

B I G D A T A



# LES FONDAMENTAUX D'HADOOP

## HDP



# AGENDA

Le stockage des données dans HDFS




B I G D A T A



## Le stockage des données dans HDFS



# Objectif


- 
- Hadoop Distributed file system
  - Architecture HDFS
  - Le Name Node
  - Le Data Node
  - Le Name Node Secondaire
  - Intégration des données dans le HDFS
  - Les blocs HDFS
  - La réplication des blocs
  - Les commandes HDFS

# Hadoop Distributed File System



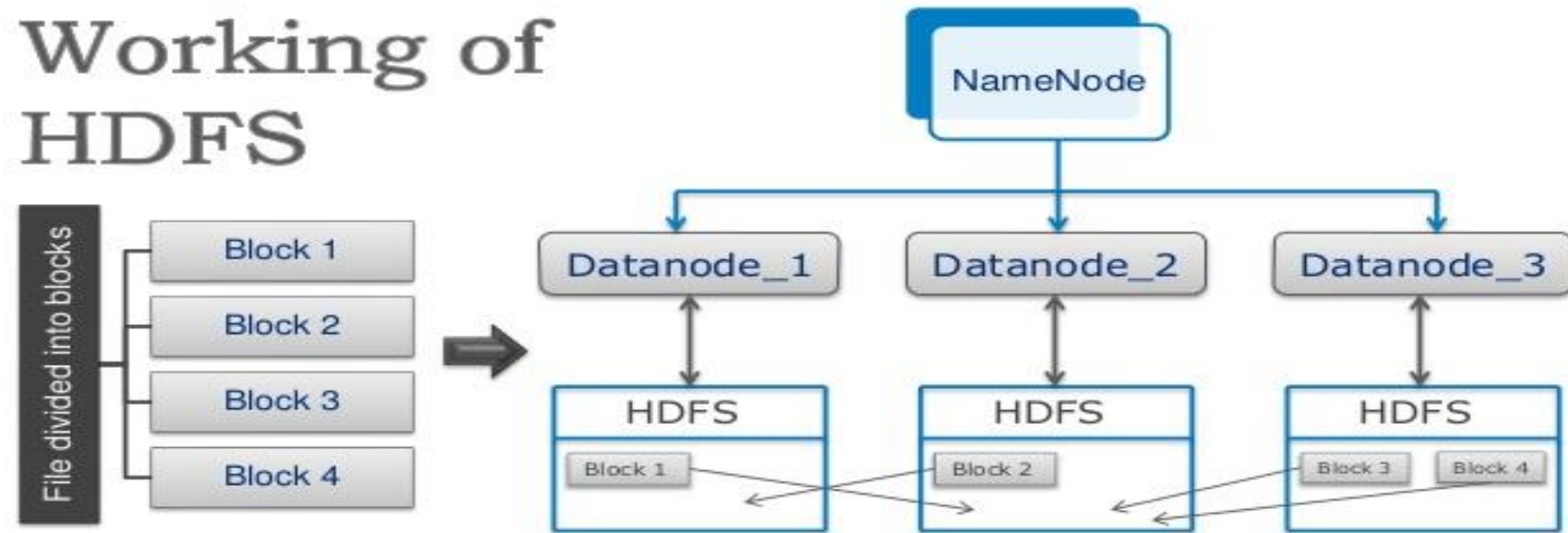


# Hadoop Distributed File System

- HDFS (Hadoop Distributed File System) est un système de fichier distribué permettant de stocker et de récupérer des fichiers en un temps record.
  - Il s'agit de l'un des composants basiques du Framework Hadoop Apache, et plus précisément de son système de stockage. HDFS Hadoop compte parmi les projets « top level » de Apache.
  - Développé en Java
  - Tolérance aux pannes
  - Haute disponibilité
  - Il fonctionne sur une théorie du modèle d'accès en écriture une fois-lecture-plusieurs pour les fichiers.
- 

# Architecture HDFS

## Working of HDFS



Storage & Replication of Blocks in HDFS



# Architecture HDFS

- HDFS suit l'architecture maître esclave.
- Chaque cluster possède un seul nœud maître et plusieurs nœuds esclave
- Les données sont divisées en plusieurs blocs.
- Chaque bloc est stocké sur plusieurs machines, en fonction de facteur de réplication.
- Le nœud maître stocke et gère l'espace de noms du système de fichiers, cela veut dire des informations sur les blocs des fichiers, tels que les emplacement des blocs les autorisations etc.
- Les nœuds esclave stockent les blocs des données des fichiers.
- Le nœud maître est le Name Nœud et les esclaves sont les Data Nœud






