

UNIVERSITÉ PARIS-SACLAY POLYTECH PARIS-SACLAY



Big Data

Technologies Internes d'Hadoop

Stéphane Vialle & Gianluca Quercini



1

POLYTECH PARIS-SACLAY

Principes et technologie d'Hadoop

1. **Localité des données et des traitements**
2. Framework d'Hadoop
3. Mécanismes du Map-Reduce d'Hadoop
4. Système de fichiers distribué d'Hadoop (HDFS)
5. Allocation et gestion de ressources d'Hadoop

2

POLYTECH PARIS-SACLAY

Localité des données et des traitements

C'est toujours l'accès aux données qui coute cher, pas le calcul lui-même (une fois les données arrivées dans l'unité de calcul)

Big Data façon Hadoop :
 Amener les codes de traitements aux données
 → Transformer momentanément en nœuds de traitement les nœuds de stockage des données traitées
 → Eviter de déplacer des données (très volumineuses) ... mais ...
 → les relire et les réécrire localement chaque fois que la RAM est pleine

Big Data façon Spark :
 Amener les données à toutes les unités de calcul disponibles
 → Mais ... lire les données et les écrire sur disque une seule fois
 → et ... garder les données en RAM durant tout le traitement

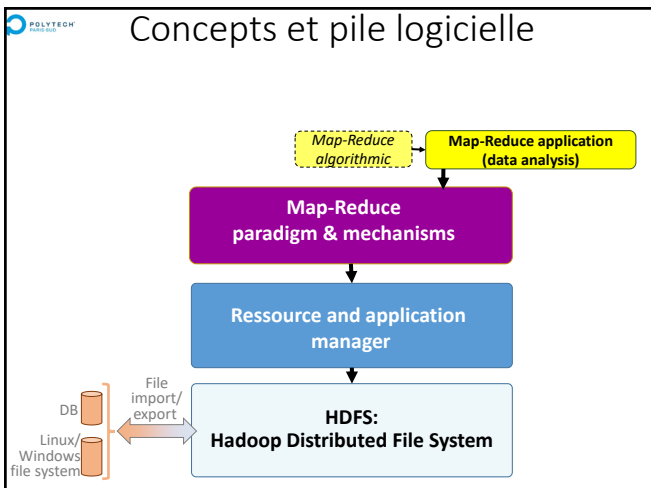
3

POLYTECH PARIS-SACLAY

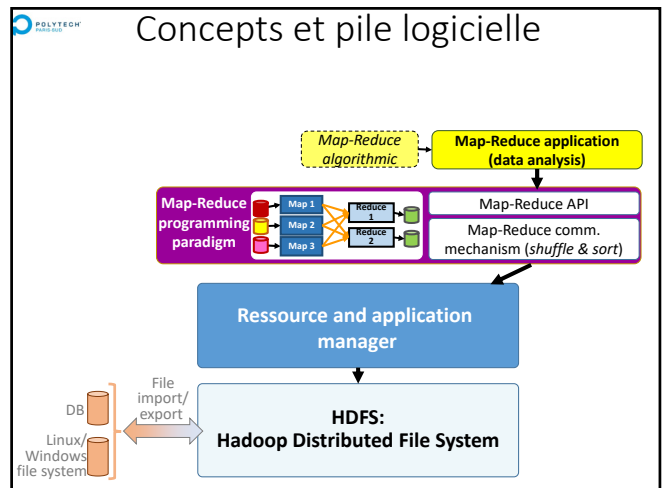
Principes et technologie d'Hadoop

1. Localité des données et des traitements
2. **Framework d'Hadoop**
3. Mécanismes du Map-Reduce d'Hadoop
4. Système de fichiers distribué d'Hadoop (HDFS)
5. Allocation et gestion de ressources d'Hadoop

