ITAB
Academy

Présentation de

BIG DATA

ITAB ACADEMY

HDFS

- HDFS
- Common HDFS commands
- HDFS Lab 1
- Browsing HDFS DATA

ITAB ACADEMY

ITAB Academy

HDFS

Le système de fichiers distribués Hadoop (HDFS) est basé sur le système de fichiers Google (GFS).

Il fournit un système de fichiers distribué conçu pour fonctionner de manière fiable et tolérante aux pannes sur de grands groupes (des milliers d'ordinateurs).

HDFS utilise une architecture maître / esclave où le maître est constitué d'un seul NameNode qui gère les métadonnées du système de fichiers et d'un ou de plusieurs DataNodes esclaves qui stockent les données réelles.

- **HDFS**
  - The Hadoop Distributed File System (HDFS) is based on the Google File System (GFS)
  - It provides a distributed file system that is designed to run on large clusters (thousands of computers) of small computer machines in a reliable, fault-tolerant manner.
  - HDFS uses a master/slave architecture where master consists of a single NameNode that manages the file system metadata and one or more slave DataNodes that store the actual data.

ITAB Academy

hadoop HDFS

Name Node et Data Node

Le seul et unique cluster **NameNode** (démon) effectue les tâches suivantes:

- ✓ Mappe un fichier avec son ensemble de blocs
- ✓ Mappe un bloc avec les DataNodes où il réside
- ✓ Moteur de réplication pour les blocs.

Le **DataNodes** (démon) prend en charge:

- ✓ lecture et écriture dans le système de fichiers,
- ✓ création de bloc,
- ✓ effacement,
- ✓ et réplication basée sur les instructions du NameNode.

- **Name Node and Data Node**

- The Single and only 1 cluster **NameNode** (daemon) performs these tasks:
  - Maps a file name to a set of blocks
  - Maps a block to the DataNodes where it resides
  - Replication engine for blocks.

- The **DataNodes** (daemon) takes care of:
- read and write operation with the file system,
  - block creation,
  - deletion,
  - and replication based on instruction given by NameNode.

Secondary and Standby Name Nodes

Le NameNode secondaire:

- ✓ il ne peut pas remplacer le NameNode principal en cas d'échec.
- ✓ il effectue des points de contrôle périodiques. "Points de contrôle dans les jeux"
- ✓ en cas d'échec de Namenode, les administrateurs Hadoop doivent récupérer manuellement les données de Namenode secondaire.

Le NameNode de veille:

- ✓ Hadoop 1.0, NameNode est à point unique d'échec (SPOF).
- ✓ Seulement disponible dans Hadoop 2.0.
- ✓ fournit un basculement automatique en cas d'échec de Namenode actif.

ITAB Academy

- **Secondary Name Node & Standby Name node**
- The Secondary NameNode:
  - it can not replace the primary name-node in case of its failure.
  - it perform periodic checkpoints. "Checkpoints in Gaming"
  - in case of Namenode failure, Hadoop admins have to manually recover the data from Secondary Namenode.
- The Standby NameNode:
  - Hadoop 1.0, NameNode is single point of Failure (SPOF).
  - Only Available in Hadoop 2.0.
  - provides automatic failover in case Active Namenode fails.

Battements de coeur

DataNodes envoie un hearbeat au NameNode

- ✓ Une fois toutes les 3 secondes

NameNode utilise les heartbeat pour détecter les échecs des DataNodes

- **Heart Beats**
  - DataNodes send hearbeat to the NameNode
    - Once every 3 seconds
  - NameNode uses heartbeats to detect DataNode failure

Replication des Blocs

Blocs de fichiers

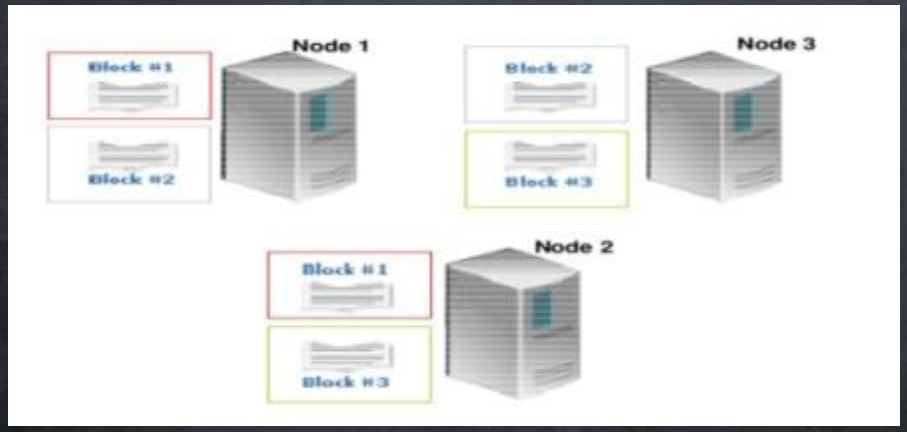
- ✓ Les données sont divisées en blocs de 64 Mo (par défaut) ou 128 Mo (recommandé).

Réplication HDFS

- ✓ Les blocs sont répliqués sur plusieurs nœuds.
- ✓ Permet une défaillance de noeud sans perte de données.

- **File Blocks**
  - Data is split into blocks of 64 MB (default for Hadoop1.x) or 128 MB (recommended, default in Hadoop 2.x).

- **HDFS Replication**
  - Blocks are replicated to multiple nodes.
  - Allows for node failure without data loss.



