Интеллектуальные информационные системы

HDFS

Кафедра управления и интеллектуальных технологий НИУ «МЭИ»

2023 г.

HDFS

HDFS – Hadoop Distributed File System – распределенная файловая система Hadoop.
Основана на идее GFS – Google File System – проприетарный продукт Google.

- Для пользователя – как один большой жесткий диск
- Работает поверх обычных файловых систем
- HDFS – приложение пользовательского уровня

- Данные не теряются, если выходит из строя диск или сервер
- Есть механизм восстановления данных, если часть данных теряется

По мотивам Hadoop. Система для обработки больших объемов данных

HDFS подходит для:

Хранение больших файлов

- Терабайты, петабайты
- Миллионы (но не миллиарды) файлов
- Файлы размером от 100 Мб. Можно хранить большие файлы, но желательно не хранить маленькие файлы (в отличие от обычной файловой системы).

Стриминг данных

- Паттерн «Write once / read many times»
- Оптимизация под последовательное чтение (не произвольное) больших файлов с большой скоростью
  - Большой файл бьется на блоки, блоки хранятся последовательно -> Чтение проходит быстро
- Можем дописывать файл (append)

HDFS НЕ подходит для:

- **Low latency reads – хотим быстро получать часть данных (веб-сервис)**
  HDFS Больше заточен под большие офлайн расчеты
  Hbase отчасти помогает решить такую задачу

- **Большое количество маленьких файлов**
  Лучше 1000 файлов по 1 Гб, чем 100 000 по 10 Мб

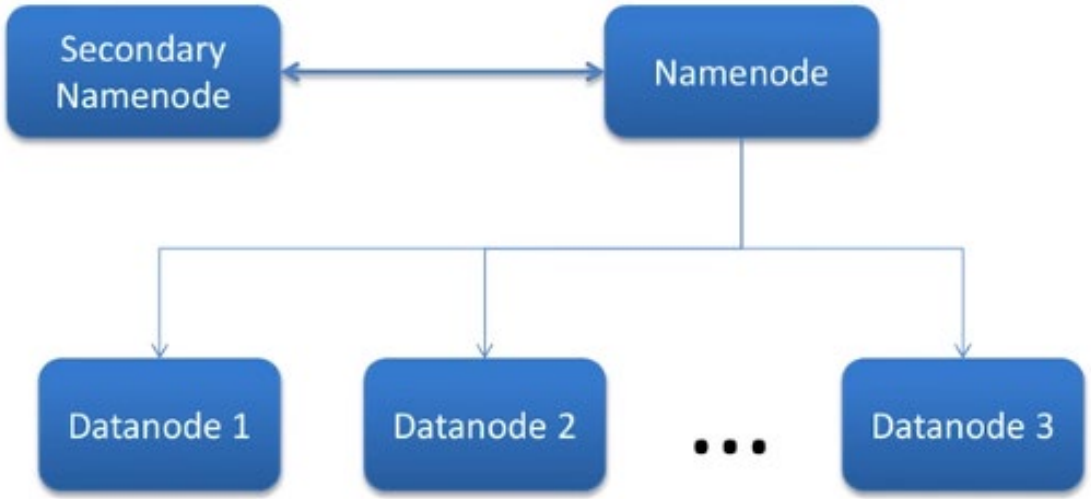
- **Многопоточная запись в файл**
  Один процесс записи на один файл

- **Нет поддержки записи по смещению**
  Данные дописываются в конец файла

Демоны HDFS

HDFS – распределенное приложение, запущенное не всех машинах кластера.
Есть несколько типов процессов – демонов (daemon) – которые постоянно запущены в ОС:

- NameNode
- DataNode
- Secondary Node



NameNode

Запускается на одной отдельной надежной машине
Количество файлов в FS ограничено памятью NameNode (NN).

Отвечает за:

- Файловое пространство (namespace)
- Метаинформацию
- Расположение блоков файлов

Namenode знает все о структуре файловой системы: иерархию файлов и директорий, как файл разбит на блоки, где эти блоки и их реплики находятся (на каких серверах). При этом Namenode знает сколько всего свободного места в кластере и на каждом сервере.

Namenode хранит все в оперативной памяти для получения быстрого доступа на чтение и изменение файловой структуры.
Данные обо всех изменениях хранятся в логе => нужен большой жесткий диск
Мощный CPU для обработки heartbeats от DataNode-ов и запросов от клиентов

Secondary NameNode

Если NameNode выходит из строя, что HDFS работать не будет (данные останутся на серверах, но не известно где и какие).