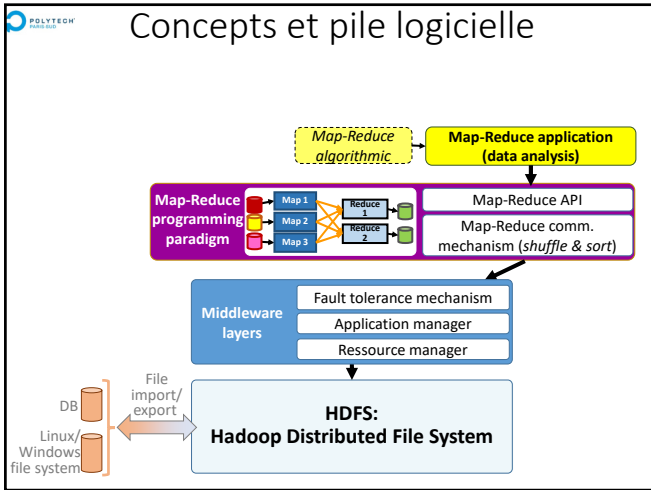
4



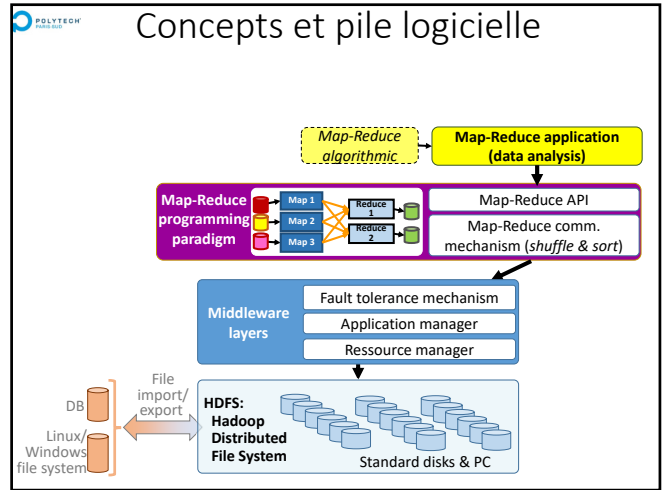
5



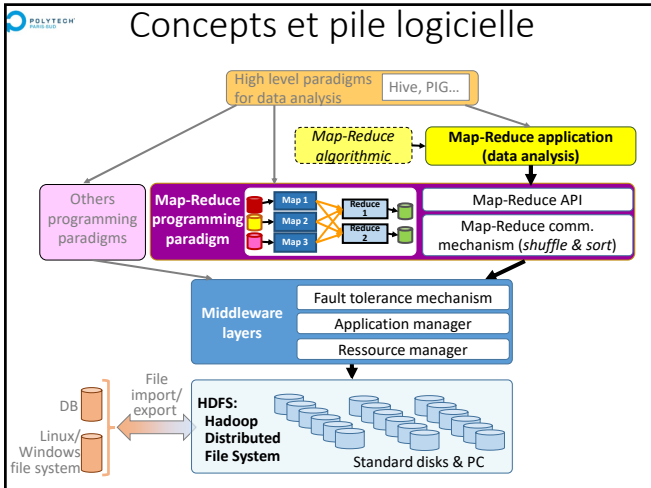
6



7



8

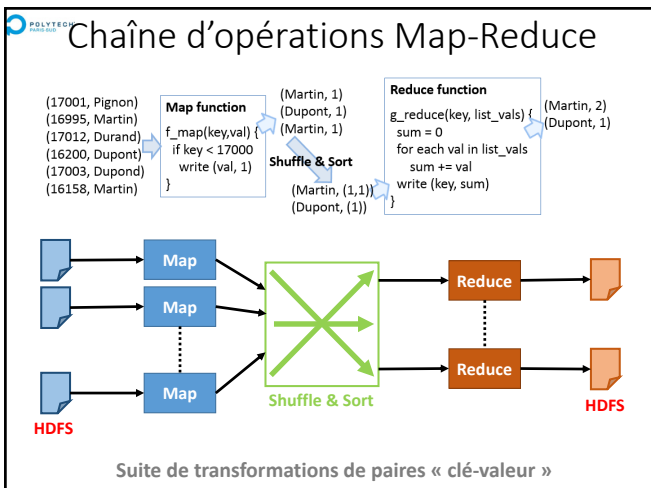


9

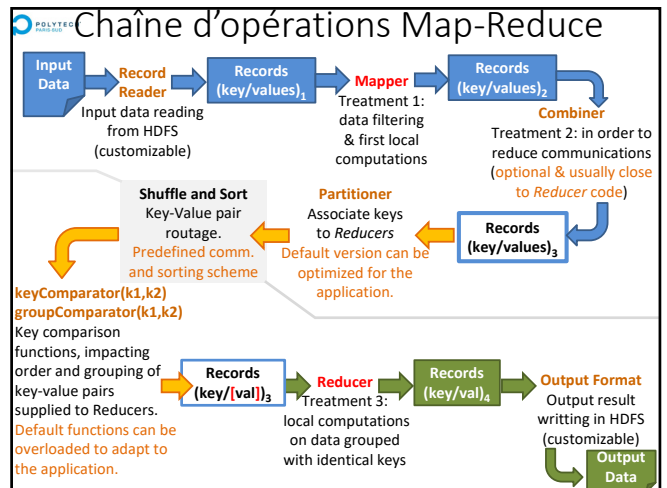
Principes et technologie d'Hadoop

1. Localité des données et des traitements
2. Framework d'Hadoop
3. Mécanismes du Map-Reduce d'Hadoop
4. Système de fichiers distribué d'Hadoop (HDFS)
5. Allocation et gestion de ressources d'Hadoop

