esprit
Ecole Supérieure Privée
d'Ingénierie et de Technologies

Chp3 –HDFS

2016-2017

Mme Ines Slimene
ines.slimen@esprit.tn

Plan module

- Introduction
- Écosystème Hadoop
- **HDFS**
- MapReduce
- Langages de requête Hadoop :Pig, Hive
- SGBDNR
  - Différences entre une BDNR et une BD relationnelle
  - Typologies des BD non relationnelles
- Etude d'un SGBDNR : HBase

Plan

- Objectifs de HDFS
- Concepts de base de HDFS
  - DataNode
  - NameNode
  - NameNode secondaire
- Lecture et écriture de données
- HDFS 2
  - HDFS Federation
  - Haute disponibilité du NameNode
- Commandes Shell

Pourquoi DFS ?

Lecture de 1 TB de données

1 Machine

4 I/O canal
Chaque canal – 100MB/s

45 minutes

10 Machines

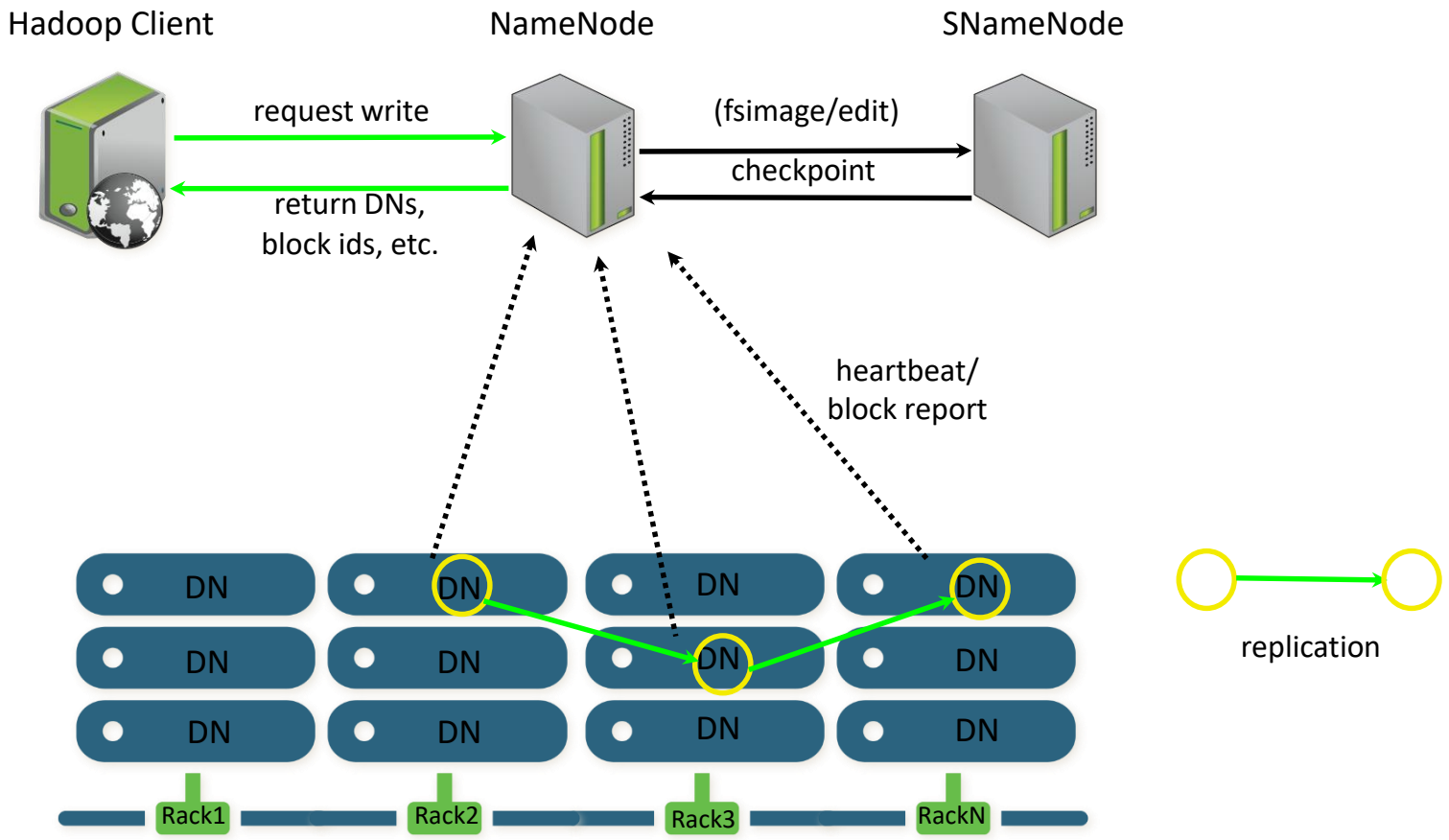
4 I/O canal
Chaque canal – 100MB/s

4.5 minutes

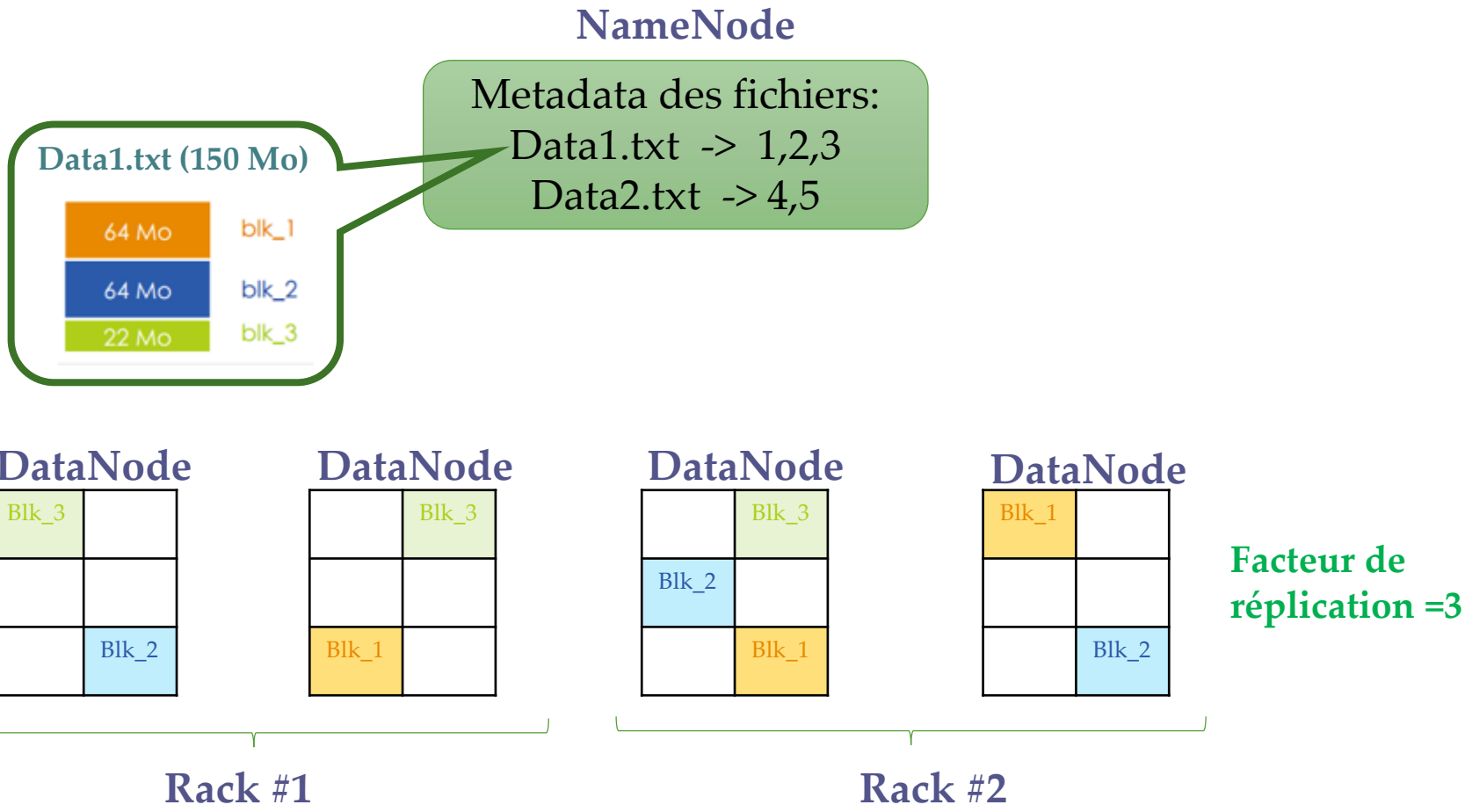
Objectifs de HDFS

- Garantir l'intégrité des données malgré une défaillance système (plantage d'une machine)