Информация о файловой системе – в файле fsimage
Все изменения - в лог Write Ahead Log
Если NameNode падает, то текущее состояние образа HDFS можно восстановить с помощью старого fsimage и этих логов изменений – получается новый образ fsimage, который NameNode грузит в память.
Процесс восстановления долгий и выполняется на Secondary NameNode (SNN).
С некоторой периодичностью SNN забирает fsimage, накатывает лог обновлений –> получаем новый образ, который может использоваться NN в случае сбоя

Для SNN требуется то же железо, что и NN, но по факту не используется как backup NN

DataNode

- Запущен на каждой машине кластера
- Отвечает за хранение информации на машине.
- Отправляет сигналы о состоянии сервера (heartbeat)

Файлы и блоки

Файл делится на части (chunks), которые записываются в блоки
Блок – единица хранения данных

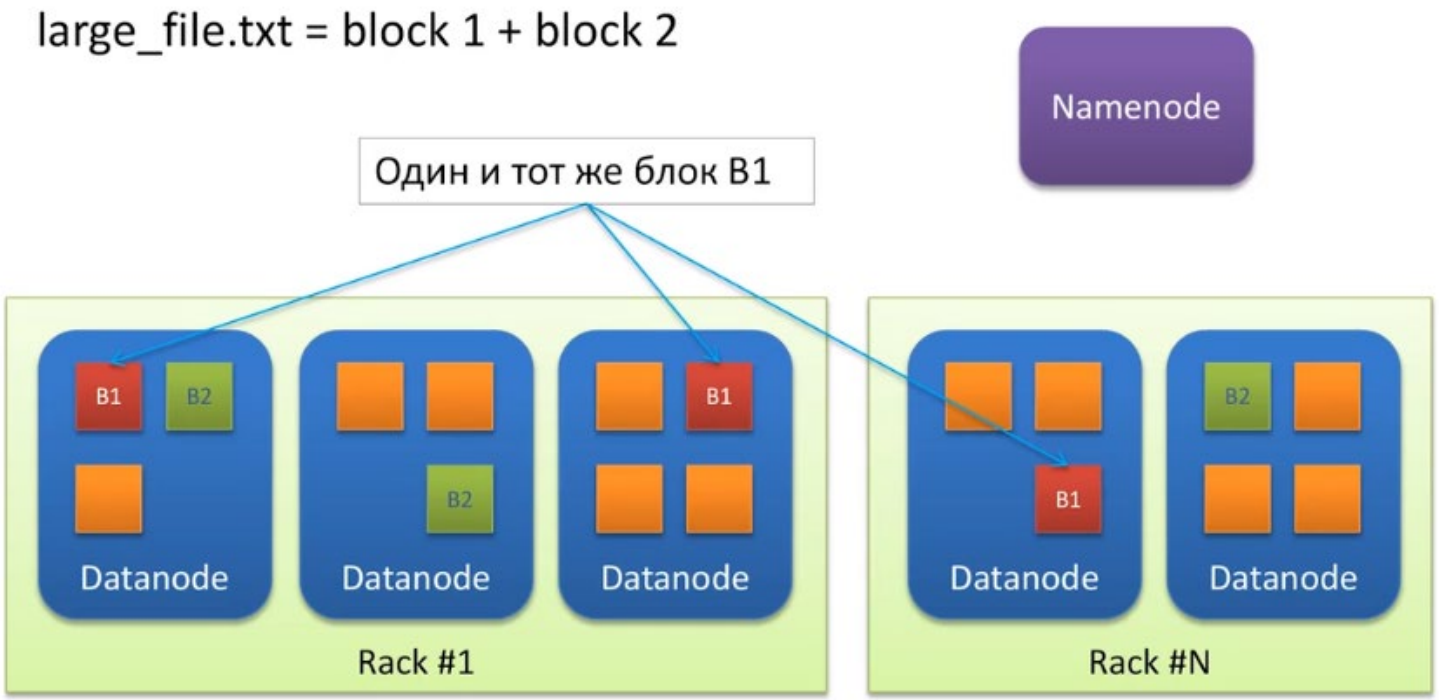
Управляется через NameNode
Хранится на DataNode

DataNode не знает, частью какого файла является данный блок

Клиент, чтобы прочитать файл, обращается к NameNode, узнает из каких блоков состоит файл и где они находятся и общается с DataNode чтобы прочитать данные.

Репликация данных

- Реплицируется по машинам в процессе записи
- Один и тот же блок хранится на нескольких DataNode
  обычно на 3 – фактор репликации
- Это нужно для надежности и упрощения доступа разными процессами



Блок данных в HDFS

Размер блока в обычной файловой системе обычно 2 - 8Кб
Размер блока в HDFS: 64 – 256Мб

Seek time – время на перемещение головки на нужную позицию диска
Transfer rate – скорость чтения и передачи клиенту.

Transfer Time > Seek Time

Основной мотив – снизить seek time по сравнению с transfer time

Seek time = 10ms
Transfer rate = 100 Mb/s
Каким должен быть размер блока, чтобы seek time составил 1% от transfer rate?

