Map Reduce

OUMAIMA
ROUAHI

CHAIMA
SLITI

PLAN

1. Introduction
2. Principe de Map-Reduce
3. Fonctionnement de MapReduce / Exemple d’utilisation
4. Les domaines d'application de MapReduce
5. Les limites
6. Conclusion

Sur le marché actuel, qui est piloté par les données, les algorithmes et les applications collectent des données sur les personnes, les processus, les systèmes et les entreprises, ce qui entraîne la génération de volumes de données considérables. Le défi consiste à définir la solution qui permettra de traiter ces volumes de données rapidement, efficacement et sans pertes de connaissance significatives.

C'est là qu'intervient le modèle de programmation MapReduce. Ce mécanisme a été mis en avant par Google en 2004 et a connu un très grand succès auprès des sociétés utilisant des DataCenters telles que Facebook ou Amazon.

Pourquoi MapReduce ?

Le traitement de données de façon distribuée soulève certaines questions:

- Comment distribuer le travail entre les serveurs ?
- Comment synchroniser les différents résultats ?
- Comment gérer une panne d'une unité de traitement ?

MapReduce répond à ces problèmes.

Il permet d'effectuer des calculs de façon parallèles, et souvent distribués, de données potentiellement très volumineuses, typiquement supérieures à 1 téraoctet.

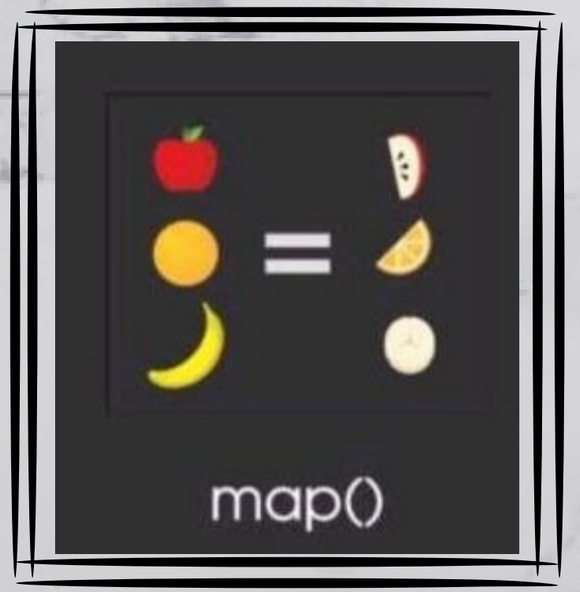
Principe de Map-Reduce Version 1

Le principe de Map-Reduce est simple:

Au cœur de MapReduce se trouvent deux fonctions, Map et Reduce, qui sont successivement séquencées.

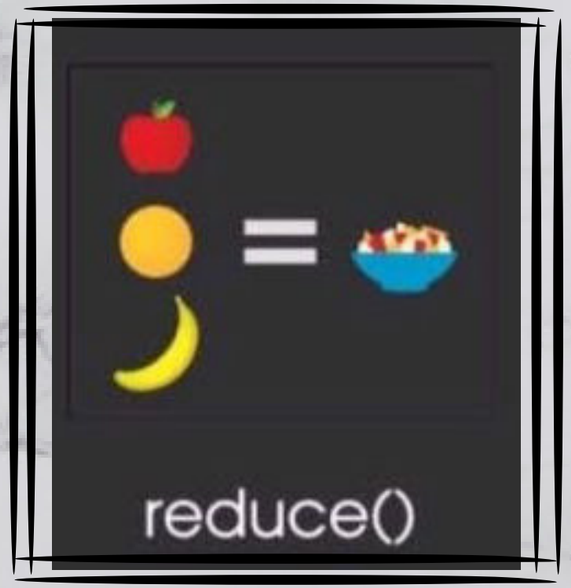
- La fonction Map transforme les entrées du disque en paires <key,value>, les traite et génère un autre ensemble de paires <key,value> intermédiaires en sortie.

->Le but est de partir d'un couple <clé, valeur> et d'y associé de nouveaux couples <clé, valeur>.