- Permettre un traitement rapide des fichiers d'une grande taille (GB et plus)
- HDFS est moins adapté à beaucoup de fichiers d'une petite taille
- Rapprocher la lecture de la localisation des fichiers plutôt que de rapprocher les fichiers à la lecture

Concepts de base de HDFS



Écriture d'un fichier

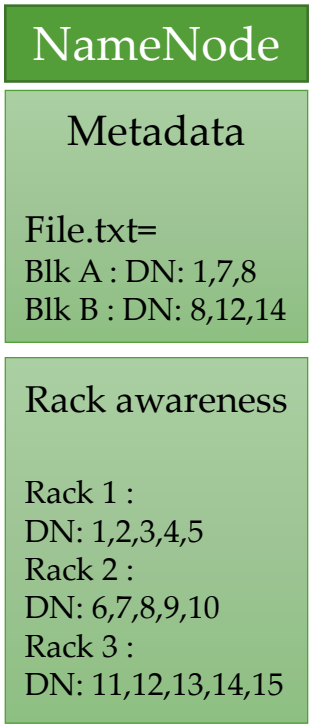


Problèmes DataNode

- Si l'un des nœuds a un problème, les données seront perdues
- Solution :
  - Hadoop réplique chaque bloc 3 fois
  - Il choisit 3 nœuds au hasard, et place une copie du bloc dans chacun d'eux
  - Si le nœud est en panne, le NN le détecte, et s'occupe de répliquer encore les blocs qui y étaient hébergés pour avoir toujours 3 copies stockées

NameNode : Méta data

- Types de Méta-data
  - Liste de fichiers
  - Liste de blocs pour chaque fichiers (taille par défaut 64MB)
  - Liste des dataNodes pour chaque bloc
  - Attributs de fichier, exemple : temps d'accès, facteur de réplication
  - Fichier log
  - Enregistrement de création et suppression de fichiers.



NameNode

- Chaque namenode contient les fichiers suivants :
  - Version :
    - Le fichier qui contient les informations sur le namenode.
    - On y retrouve : l'identifiant pour le système de fichiers, la date de création, le type de stockage, la version de la structure des données HDFS.
  - Edits :
    - Regroupe toutes les opérations effectuées dans le système.
    - Chaque fois quand une action d'écriture est effectuée, elle est d'abord placée dans ce fichier de logs. Le fichier edits sert un peu comme le backup
  - Fsimage :
    - C'est un fichier binaire utilisé comme un point de contrôle pour les méta-données.
    - Il joue aussi un rôle important dans la récupération des données dans le cas d'une défaillance du système
  - Fstime : stocke les informations sur la dernière opération de contrôle sur les données.