Placement de bloc

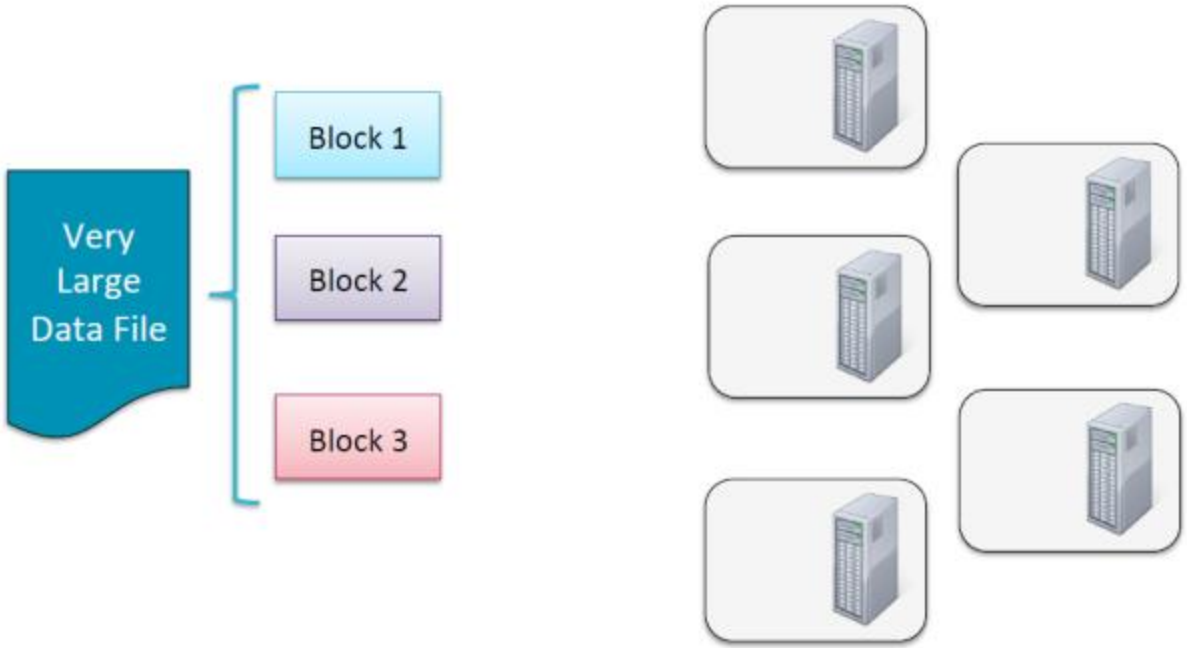
Stratégie actuelle

- ✓ Une réplique sur le nœud local
- ✓ Deuxième réplique sur un rack distant
- ✓ Troisième réplique sur le même rack distant
- ✓ Des répliques supplémentaires sont placées au hasard

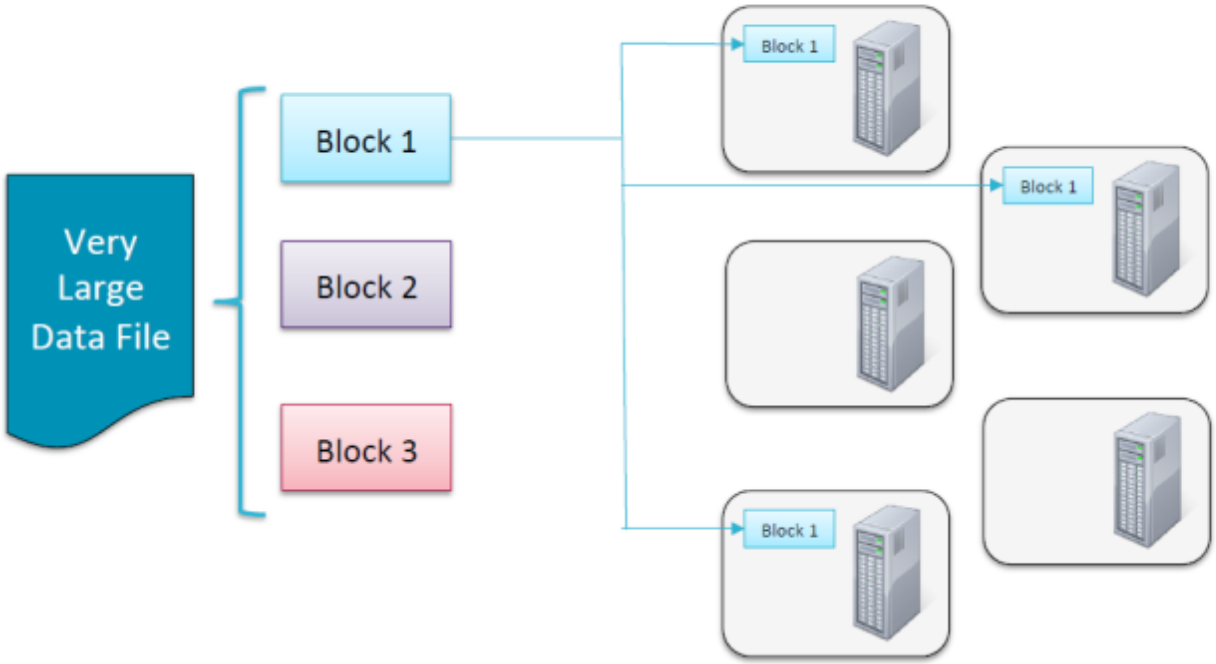
Les clients lisent à partir des répliques les plus proches

- **Block Placement**
  - Current Strategy
    - One replica on local node
    - Second replica on a remote rack
    - Third replica on same remote rack
    - Additional replicas are randomly placed
  - Clients read from nearest replicas