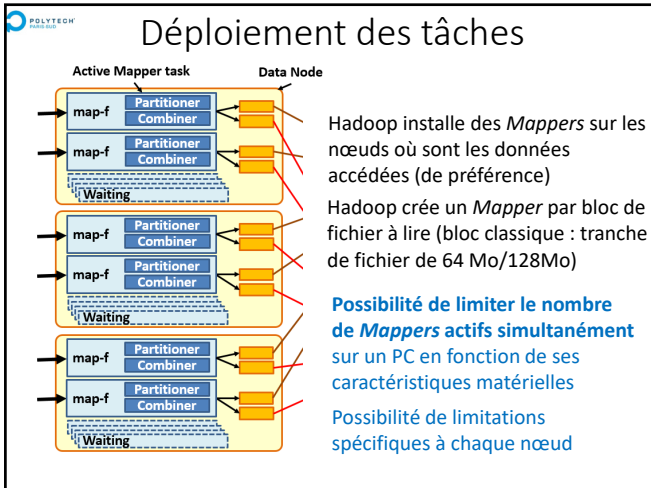
10



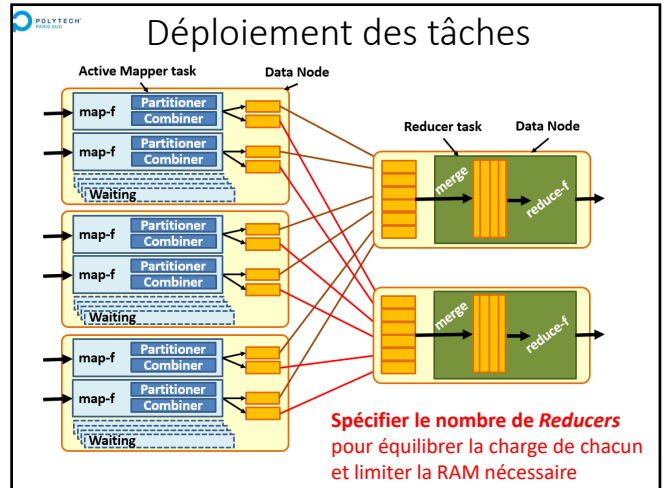
11



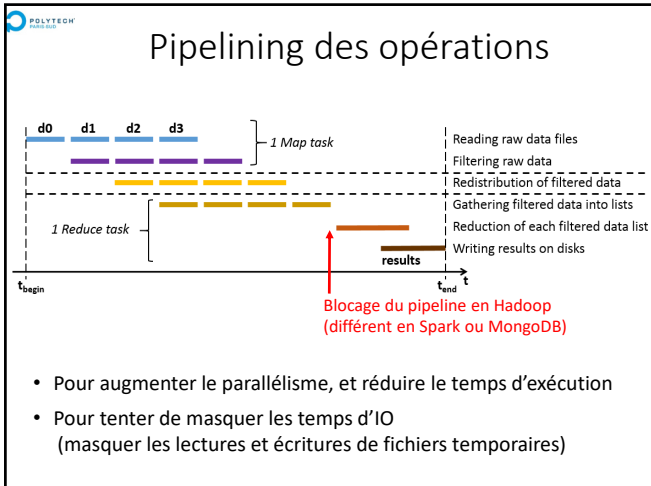
12



13



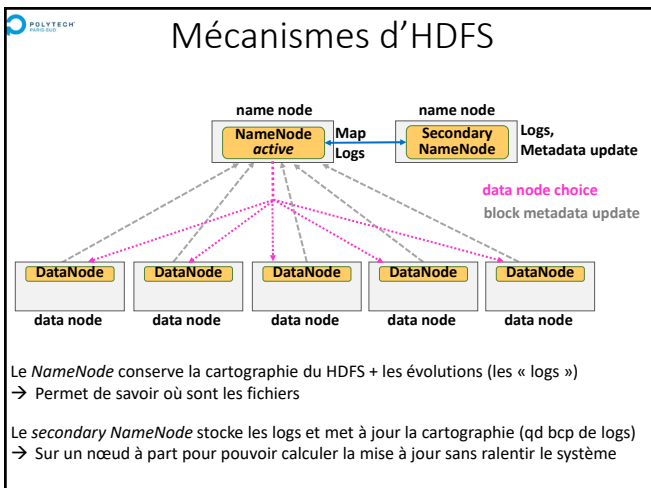
14



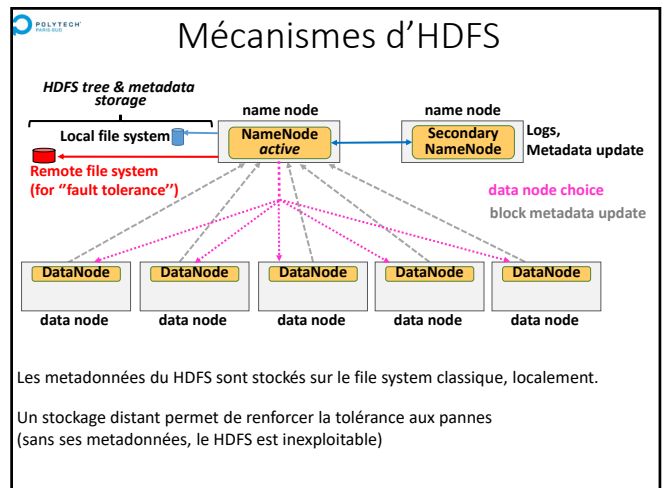
15

- ### Principes et technologie d'Hadoop
1. Localité des données et des traitements
 2. Framework d'Hadoop
 3. Mécanismes du Map-Reduce d'Hadoop
 4. **Système de fichiers distribué d'Hadoop (HDFS)**
 - Distribution et réplication des fichiers sous HDFS
 - Lecture de fichiers sous HDFS
 - Ecriture de fichiers sous HDFS
 5. Allocation et gestion de ressources d'Hadoop