# Le Name Node HDFS

- Le Name Node est la pièce maîtresse de HDFS
- Il maintient et gère l' espace de noms du système de fichiers
- Le Name Node stock les informations sur les emplacements des blocs, les autorisations etc...
- Les blocs sont stockés sur le disque sous la forme des deux fichiers :
  - Fimage : Fimage signifie image du système de fichiers. Il contient l'espace de noms complet du système de fichiers Hadoop depuis la création du NameNode.
  - Journal d'édition : il contient toutes les modifications récentes apportées à l'espace de noms du système de fichiers par rapport à la Fimage la plus récente.




# Le fonctionnement Name Node

- Name node gère et maintient les data nodes
  - Il détermine le mappage des blocs d'un fichiers vers le data node
  - Il enregistre toutes les modifications apportées à l'espace de noms du système d'un fichiers.
  - Il garde l'emplacement de chaque bloc d'un fichier.
  - Il prend en charge le facteur de réplication des blocs
  - Il reçoit le rapport de l'état de santé de chaque bloc
  - si le data échoue ,le NameNode choisit de nouveaux DataNodes pour les nouveaux réplicas.
- 



# Le fonctionnement DataNode

- Les DataNode sont les nœuds esclaves dans le HDFS.
  - Ils sont des machines de bases moins coûteuses
  - Le data nœud est soumis de répondre aux demandes clients (lectures et écritures)
  - À la demande de NameNode le DataNode effectue les opérations de création , de suppression et modification de blocs.
  - Le DataNode envoie des pulsations à NameNode pour signaler l'état de santé de HDFS.
  - Le dataNode informe le NameNode sur les blocs qu'il contient
- 



# Le Name Node Secondaire

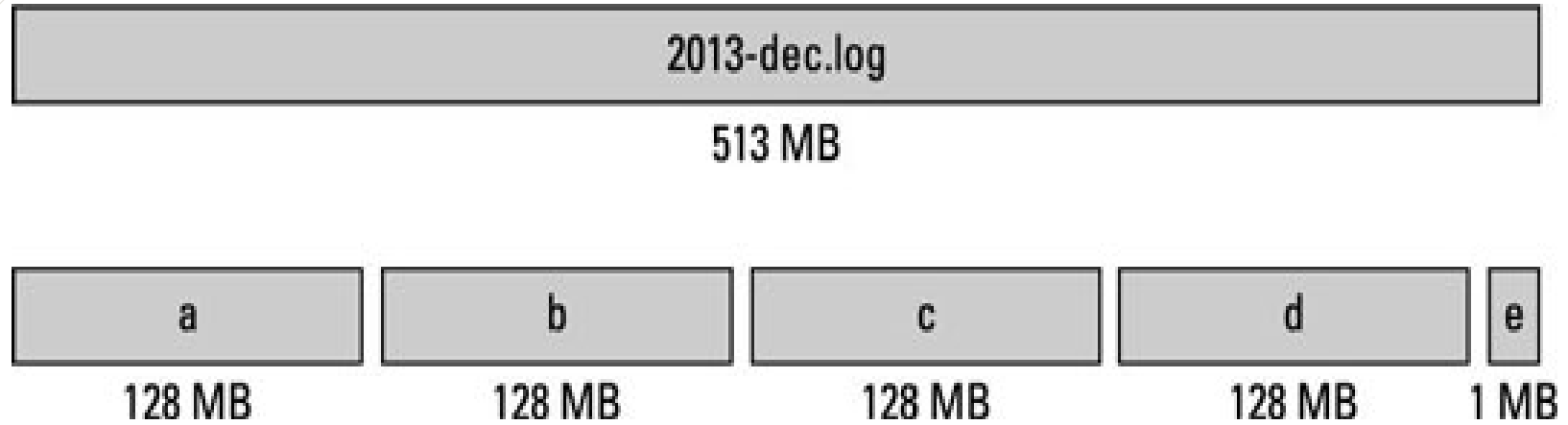
- Le NameNode secondaire fonctionne comme un assistant de NameNode principale et ne le remplace pas.
- Lorsque les logs des journaux de NameNode devient très importants ce dernier prend plus de temps pour le redémarrage ce qui n'est pas optimale.
- Le NameNode secondaire résout ce problème.
- Le NameNode Secondaire copie le fichier Fsimage et modifie les fichiers journaux à partir du NameNode.
- Il applique souvent le fichier Fsimage et actualise les journaux d'éditions
- Dans le cas ou le NameNode échoué , la dernière version de Fsimage enregistrée dans le NameNode secondaire peut être utilisé.