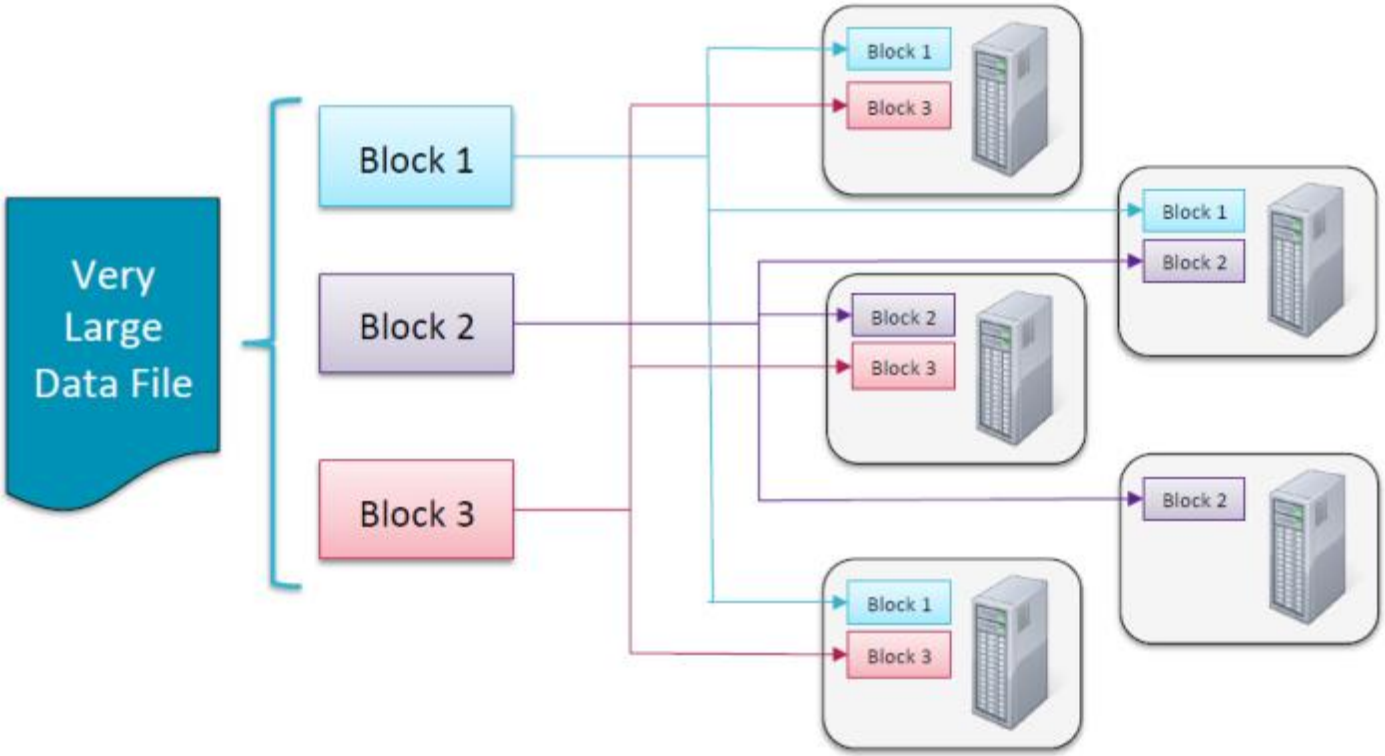
How files are stored



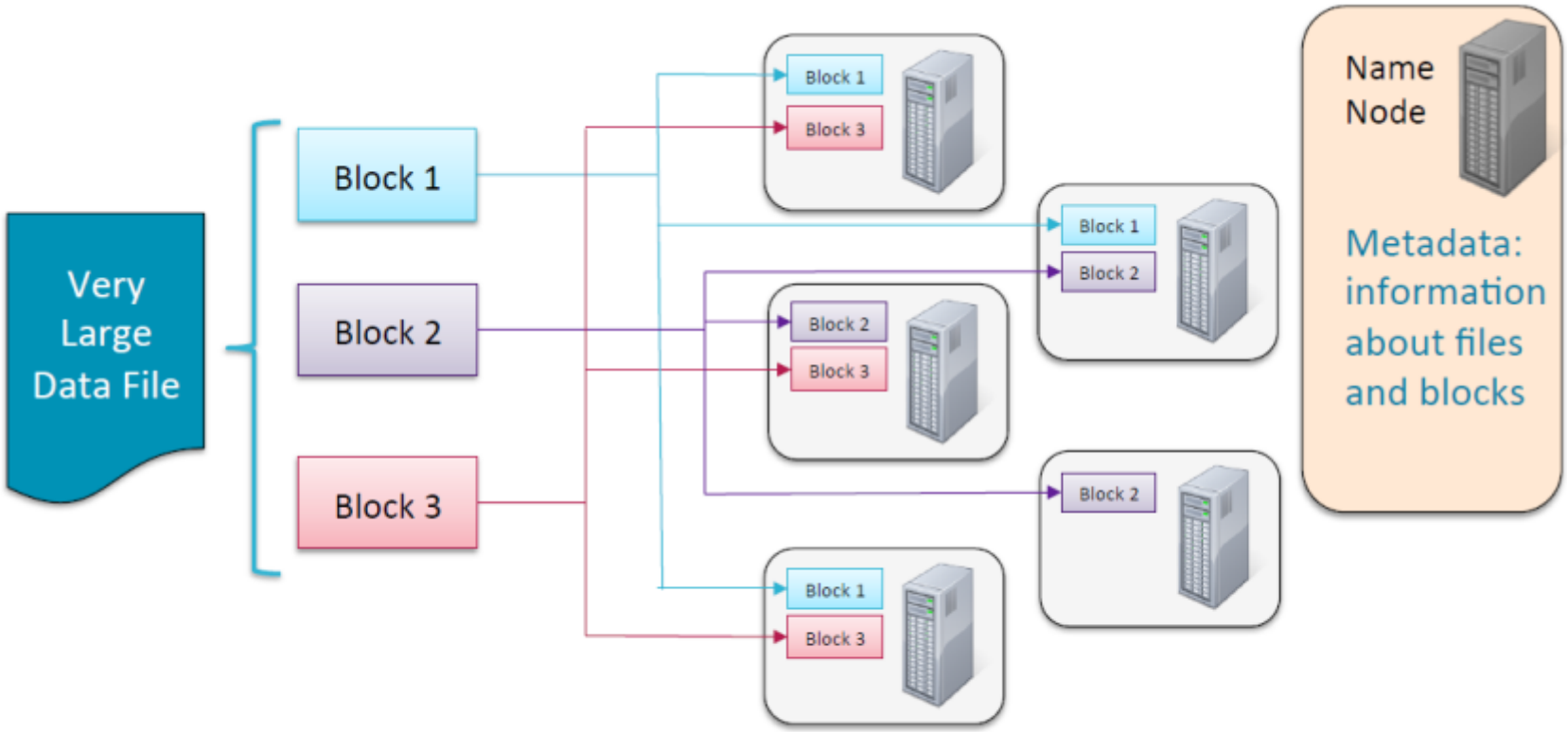
How files are stored



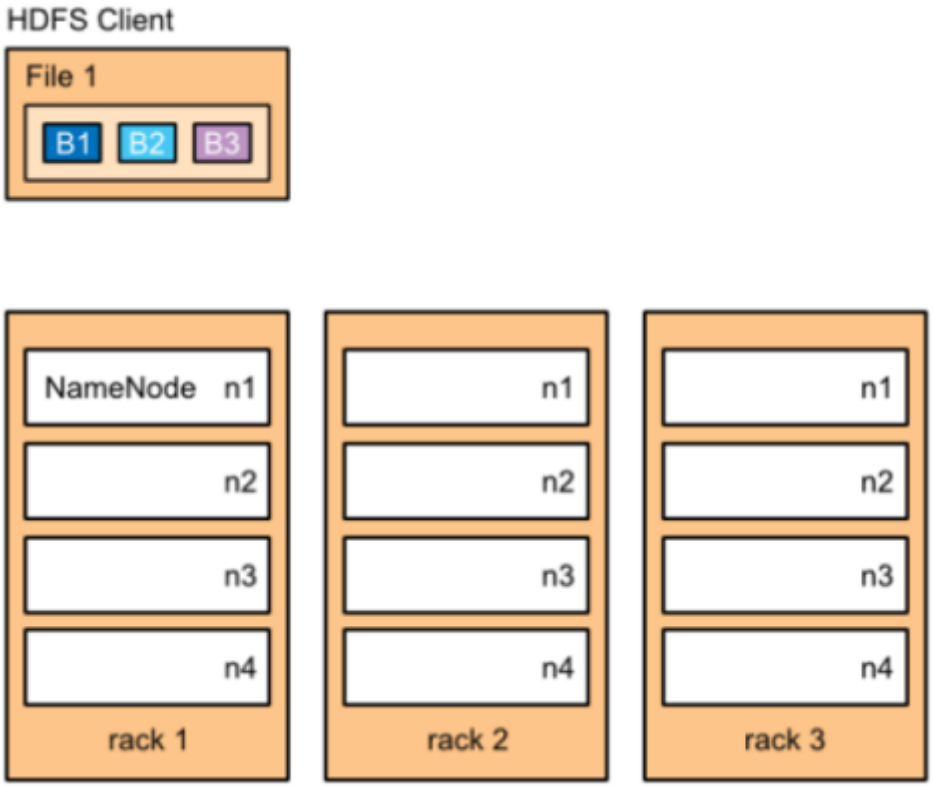
How files are stored



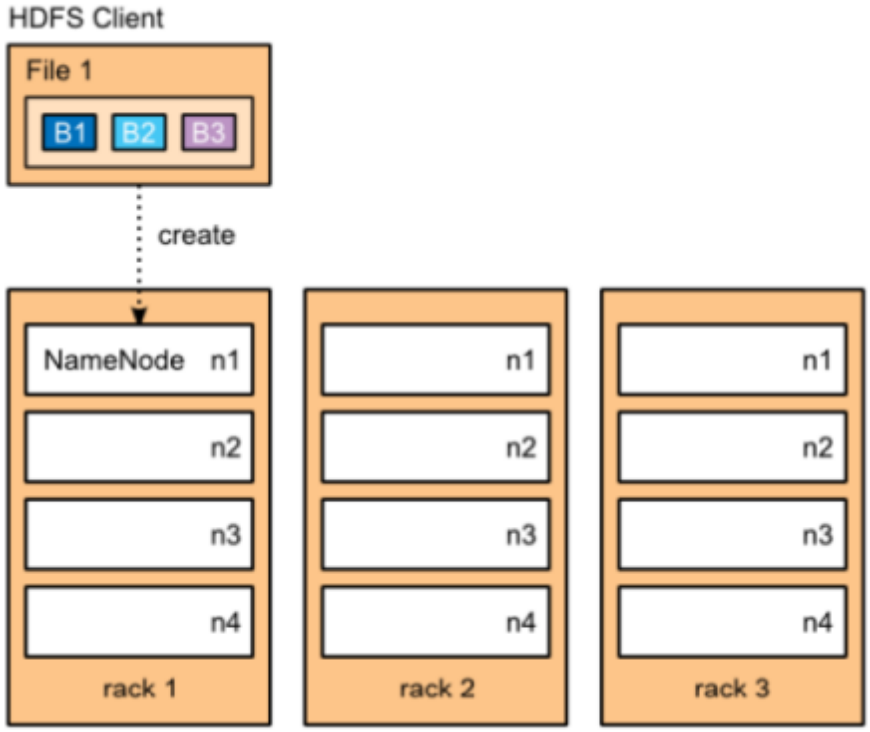
How files are stored



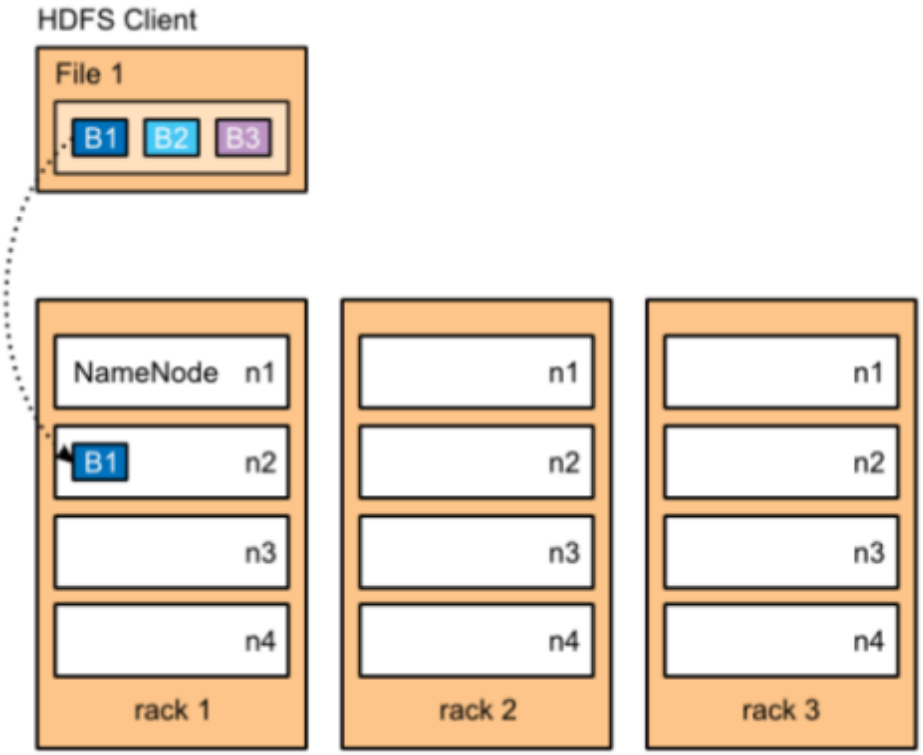
Writing a file to HDFS



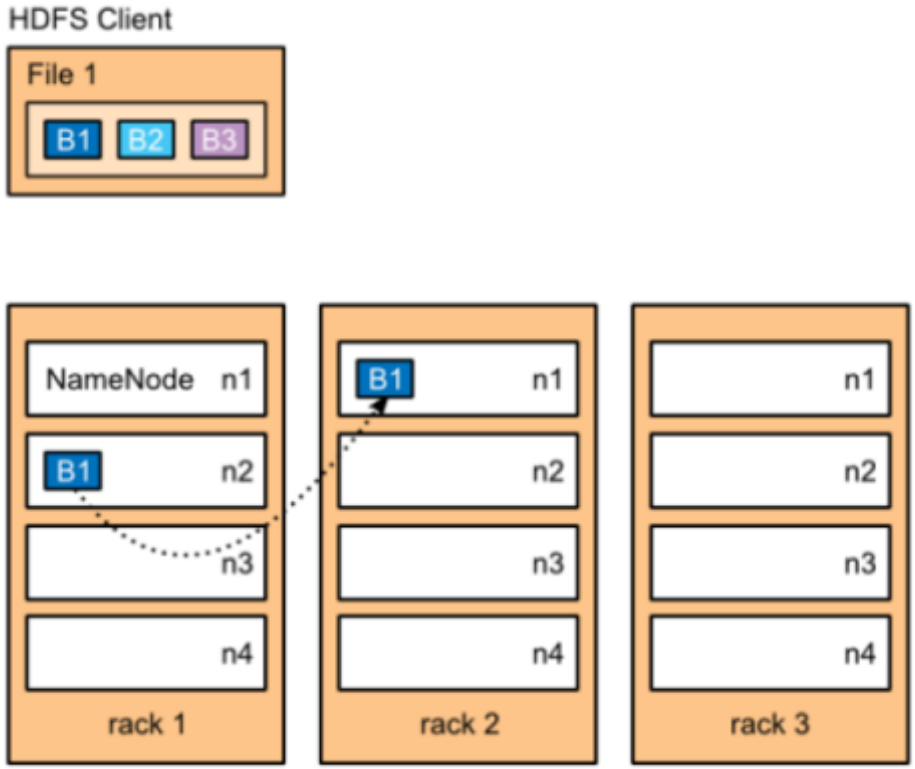
Writing a file to HDFS



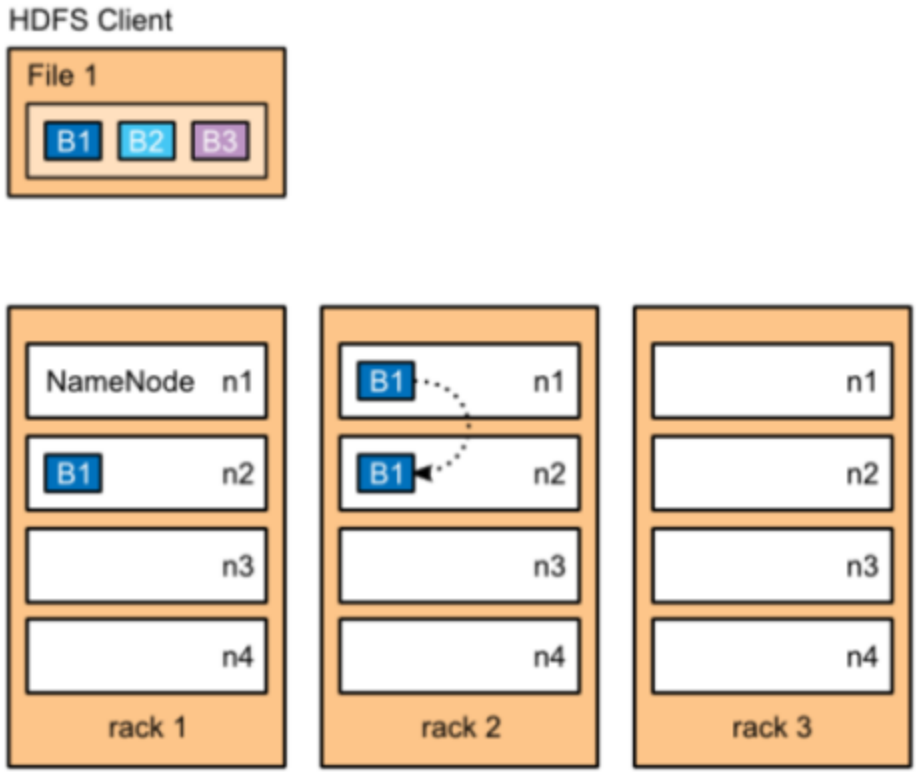
Writing a file to HDFS



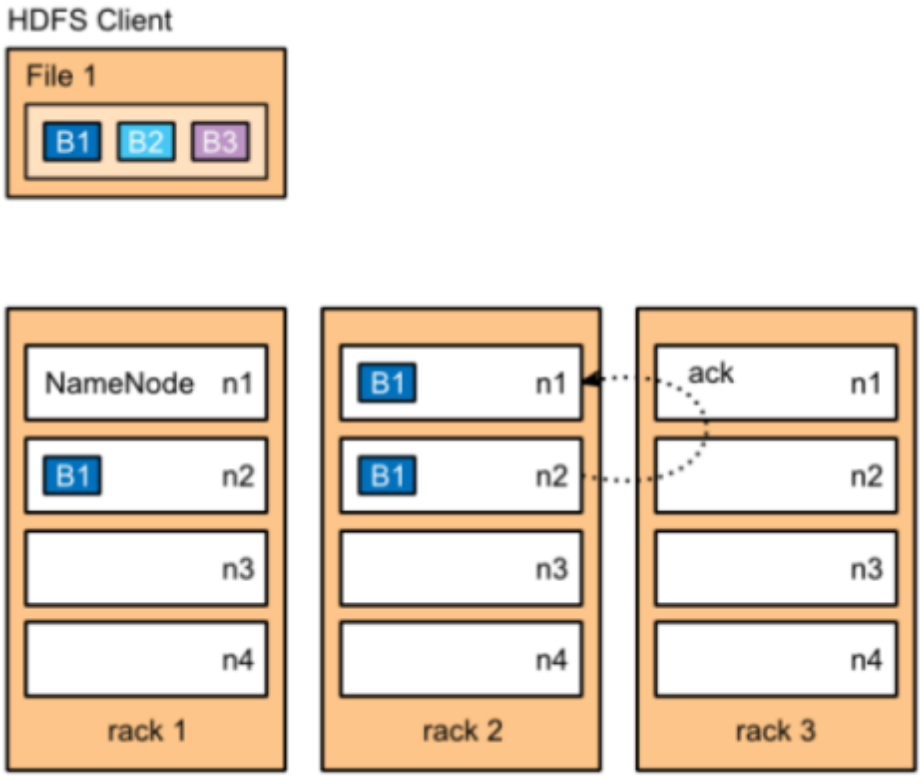
Writing a file to HDFS



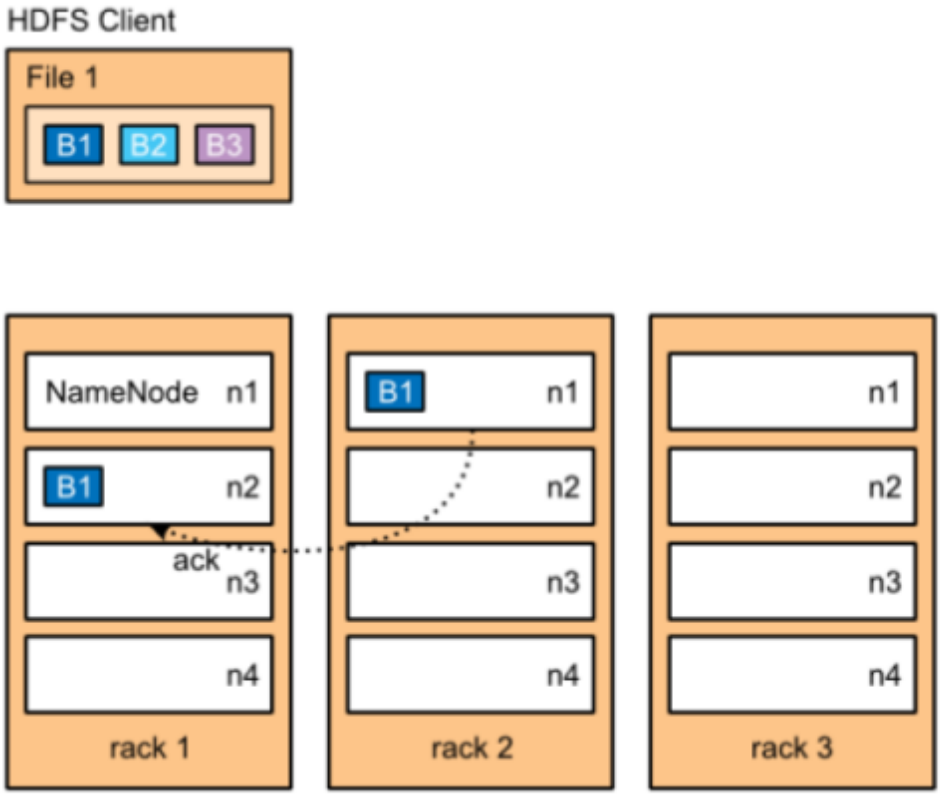
Writing a file to HDFS



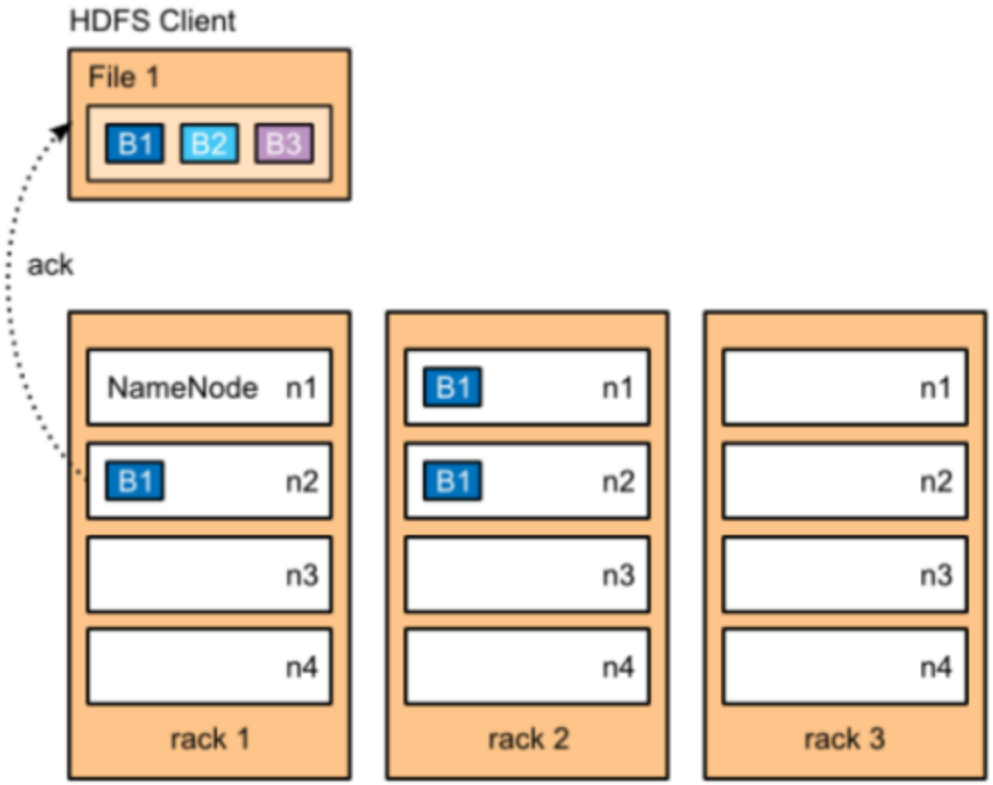