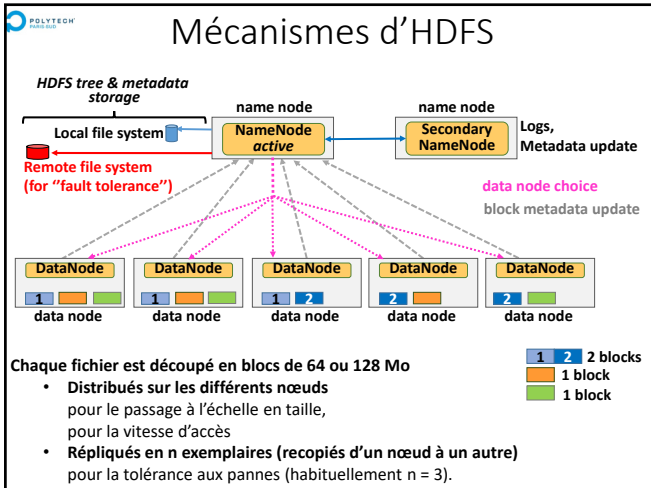
16



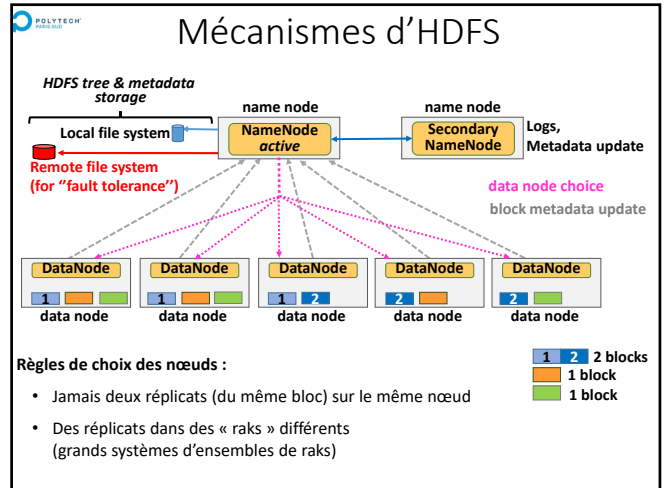
17



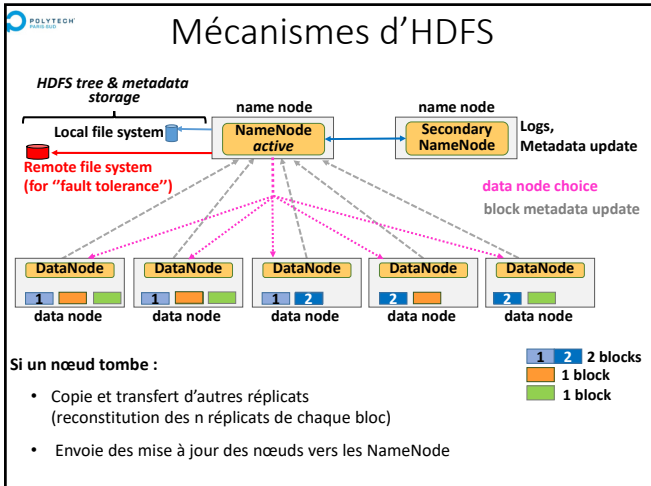
18



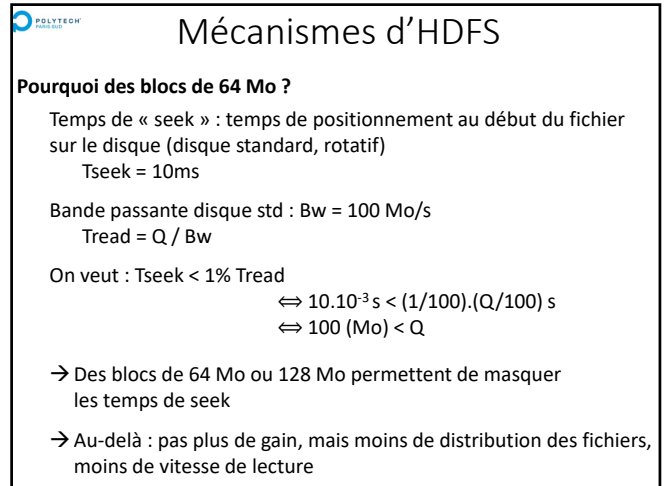
19



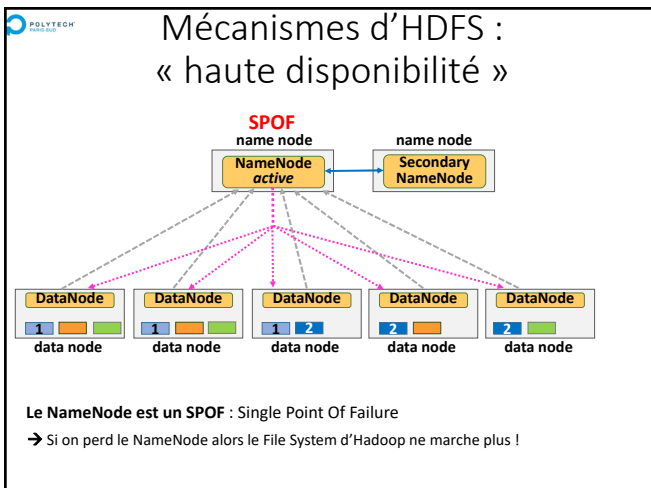
20



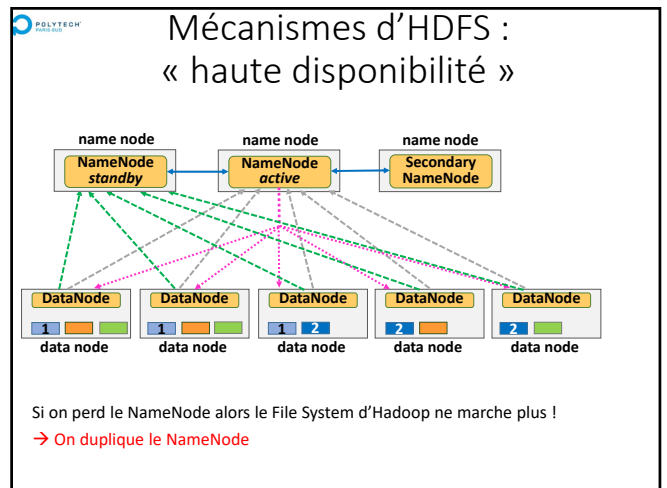
21



22



23



24

Mécanismes d'HDFS : « haute disponibilité »

- Le NameNode en *standby* est passé *actif* « très rapidement »
- Il n'y a plus de SPOF
- On « ne sent plus passer la panne » : Haute Disponibilité

25

Principes et technologie d'Hadoop

1. Localité des données et des traitements
2. Framework d'Hadoop
3. Mécanismes du Map-Reduce d'Hadoop
- 4. Système de fichiers distribué d'Hadoop (HDFS)**
 - Distribution et réplication des fichiers sous HDFS
 - **Lecture de fichiers sous HDFS**
 - **Ecriture de fichiers sous HDFS**
5. Allocation et gestion de ressources d'Hadoop

26

Mécanismes de lecture d'HDFS

- 0 – Création d'un objet permettant de s'interfacier à HDFS (un stub/proxy d'HDFS)
- 1 – Demande d'ouverture d'un fichier HDFS en lecture (« open »)
- 2 – Demande de localisation du fichier
 - 2a - Le proxy interroge le NameNode : pour savoir quels blocs lire et où les lire
Le NameNode répond au proxy
 - 2b - Le proxy crée et retourne un objet lecteur spécialisé sur le fichier ciblé

