

Chapitre II

Principes fondamentaux de Hadoop MapReduce

I. Système de fichiers distribué

Qu'est-ce qu'un système de fichiers distribués ?

- Un système de fichiers distribués (DFS, distributed file system) est un système de fichiers qui s'étend sur plusieurs serveurs de fichiers ou plusieurs sites, comme par exemple des serveurs de fichiers situés dans différents lieux physiques. Les fichiers sont accessibles à partir de n'importe quel appareil et de n'importe quel endroit du réseau, comme s'ils étaient stockés localement. Un DFS permet aux utilisateurs d'un réseau de partager des informations et des fichiers de manière contrôlée et autorisée.

I. Système de fichiers distribué

Qu'est-ce qu'un système de fichiers distribués ?

- Un système de fichiers distribué, ou DFS, est un système de stockage et de gestion des données qui permet aux utilisateurs ou aux applications d'accéder à des fichiers de données tels que des fichiers PDF, des documents Word, des images, des fichiers vidéo, des fichiers audio, etc., à partir d'un stockage partagé sur l'un des multiples serveurs en réseau. Avec des données partagées et stockées sur un cluster de serveurs, un DFS permet à de nombreux utilisateurs de partager des ressources de stockage et des fichiers de données sur de nombreuses machines.

I. Système de fichiers distribué

Les systèmes de fichier distribués les plus répandus sont :

- Ceph (<https://docs.ceph.com/en/quincy/>) est une plateforme libre de stockage distribué.
- GlusterFS (<https://docs.gluster.org/en/latest/>) est un système de fichiers réseau évolutif adapté aux tâches “data-intensive” telles que le stockage en nuage et le streaming multimédia.
- Google File System (GFS) est un système de fichiers distribué propriétaire. Il est développé par Google pour leurs propres applications
- Hadoop Distributed File System (HDFS) est un système de fichiers distribué, extensible et portable développé par Hadoop à partir du GoogleFS.

I. Système de fichiers distribué

Quand un système de fichiers distribués est-il essentiel ?

- Pour stocker des données de manière permanente.
- Pour partager facilement, efficacement et en toute sécurité des informations entre les utilisateurs et les applications.

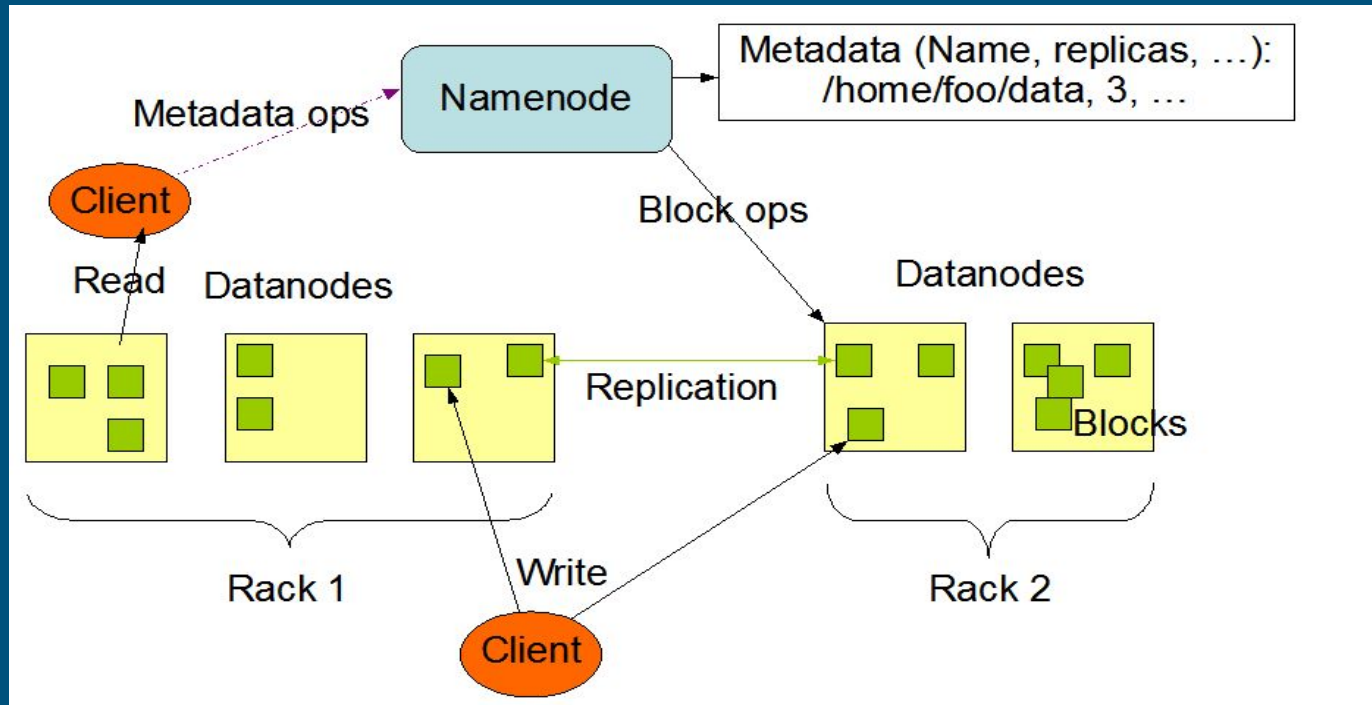
I. Système de fichiers distribué

Avantages d'un système de fichiers distribués :