Writing a file to HDFS



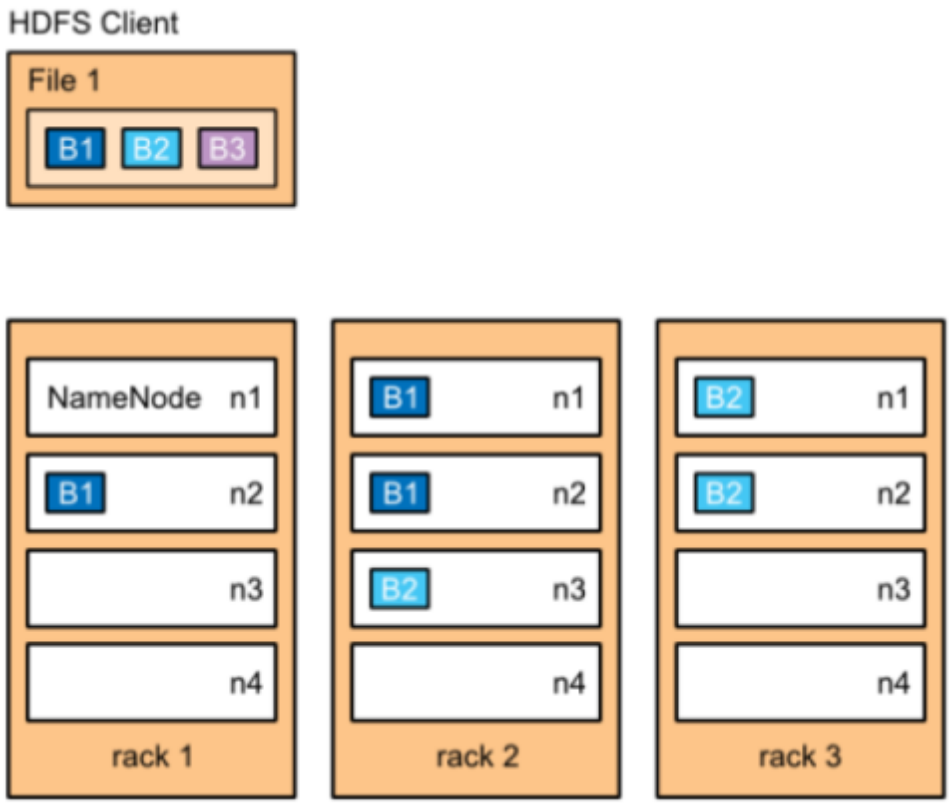
Writing a file to HDFS



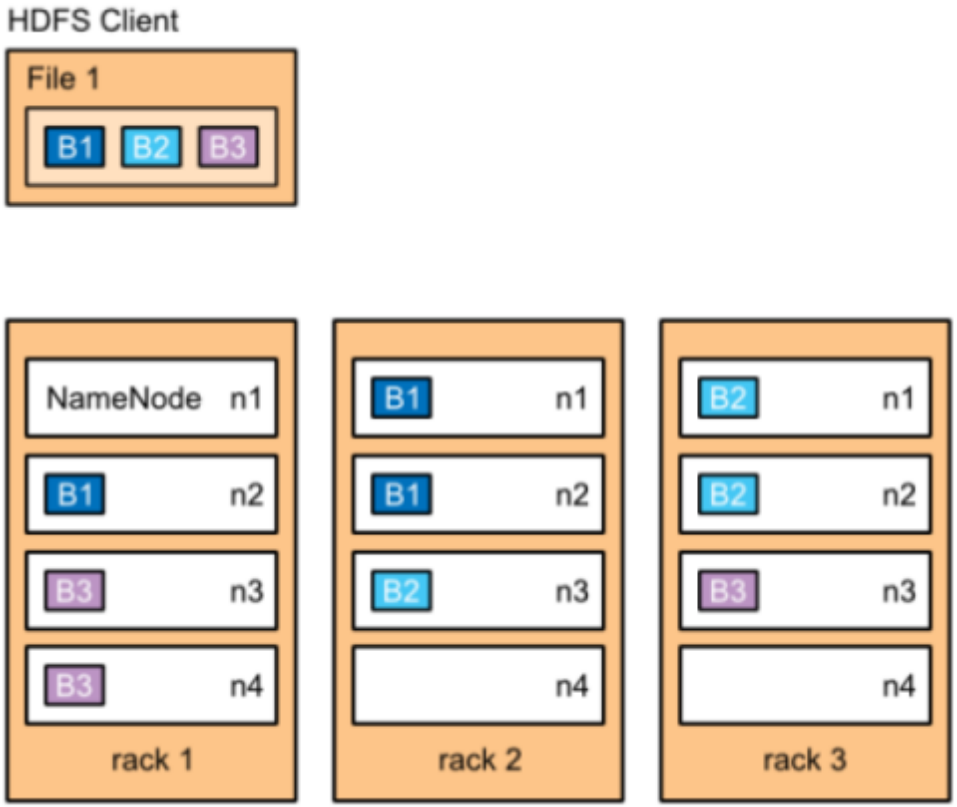
Writing a file to HDFS



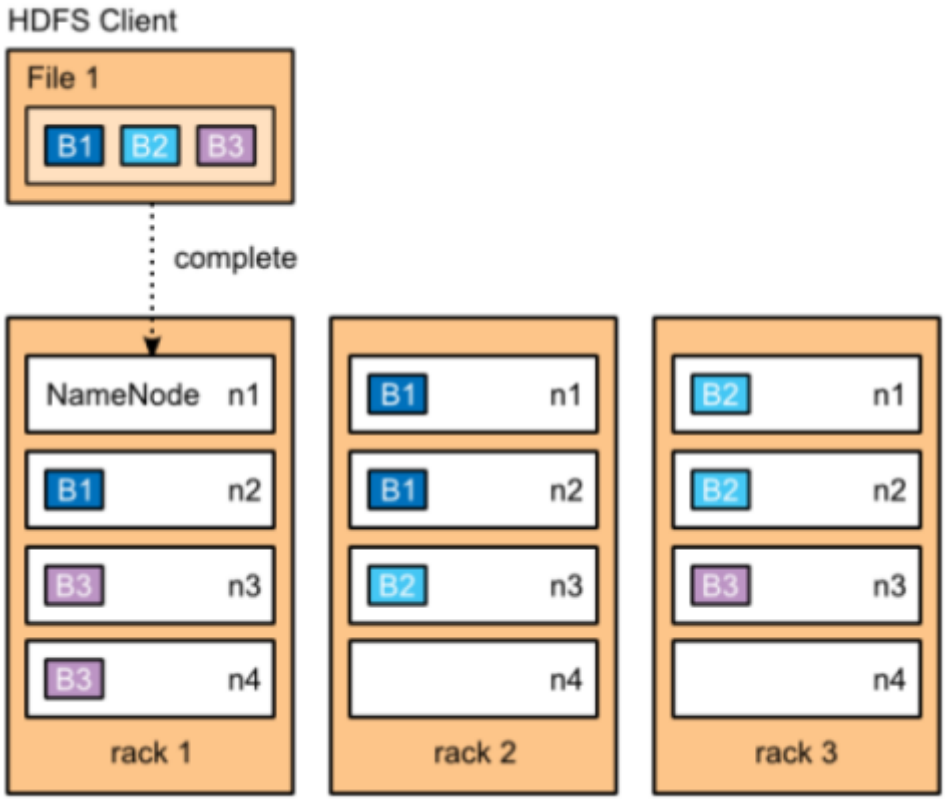
Writing a file to HDFS



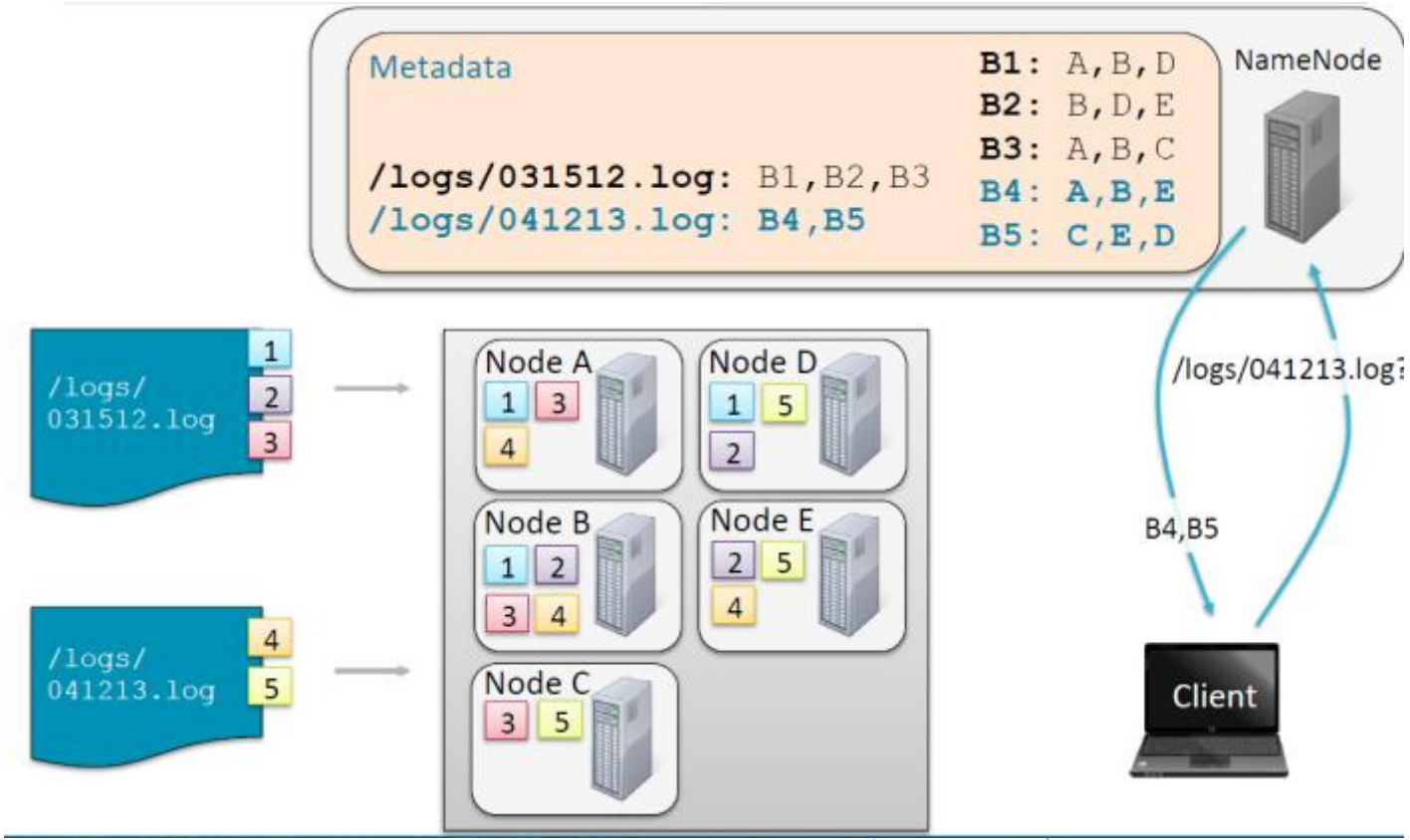
Writing a file to HDFS



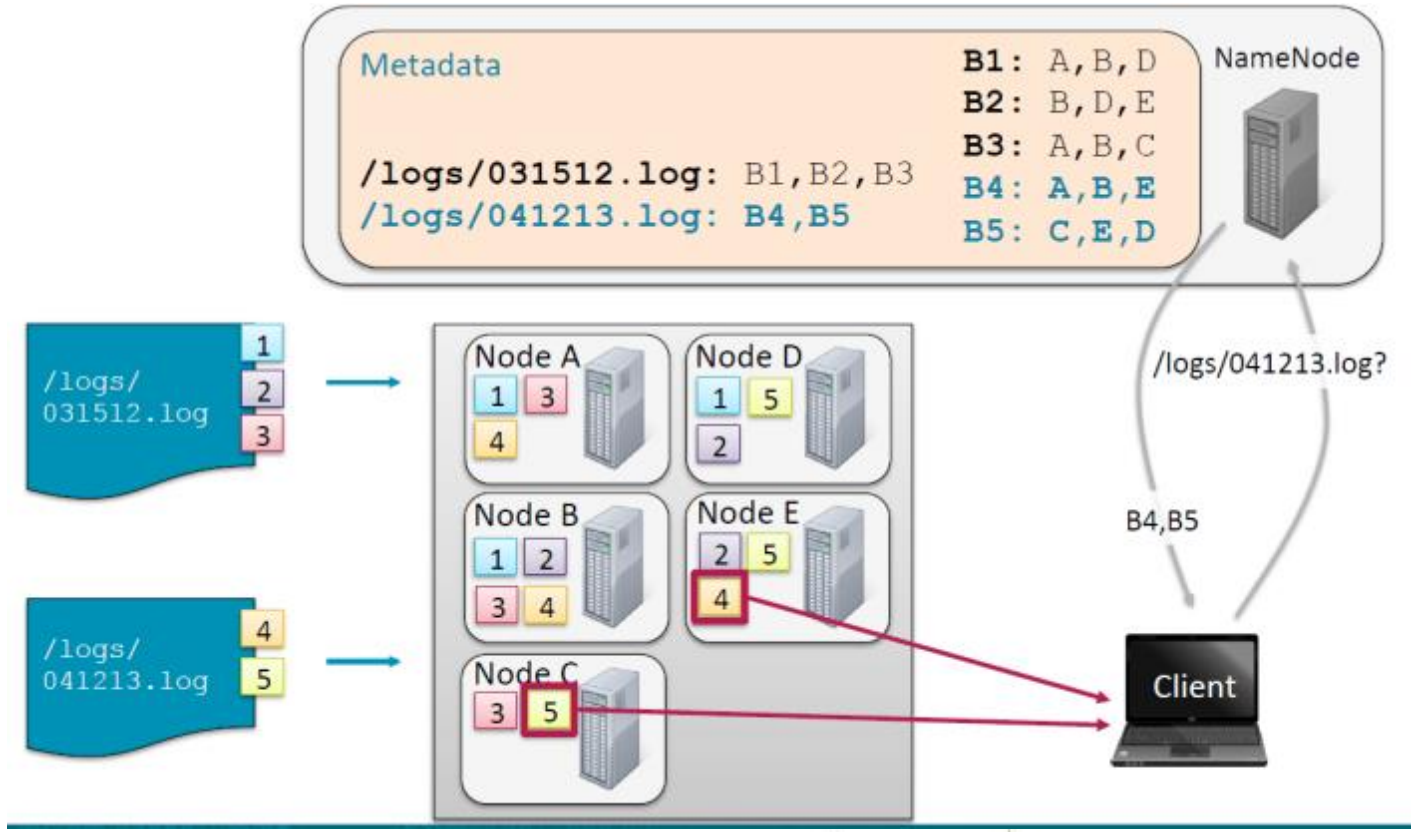
Writing a file to HDFS



Storing and Retrieving Files



Storing and Retrieving Files



ITAB Academy

Common Hadoop Commands

Common Hadoop Commands

Common Hadoop commands to manage HDFS:

1. Create a directory in HDFS at given path(s).

Usage:
hadoop fs -mkdir <paths>

Example:
hadoop fs -mkdir /user/saurzcode/dir1 /user/saurzcode/dir2

2. List the contents of a directory.

Usage :
hadoop fs -ls <args>

Example:
hadoop fs -ls /user/saurzcode

Common Hadoop Commands

Common HDFS commands:

3. Upload and download a file in HDFS.

***Upload*:**
hadoop fs -put:
Copy single src file, or multiple src files from local file system to the Hadoop data file system

Usage:
hadoop fs -put <localsrc> ... <HDFS_dest_Path>

Example:
hadoop fs -put /home/saurzcode/Samplefile.txt /user/saurzcode/dir3/

Download:
hadoop fs -get:
Copies/Downloads files to the local file system

Usage:
hadoop fs -get <hdfs_src> <localdst>

Example:
hadoop fs -get /user/saurzcode/dir3/Samplefile.txt /home/

Common Hadoop Commands

Common HDFS commands:

4. See contents of a file
Same as unix cat command:

Usage:
hadoop fs -cat <path[filename]>

Example:
hadoop fs -cat /user/saurzcode/dir1/abc.txt

5. Copy a file from source to destination
This command allows multiple sources as well in which case the destination must be a directory.

Usage:
hadoop fs -cp <source> <dest>

Example:
hadoop fs -cp /user/saurzcode/dir1/abc.txt /user/saurzcode/dir2

Common Hadoop Commands

Common HDFS commands:

6. Copy a file from/To Local file system to HDFS
copyFromLocal

Usage:
hadoop fs -copyFromLocal <localsrc> URI