27

Mécanismes de lecture d'HDFS

- 3 – Le client exploite le lecteur du fichier
- 4 – Le lecteur accède au data node du premier bloc et récupère ce bloc
- 5 – Le lecteur accède au data node du second bloc et récupère ce bloc
- 6 – Le client referme le fichier en s'adressant au lecteur du fichier

28

Principes et technologie d'Hadoop

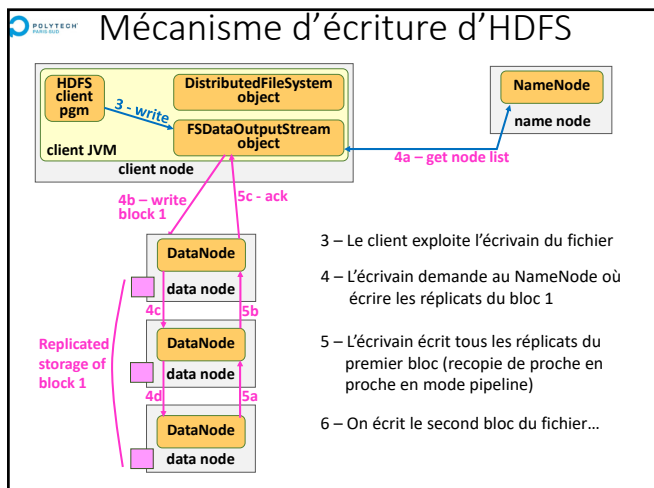
1. Localité des données et des traitements
2. Framework d'Hadoop
3. Mécanismes du Map-Reduce d'Hadoop
- 4. Système de fichiers distribué d'Hadoop (HDFS)**
 - Distribution et réplication des fichiers sous HDFS
 - Lecture de fichiers sous HDFS
 - **Ecriture de fichiers sous HDFS**
5. Allocation et gestion de ressources d'Hadoop

29

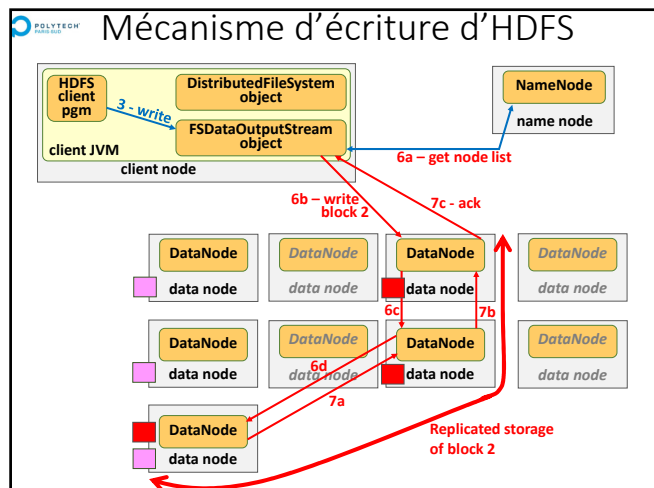
Mécanisme d'écriture d'HDFS

- 0 – Création d'un objet permettant de s'interfacier à HDFS (un stub/proxy d'HDFS)
- 1 – Demande d'ouverture d'un fichier HDFS en écriture (« create »)
- 2 – Demande de localisation des nœuds d'accueil des futurs blocs du fichier
 - 2a - Le proxy interroge le NameNode : pour savoir où écrire les blocs
Le NameNode répond au proxy
 - 2b - Le proxy crée et retourne un objet écrivain spécialisé sur le fichier ciblé