- Tolérance aux pannes
- Un accès local transparent
- Une indépendance par rapport à l'emplacement
- De capacités scale-out : pouvoir monter en charge massivement de façon parallèle en ajoutant plus de machines. Les systèmes DFS peuvent évoluer vers des clusters très importants qui comptent des milliers de serveurs.

I. Système de fichiers distribué

HDFS architecture



I. Système de fichiers distribué

Comment fonctionne un système de fichiers distribués ?

- Distribution : le DFS distribue d'abord les jeux de données sur plusieurs clusters ou nœuds. Chaque nœud fournit sa propre puissance de calcul, ce qui permet au DFS de traiter les jeux de données en parallèle.
- Réplication : le DFS répliquera également les jeux de données sur différents clusters en copiant les mêmes informations sur plusieurs clusters. Cela rend le système de fichiers distribués tolérant aux pannes

II. Modèle de programmation MapReduce

Idée/Principe :

- Pour exécuter un problème large (Big Data) de manière distribué, il faut pouvoir découper le problème en plusieurs problèmes de taille réduite à exécuter sur chaque machine du cluster
- De multiples approches existent et ont existé pour cette division d'un problème en plusieurs

MapReduce :

- MapReduce inventé par Google, un cadre/modèle de traitement extrêmement parallèle adapté au traitement de très grandes quantités de données

II. Modèle de programmation MapReduce

MapReduce définit deux opérations distinctes à effectuer sur les données d'entrée:

- La première, MAP, va transformer les données d'entrée en une série de couples clé/valeur. Elle va regrouper les données en les associant à des clefs, choisies de telle sorte que les couples clé/valeur aient un sens par rapport au problème à résoudre. Par ailleurs, cette opération doit être parallélisable: on doit pouvoir découper les données d'entrée en plusieurs fragments, et faire exécuter l'opération MAP à chaque machine du cluster sur un fragment distinct.

II. Modèle de programmation MapReduce

MapReduce définit deux opérations distinctes à effectuer sur les données d'entrée:

- La seconde, REDUCE, va appliquer un traitement à toutes les valeurs de chacune des clés distinctes produite par l'opération MAP. Au terme de l'opération REDUCE, on aura un résultat pour chacune des clés distinctes. Ici, on attribuera à chacune des machines du cluster une des clefs uniques produites par MAP, en lui donnant la liste des valeurs associées à la clef. Chacune des machines effectuera alors l'opération REDUCE pour cette clef.

II. Modèle de programmation MapReduce

On distingue donc 4 étapes distinctes dans un traitement MapReduce:

- Découper (split) les données d'entrée en plusieurs fragments.
- Mapper chacun de ces fragments pour obtenir des couples (clé; valeur).
- Grouper (shuffle) ces couples (clé; valeur) par clef.
- Réduire (reduce) les groupes indexés par clé en une forme finale, avec une valeur pour chacune des clefs distinctes.

II. Modèle de programmation MapReduce

Exemple1 :

- **Objectif** : Imaginons qu'on nous donne un texte écrit en langue Française. On souhaite déterminer pour un travail de recherche quels sont les mots les plus utilisés au sein de ce texte
- **Texte**:

“ Que la paix soit sur toi, que la paix soit sur nous,Que la paix vienne à toi, que la paix vienne à nous,Que la paix soit sur celui qui nous la souhaite “

II. Modèle de programmation MapReduce

Exemple1 :

- **Première étape:** déterminer une manière de découper (split) les données d'entrée pour que chacune des machines puisse travailler sur une partie du texte.

→ On peut par exemple décider de découper les données d'entrée ligne par ligne

que la paix soit sur toi

que la paix soit sur nous

que la paix vienne à toi

que la paix vienne à nous

II. Modèle de programmation MapReduce

Exemple1 :

- **Première étape:** on obtient 4 fragments depuis nos données d'entrée.

que la paix soit sur toi

que la paix soit sur nous

que la paix vienne à toi

que la paix vienne à nous

II. Modèle de programmation MapReduce

Exemple1 :

- **Deuxième étape:** Mapper chacun de ces fragments pour obtenir des couples.
 - On doit désormais déterminer la clé à utiliser pour notre opération MAP, et écrire le code de l'opération MAP elle-même
 - Puisqu'on s'intéresse aux occurrences des mots dans le texte, et qu'à terme on aura après l'opération REDUCE un résultat pour chacune des clefs distinctes, la clé qui s'impose logiquement dans notre cas est: le mot-lui-même.
 - Générer le couple clef/valeur: (MOT :1)