Problème NameNode

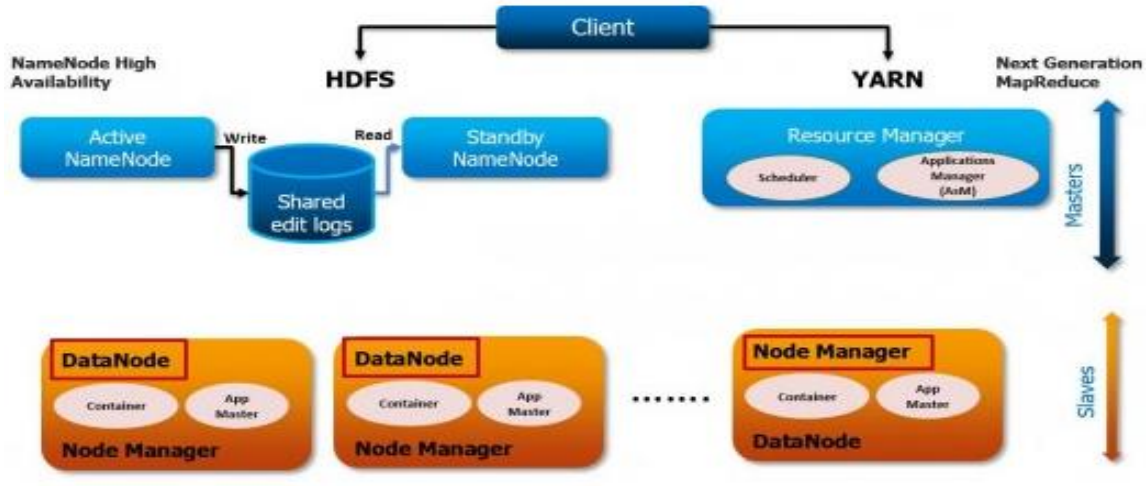
- Limité à 4,000 nœud par cluster
- Dans le cas d'un problème d'accès (réseau), les données seront temporairement inaccessibles
- Si le disque du NN est défaillant, les données seront perdues à jamais
- Dans ce cas, l'administrateur doit restaurer le NameNode, MANUELLEMENT, en utilisant le NameNode secondaire.

Namenode secondaire

- Son arborescence de répertoire est quasi identique a celui du **NN primaire**.
- Le **NN secondaire** lit les méta-data à partir du RAM du **NN** et les écrit sur le disque dur.
- **Le NN secondaire** vérifie périodiquement l'état du **NN primaire**.
- A chaque vérification, il copie le fichier edits à partir du **NN primaire** à des intervalles régulier et applique les modifications au fichier fsimage.
- Le nouveau fsimage est copié sur le **NN primaire**, et est utilisé au démarrage suivant du **NN primaire**
- Le **namenode secondaire** n'est pas :
  - Un fail-over pour le Namenode
  - Ne garantit pas la haute disponibilité
  - N'améliore pas les performances du namenode