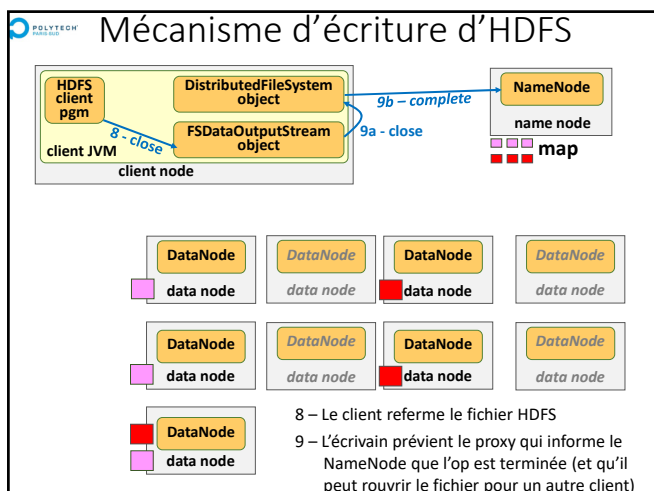
30



31



32

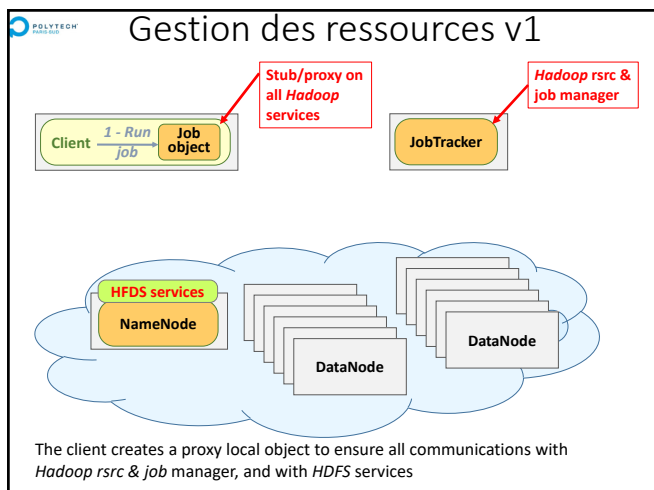


33

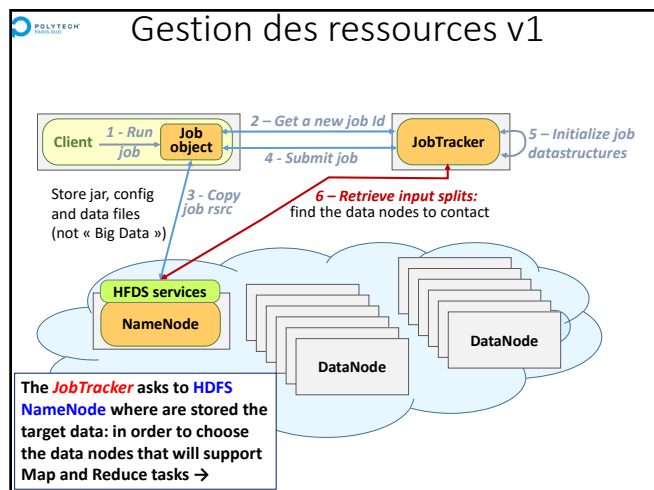
Principes et technologie d'Hadoop

1. Localité des données et des traitements
2. Framework d'Hadoop
3. Mécanismes du Map-Reduce d'Hadoop
4. Système de fichiers distribué d'Hadoop (HDFS)
5. Allocation et gestion de ressources d'Hadoop
 - Hadoop v1
 - Hadoop v2 (YARN) : passage à l'échelle amélioré