Блок данных в HDFS

Сколько блоков в HDFS будет иметь файл размером 1025 Мб?
Характеристики HDFS следующие:
Размер блока: 128 Мб
Фактор репликации: 3

Пусть есть файл размером 200Тб = 209.715.200 Мб

- При размере блока 64Мб:
  209.715.200 / 64 = 3.276.800 блоков
- При размере блока 128Мб:
  209.715.200 / 128 = 1.638.400 блоков

} Домножить на фактор репликации.

Репликация блоков

Об упавших серверах знает NameNode, поэтому она знает, для каких файлов реплик стало не хватать. Для них NN посылает команду на создание новых реплик на оставшихся доступных серверах.

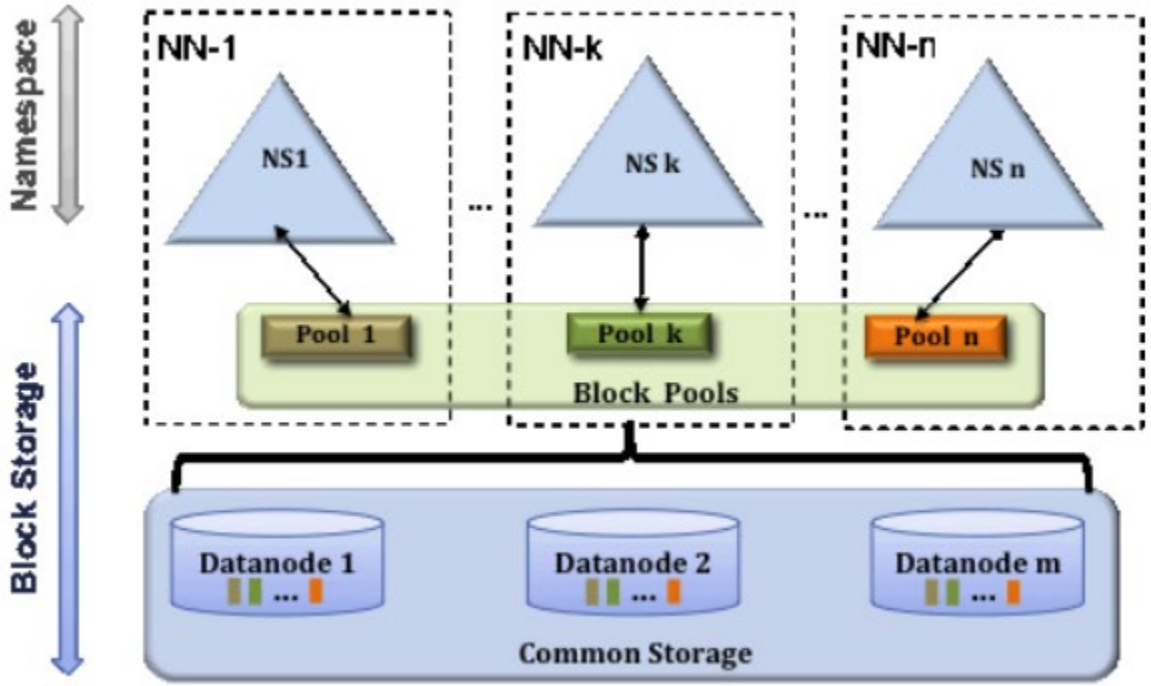
Datanode посылает сигнал (heartbeat) Namenode каждые 3 секунды. Если от Datanode нет сигнала в течении 10 минут, то DataNode считается недоступной, в все хранящиеся на ней блоки - недоступными. Нехватающие блоки NameNode автоматически дореплицирует

Механизм федерации NameNode

Для масштабирования системы – федерация NameNode. NameNode’ы объединены, являются независимыми и не требуют координации друг с другом.