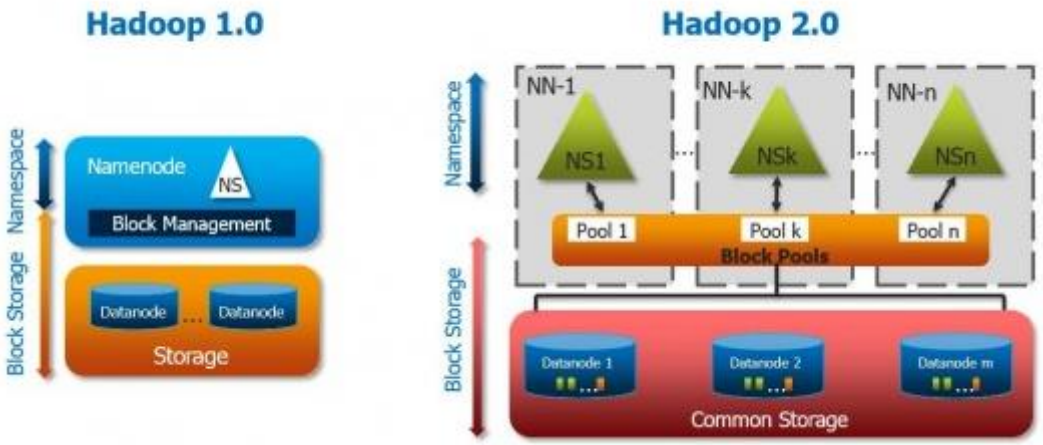
Hadoop1 vs Hadoop2

Propriété	Hadoop 1.0	Hadoop 2.0
Fédération	Un seul namenode	Plusieurs namenode
Haute disponibilité	inexistante	Hautement disponible
Gestion des ressources	jobTracker et task tracker	Manager de ressource, manager de nœud



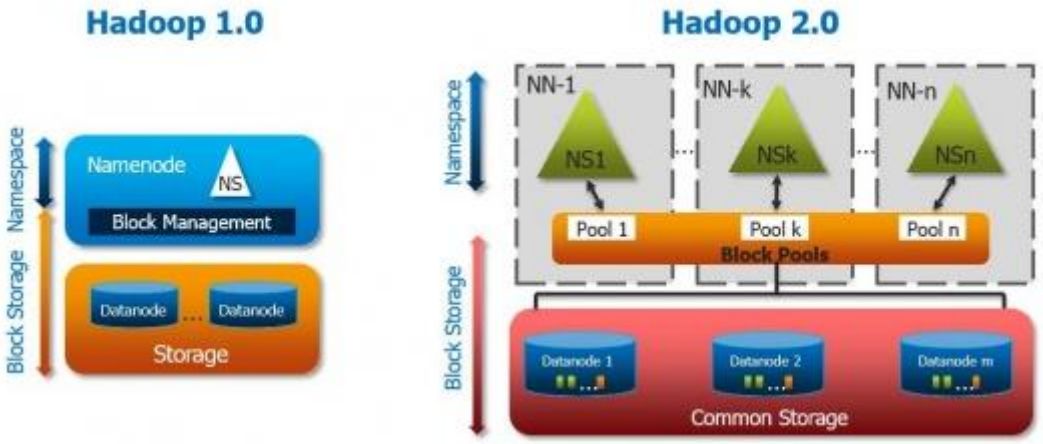
Avantages HDFS2

- **HDFS Federation –** scalabilité horizontale du namenode
- **Haute disponibilité du NameNode –** NameNode n'est plus un 'Single Point of Failure'



Fédération HDFS

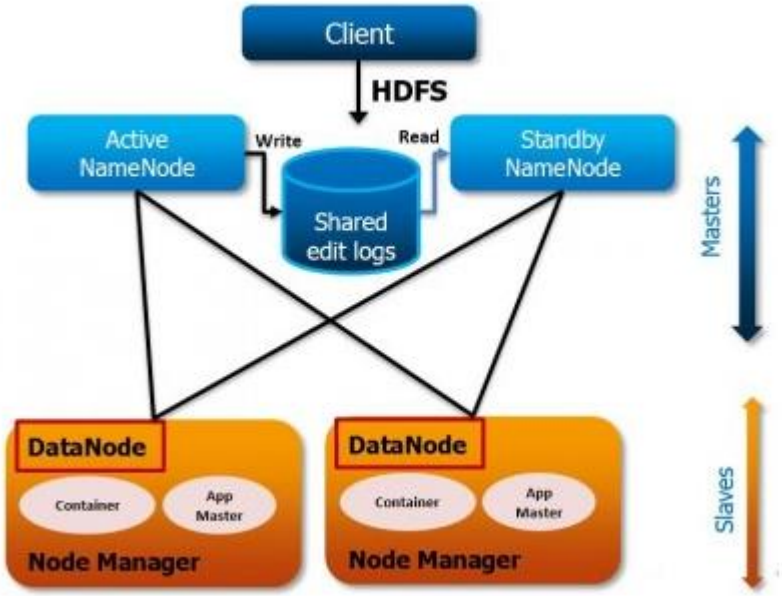
- Les Namenodes sont indépendants
- Pas de coordination entre les différents NN
- Les dataNodes sont des supports de stockage des blocs provenant de tous les NN
- Les DN envoie un heartbeat et un rapport à tous les NN.