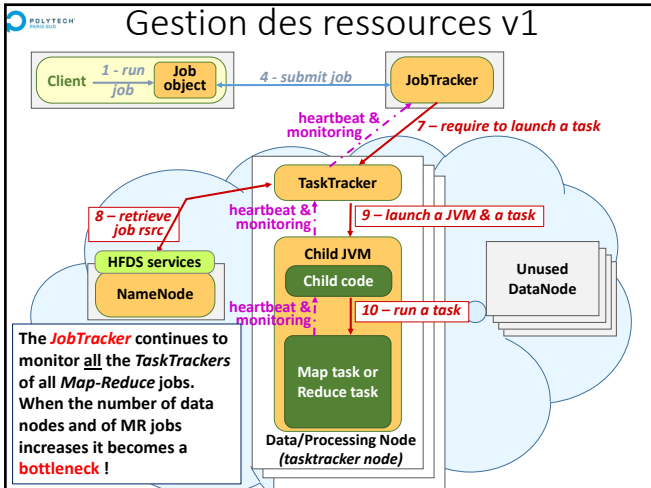
34



35



36

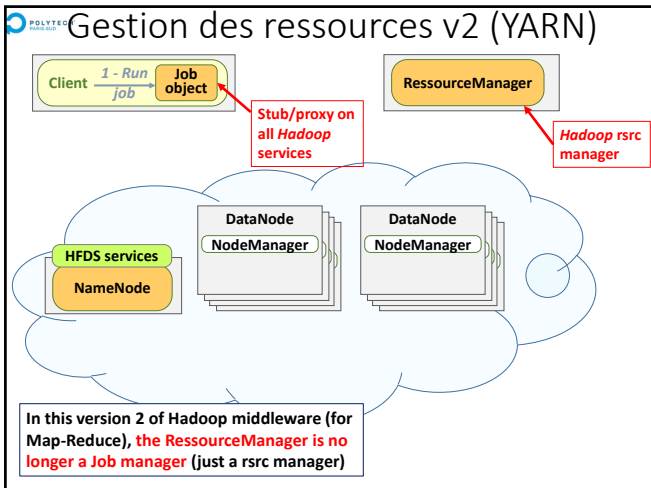


37

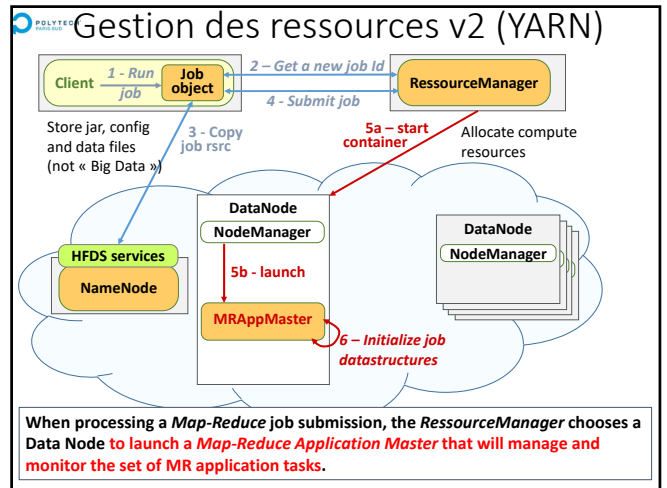
Principes et technologie d'Hadoop

1. Localité des données et des traitements
2. Framework d'Hadoop
3. Mécanismes du Map-Reduce d'Hadoop
4. Système de fichiers distribué d'Hadoop (HDFS)
5. Allocation et gestion de ressources d'Hadoop
 - Hadoop v1
 - Hadoop v2 (YARN) : passage à l'échelle amélioré

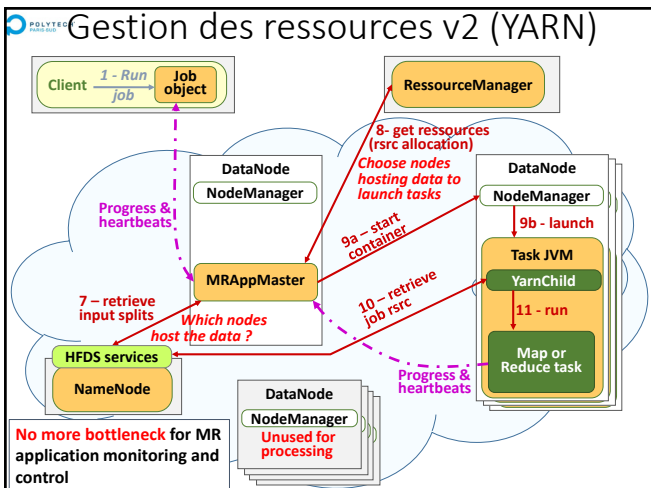
38



39



40



41

Exécution spéculative

Hadoop lance de nombreuses tâches (map, reduce, ...). Certaines peuvent « planter » ou « ralentir ».

Le TaskTracker (MR v1) ou l'ApplicationManager (MR v2) monitorre fortement les exécutions des tâches, et détectent ces « ralentissements ».

Hadoop peut faire une exécution spéculative : il lance de nouvelles instances des tâches « en retard », et dès qu'il obtient des résultats d'une tâche, il tue son doublon.

Mais cette démarche a un coût :

- en charge de calcul (création fréquentes de tâches redondantes)
- en déplacement de données (surtout si on redonne un reducer)

On peut débrayer ce comportement (ex : cluster HPC très fiable)

42

POLYTECH
UNIVERSITÄT
LIMBURG

QUIZ

Q1: a Hadoop Map-Reduce program ends up generating a huge number of key-value pairs with (always) the same key

- Will the HDFS output file be stored in one large block or in several small ones?
- Will the “reduce” treatment be processed in parallel or sequentially?

43

POLYTECH
UNIVERSITÄT
LIMBURG

QUIZ

Q2: a user connects his client program, running on his laptop, to a 100-node Hadoop cluster, and submits Map-Reduce queries, to compute the histogram of the age of the French (with one-year increments)

- Technically, can he download the results to his laptop?
- Technically, can he upload new input data to the HDFS of the 100-node cluster?
- Technically, can he download the input data to his laptop and then load it into a second Hadoop cluster?
- Is it possible to copy data from the HDFS of a first Hadoop cluster directly to the HDFS of a second?

44

POLYTECH
UNIVERSITÄT
LIMBURG

QUIZ

Q3: a failure occurs on a Hadoop data node used during the execution of a Map-Reduce program (the node disappears)

- Does the user have to resubmit the Map-Reduce request?
- Does the user get the result later when a failure occurs?


Q4: to improve fault tolerance, you can install HDFS on top of a RAID-enabled storage array (*Redundant Array of Independent Disks*)

- Do you think this is a logical approach?

45

POLYTECH
UNIVERSITÄT
LIMBURG

Technologies Internes d'Hadoop



46