# Intégration des données dans HDFS

- Intégration des données dans le HDFS se fait de la façons suivantes :
  - Une application cliente envoie une demande au NameNode pour placer le fichier dans le HDFS.
  - Le NameNode définit la façons dont les données seront divisées en bloc.
  - Il choisit les Datanodes qui vont être utilisés pour stocker ce fichier.
  - Ces informations sont fournit à l'application cliente.
  - L'application cliente communique directement avec les DataNodes.


# Les blocs HDFS



- Le HDFS divise le fichier en morceaux appelé bloc, la taille d'un bloc par défaut est 128 MB.
- La taille du bloc peut être configurée selon le besoin
- L'utilisateur n'a aucun contrôle sur l'emplacement du bloc



# La réplication des blocs

- 
- Pour un système distribué les données doivent être dupliquées dans plusieurs endroits afin d'éviter la perte de données si une machine tombe en panne et éventuellement les données restent accessibles.
  - Le facteur de réplique est 3 par défaut mais il reste configurable.
  - Par exemple si la taille de fichier est 128 MO et le facteur de réplique est 3 alors l'espace disque occupé par ce fichier est : 384 Mo.
  - Ce mécanisme de réplication rend le HDFS tolérant aux pannes

# Les commandes HDFS

- L'application hdfs est une application cliente Hadoop qui permet d'émettre des commandes à HDFS à partir d'une ligne de commande .
- Toutes les commandes hadoop sont appelées par le script /usr/bin/hdfs . L'exécution du script hdfs sans aucun argument imprime la description de toutes les commandes .
- La commande hdfs a la syntaxe suivante :

**Hdfs [COMMAND] [GENERIC\_OPTIONS] [COMMAND\_OPTIONS]**



# Les commandes HDFS

```
[root@sandbox-hdp ~]# hdfs
Usage: hdfs [--config confdir] [--loglevel loglevel] COMMAND
    where COMMAND is one of:

    dfs                run a filesystem command on the file systems supported in Had
    classpath          prints the classpath
    namenode -format    format the DFS filesystem
    secondarynamenode  run the DFS secondary namenode
    namenode           run the DFS namenode
    journalnode        run the DFS journalnode
    zkfc               run the ZK Failover Controller daemon
    datanode           run a DFS datanode
    dfsadmin           run a DFS admin client
    envvars            display computed Hadoop environment variables
    haadmin            run a DFS HA admin client
    fsck              run a DFS filesystem checking utility
    balancer          run a cluster balancing utility
    jmxget            get JMX exported values from NameNode or DataNode.
    mover             run a utility to move block replicas across
                    storage types
    oiv              apply the offline fsimage viewer to an fsimage
    oiv_legacy        apply the offline fsimage viewer to an legacy fsimage
    oev              apply the offline edits viewer to an edits file
    fetchdt          fetch a delegation token from the NameNode
    getconf           get config values from configuration
    groups            get the groups which users belong to
    snapshotDiff      diff two snapshots of a directory or diff the
                    current directory contents with a snapshot
    lsSnapshottableDir list all snapshottable dirs owned by the current user
                    Use -help to see options

    portmap          run a portmap service
    nfs3             run an NFS version 3 gateway
    cacheadmin       configure the HDFS cache
    crypto           configure HDFS encryption zones
    storagepolicies  list/get/set block storage policies
    version          print the version
```

Most commands print help when invoked w/o parameters.

# Les commandes HDFS

- Version : pour imprimer la version Hadoop
- mkdir : pour créer un nouveau répertoire
- ls : pour lister les fichiers et répertoires présents dans HDFS.
- Put : pour copier le fichier du système local vers le système du fichier Hadoop.
- CopyfromLocal : le même principe que put .
- Get : pour copier le fichier ou le répertoire du système du fichier hadoop vers le système du fichier local.
- CopyToLocal : elle copie le fichier de HDFS vers le système de fichiers local.
- Cat : pour afficher le contenu du fichier dans le console.
- Mv : déplace des fichiers ou répertoires de la source vers ma destination.
- CP : copie un fichier d'un répertoire vers un autre répertoire au sein de HDFS.



# LAB.

# BIG DATA

QUESTION ?



# B I G   D A T A

Merci !!

▲ 18.75

▼ 25.01

▼ 22.10

▲ 25.21

▼ 30.12

▲ 29.79

▼ 18.07

▲ 07.28

▲ 24.78