II. Modèle de programmation MapReduce

Exemple1 : Deuxième étape:

POUR MOT dans LIGNE, FAIRE:
GENERER COUPLE (MOT: 1)

que la paix soit sur toi	(que:1)(la:1)(paix:1)(soit:1)(sur:1)(toi:1)
que la paix soit sur nous	(que:1)(la:1)(paix:1)(soit:1)(sur:1)(nous:1)
que la paix vienne à toi	(que:1)(la:1)(paix:1)(vienn:1)(à:1)(toi:1)
que la paix vienne à nous	(que:1)(la:1)(paix:1)(vienn:1)(à:1)(nous:1)

II. Modèle de programmation MapReduce

Exemple1 : Troisième étape

→ Grouper ces couples par clé; (COUPLE(s) par GROUPE)

(que:1)(que:1)(que:1) (que:1)	(soit:1)(soit:1)	(à:1) (à:1)
(la:1)(la:1)(la:1)(la:1)	(vienne:1)(vienne:1)	(toi:1) (toi:1)
(paix:1)(paix:1)(paix:1) (paix:1)	(sur:1) (sur:1)	(nous:1) (nous:1)

II. Modèle de programmation MapReduce

Exemple1 : Quatrième étape

→ Il nous reste à créer notre opération REDUCE, qui sera appelée pour chacun des groupes/clé distincte, elle va simplement consister à additionner toutes les valeurs liées à la clé spécifiée

```
nbr = 0
POUR COUPLE dans GROUPE, FAIRE:
    nbr = nbr + 1 #valeur
RENVoyer nbr
```

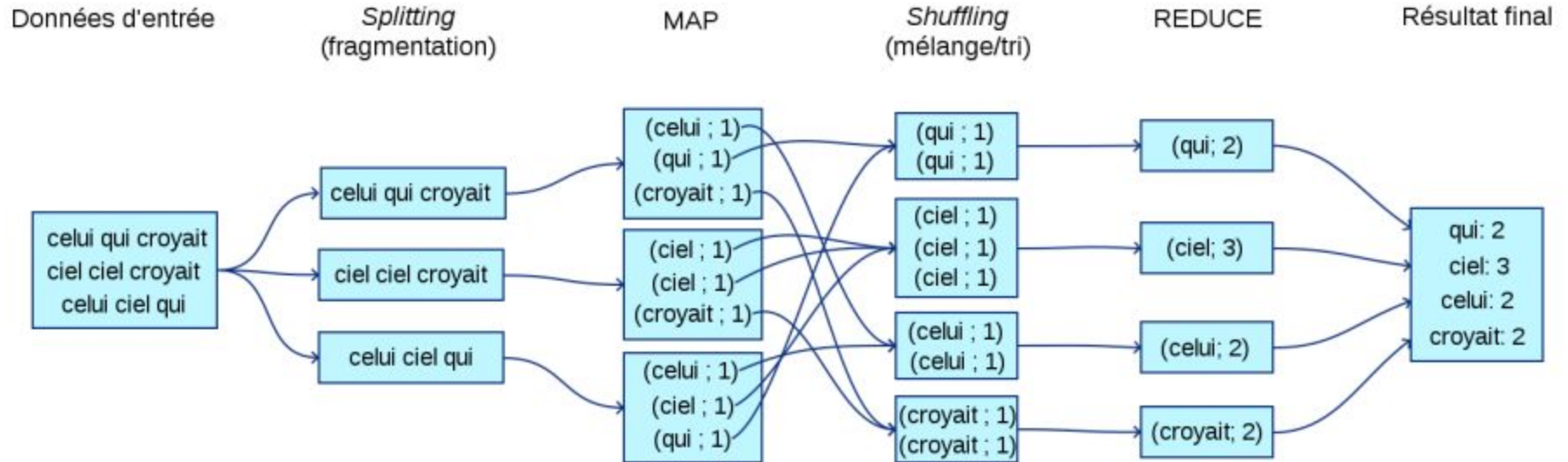
II. Modèle de programmation MapReduce

Exemple1 : Quatrième étape

que → 4
la → 4
paix → 4
soit → 2
...

II. Modèle de programmation MapReduce

Exemple 2 :



II. Modèle de programmation MapReduce

Exemple 3:

- On administre un réseau social comportant des millions d'utilisateurs.
- Pour chaque utilisateur, on a dans notre base de données la liste de ses amis.
- On souhaite afficher les amis en commun quand un utilisateur va sur la page d'un autre utilisateur .

TAF : Donner les programmes MAP et REDUCE pour atteindre cet objectif.

Fin

Merci Pour Votre Attention