Example:
hadoop fs -copyFromLocal /home/saurzcode/abc.txt /user/saurzcode/abc.txt
Similar to put command, except that the source is restricted to a local file reference.

copyToLocal

Usage:
hadoop fs -copyToLocal [-ignorecrc] [-crc] URI <localdst>
Similar to get command, except that the destination is restricted to a local file reference.

Common Hadoop Commands

Common HDFS commands:

7. Move file from source to destination.
Note:- Moving files across filesystem is not permitted.

Usage :
hadoop fs -mv <src> <dest>
Example:
hadoop fs -mv /user/saurzcode/dir1/abc.txt /user/saurzcode/dir2

8. Remove a file or directory in HDFS.
Remove files specified as argument. Deletes directory only when it is empty

Usage :
hadoop fs -rm <arg>

Example:
hadoop fs -rm /user/saurzcode/dir1/abc.txt
Delete file
hadoop fs –rm -r <arg>
Or hadoop fs –rmr <arg>

Usage :
hadoop fs -rmr <arg>
Example:
hadoop fs -rmr /user/saurzcode/

Common Hadoop Commands

- Common HDFS commands:

9. Display last few lines of a file.
Similar to tail command in Unix.
Usage :
hadoop fs -tail <path[filename]>
Example:
hadoop fs -tail /user/saurzcode/dir1/abc.txt

10. Display the aggregate length of a file.
Usage :
hadoop fs -du <path>
Example:
hadoop fs -du /user/saurzcode/dir1/abc.txt

oumsis@gmail.com