$ bin/stop-all.sh`
  - `$ bin/stop-dfs.sh`
  - `$ bin/stop-mapred.sh`

Журнальные файлы

- Hadoop записывает журналы в каталог logs
- Журналы ведутся отдельно для NameNode, JobTracker, DataNode и TaskTracker
- Два типа журналов:
  - *.log – журнал операций
  - *.out – файл стандартного вывода

Администрирование HDFS

- Просмотр статуса
- Проверка целостности файловой системы
- Управление репликацией
- Балансировка
- RackAwareness

Средства администрирования HDFS

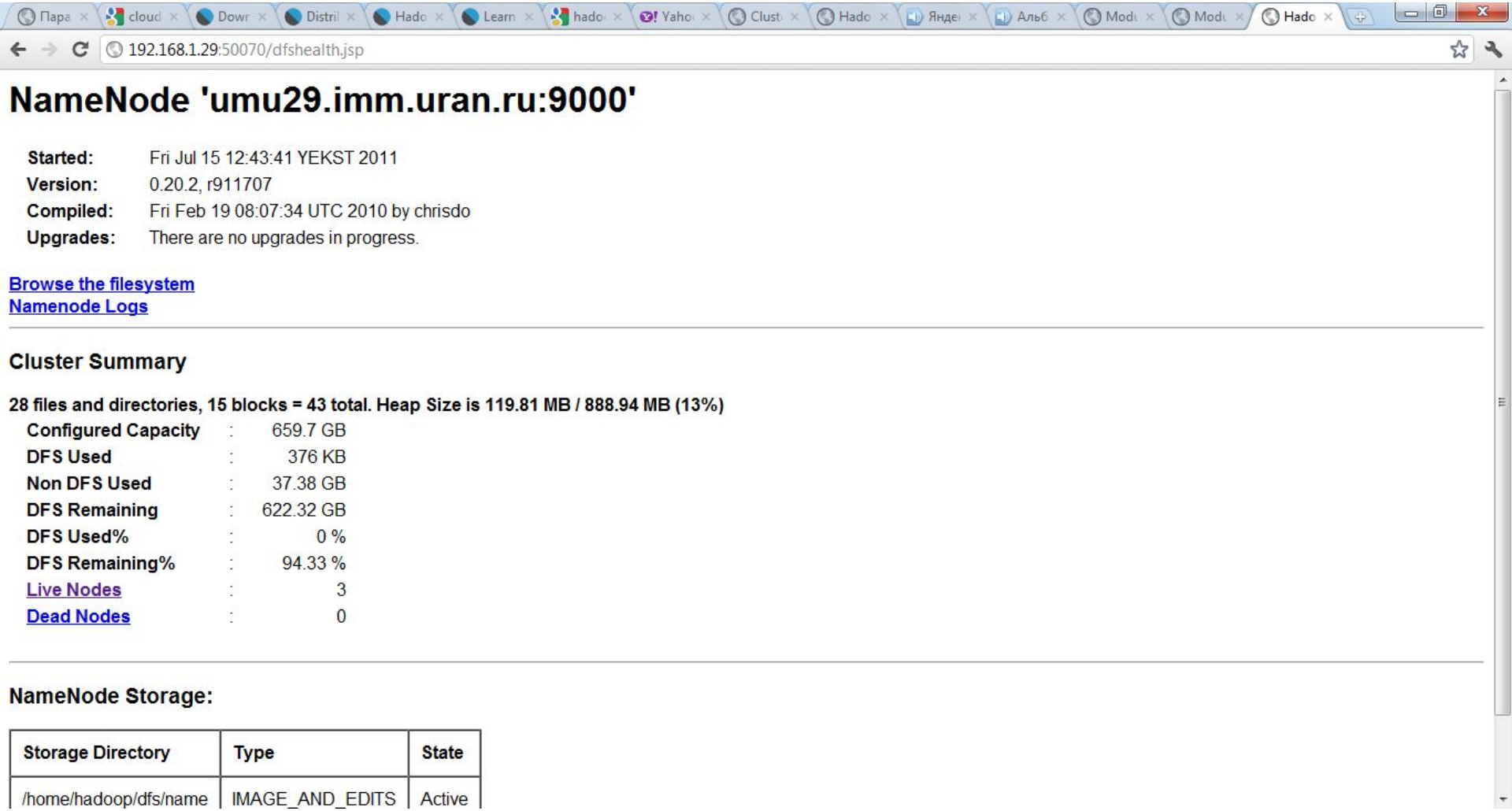
- Командная строка:
  - `$ bin/hadoop dfsadmin`
- Web:
  - http://dfs-master:50070

Состояние HDFS

```
$ bin/hadoop dfsadmin -report
Configured Capacity: 708349218816 (659.7 GB)
Present Capacity: 668208627712 (622.32 GB)
DFS Remaining: 668208242688 (622.32 GB)
DFS Used: 385024 (376 KB)
DFS Used%: 0%
Under replicated blocks: 0
Blocks with corrupt replicas: 0
Missing blocks: 0

-------------------------------------------------
Datanodes available: 3 (3 total, 0 dead)
```

Состояние HDFS



Проверка целостности HDFS

- Целостность файловой системы:
  - Файлы не повреждены
  - Блоки не потеряны
  - Присутствует необходимое количество копий блоков
- Причины проблем:
  - Отказ серверов
  - Некорректное выключение кластера
  - Ошибки в ПО
- Проверка целостности:
  - `$ bin/hadoop fsck`

Управление репликацией

- Репликация – создание нескольких копий блоков на разных машинах
- Фактор репликации – количество копий блока. Рекомендации:
  - 1 – для одного сервера
  - 2 – для кластера из 2-4 серверов
  - 3 – для кластера более 4 серверов
- Фактор репликации задается в файле conf/hdfs-site.xml, параметр `dfs.replication`

Балансировка

- Балансировка – равномерное распределение блоков данных по серверам
- Причины нарушения баланса:
  - Добавление или удаление узлов
  - Изменение фактора репликации
- Запуск балансировщика:
  - `$ bin/start-balancer.sh`
- Остановка балансировщика:
  - `$ bin/stop-balancer.sh`

RackAwareness

- RackAwareness – способность HDFS «понимать», в каком «шкафу» находятся серверы кластера и создавать копии блока в разных шкафах
- Имена «шкафов» в Hadoop иерархические и похожи на пути в файловой системе:
  - /default-rack («шкаф» по умолчанию)
  - /switch1/rack21 («шкаф» 21, подключенный к коммутатору switch1)

RackAwareness

- Имя «шкафа» Hadoop определяет по IP-адресу сервера
- Для определения имени «шкафа» Hadoop вызывает внешний скрипт:
  - Имя скрипта указывается в файле conf/core-site.xml, параметр `topology.script.file.name`
  - Скрипт получает IP-адрес сервера в качетсве аргумента
  - Скрипт должен вернуть имя «шкафа»

Итоги

- Установка Hadoop:
  - Локальный режим
  - Псевдо-распределенный режим
  - Кластер
- Администрирование Hadoop
- Администрирование HDFS

Дополнительные материалы

- Hadoop Single Node Setup
  - http://hadoop.apache.org/common/docs/stable/single_node_setup.html
- Hadoop Cluster Setup
  - http://hadoop.apache.org/common/docs/stable/cluster_setup.html
- Hadoop Commands Guide
  - http://hadoop.apache.org/common/docs/stable/commands_manual.html
- HDFS Users Guide
  - http://hadoop.apache.org/common/docs/stable/hdfs_user_guide.html

Вопросы?