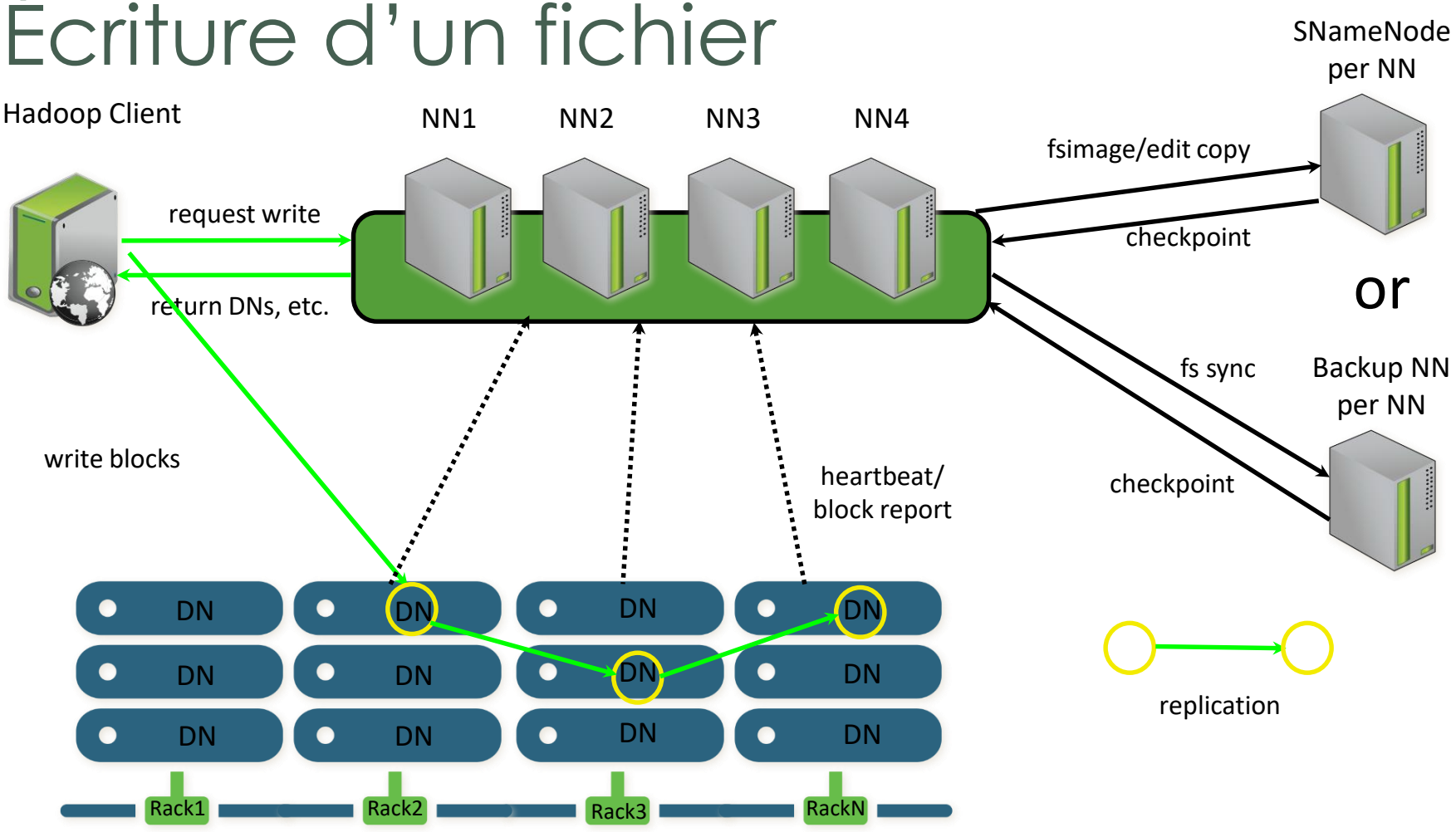
Haute disponibilité du NameNode

- Tous les fichiers edits seront stockés dans un système de fichier partagé
- A un instant données, un seul NN écrit dans les fichiers edits
- LE NN passive lit à partir de ce fichier et garde des informations des meta-data à jour

- En cas de panne de l'active NameNode, le NN passive devient le NN active et commence a écrire dans l'espace partagé.