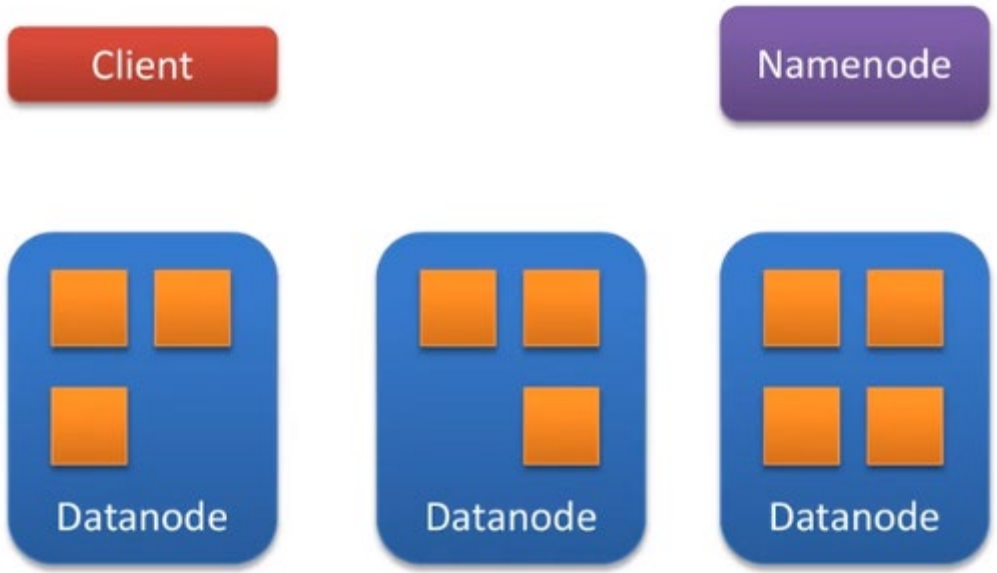
DataNode’ы используются в качестве общего хранилища для блоков всеми NameNode’ами
Пул блоков - это набор блоков, принадлежащих к одному NameNode

DataNode’ы хранят блоки разных NameNode’ов



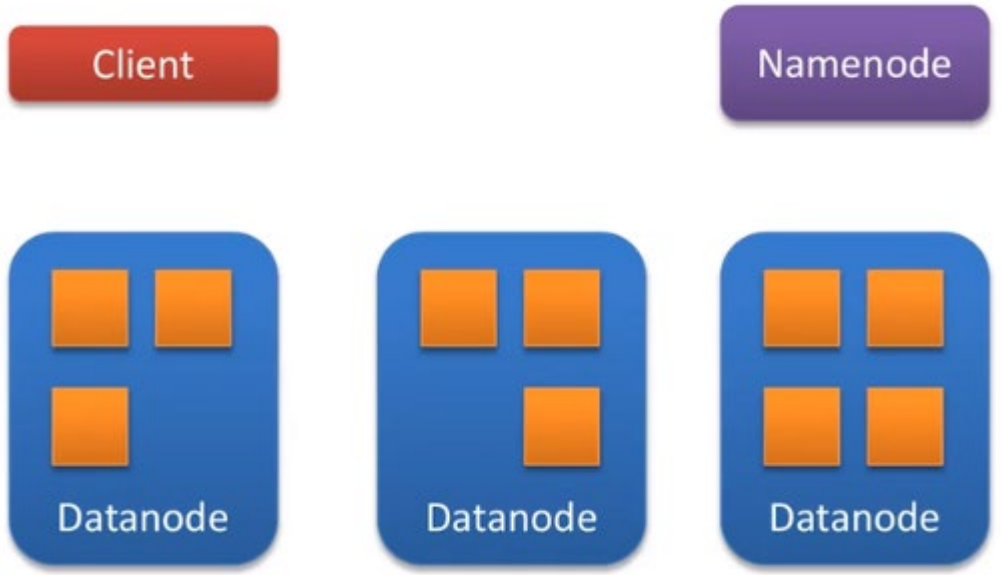
Чтение данных

Клиент запрашивает информацию о файле у NameNode, получает информацию о том, где взять данные.
Клиент запрашивает данные у DataNode



Запись данных

- Клиент делает запрос к Namenode на создание блока.
- Namenode определяет, на каких хостах должны быть расположены реплики и передает эту информацию клиенту.
- Клиент начинает писать данные блока на первую ноду из списка и сообщает Datanode, на какой следующий хост нужно реплицировать данные.
- Запись происходит небольшими порциями (по 4Кб).
- После получения каждой порции данных, Datanode передает ее следующей Datanode из списка.
- Вторая Datanode также получает данные порциями и передает их следующей Datanode.

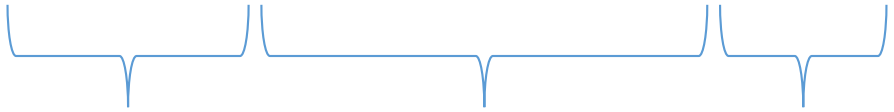


Shell команды HDFS

$hdfs dfs -<command> -<option> <URI>

URI

hdfs://localhost:8020/user **<=>** **/user** (см. fs.default.name)



file:///path/to/file - путь к файлу в локальной файловой системе

$hdfs dfs -ls / - листинг корневой директории

Shell команды HDFS

Стандартные команды Linux:

- **-cat, -text, -tail**
- **-ls**
- **-chmod**
- **-rm**
- **-cp (для маленьких файлов), distcp (для больших файлов)**
- **-mv**

Специальные:

- **-setrep**
- **-copyFormLocal, -copyToLocal**
- **…**

- **$hdfs dfs -help <command>**