Écriture d'un fichier



Configuration de l'architecture matérielle dans des fichiers XML

- Configuration du NameNode (master)

conf/core-site.xml:

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

conf/masters:

localhost

conf/mapred-site.xml:

```
<configuration>
     <property>
          <name>mapred.job.tracker</name>
          <value>localhost:9001</value>
     </property>
</configuration>
```

Configuration de l'architecture matérielle dans des fichiers XML

- Configuration du/des DataNode(s) (slaves)

conf/slaves:

localhost

- Configuration de la réplication

conf/hdfs-site.xml:

```
<configuration>
     <property>
          <name>dfs.replication</name>
          <value>1</value>
     </property>
</configuration>
```

Commandes Shell

- Usage: $hadoop fs -<command> -<option><URI>
  - Exemple $ hadoop fs -ls /
- URI usage:
  - HDFS: $ hadoop fs -ls hdfs://localhost/to/path/dir
  - Local: $ hadoop fs -ls file:///to/path/file3
- La plupart des commandes sont similaires aux commandes UNIX
  - ls, cat, du, etc..
- Lister les commandes hdfs
  - $ hadoop fs -help

Commandes Shell

- **cat – afficher le contenu d'un document**
  - Tout le document: $ hadoop fs -cat /dir/file.txt
  - Afficher les 25 premières lignes du fichier file.txt
    - $ hadoop fs -cat /dir/file.txt | head -n 25
- **cp – copier un fichier de HDFS vers HDFS**
  - $ hadoop fs -cp /dir/file1 /otherDir/file2
- **ls – afficher le contenu d'un répertoire**
  - $ hadoop fs -ls /dir/
- **mkdir – créer un chemin**
  - $ hadoop fs -mkdir /brandNewDir

Déplacement de données

- **mv – déplacer un fichier**
  - $ hadoop fs -mv /dir/file1 /dir2/file2
- **put – copier du système de fichier courant vers hdfs**
  - $ hadoop fs -put localfile /dir/file1
  - on peut utiliser aussi copyFromLocal
- **get – copier des fichiers a partir de hdfs**
  - $ hadoop fs -get /dir/file localfile
  - on peut utiliser aussi copyToLocal

Suppression de données

- **rm – supprimer des fichiers**
  - $ hadoop fs -rm /dir/fileToDelete
- **rmr – supprimer un repetoire**
  - $ hadoop fs –rm -r /dirWithStuff

États des fichiers

- **du : afficher les tailles des fichiers d'un répertoire (en bytes)**
  - hadoop fs -du /someDir/
- **du –h : afficher les tailles des fichiers d'un répertoire (en octets)**
  - hadoop fs -du –h /someDir/
- **du –s : afficher la taille totale d'un répertoire**
  - hadoop fs –du -s /someDir/

Commande fsck

- Contrôle les incohérences
- Reporte les problèmes
  - Blocs manquants
  - Blocs non répliqués
- Ne corrige pas les problèmes
  - C'est le Namenode qui corrige les erreurs reportées par fsck.
- $ hadoop fsck <path>
  - Exemple : $ hadoop fsck /

Commande DFSAdmin

- **Operations administratives du HDFS**
  - $ hadoop dfsadmin <command>
  - Exemple: $ hadoop dfsadmin -report
- **report :** affiche les statistiques du HDFS
- **safemode :** entrer en mode safemode
  - Maintenance, sauvegarde, mise a jours, etc..