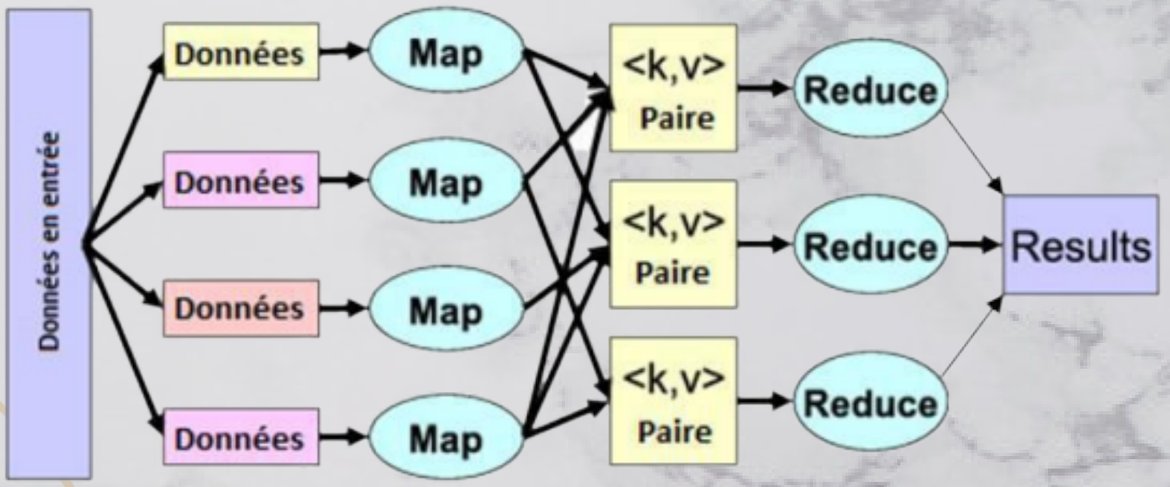
- La fonction Reduce transforme également les entrées en paires <key,value> et génère une des paires <key,value> en sortie.

-> Le but est d'associer toutes les valeurs correspondantes à la même clé



Fonctionnement de MapReduce

Les données franchissent les étapes suivantes de MapReduce en Big Data.



1) Prétraitement des données : (décompression des fichier).

2) Split : séparation des données en blocs et mis sous forme clé valeurs .

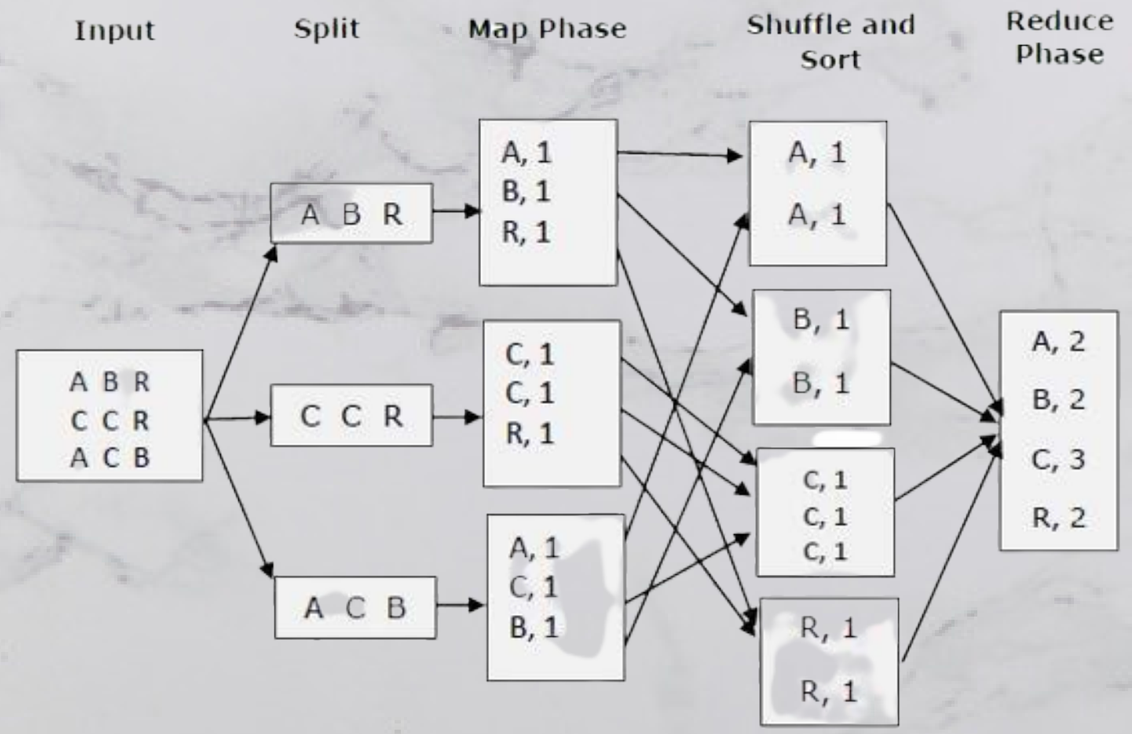
3) Map : application de la fonction map sur toutes parties clé valeur .

4)Shuffle et sort : rétribution des données afin que les paires produit par map ayant les mêmes clés soient sur les memes machines .

5) Reduce: agrégation des paires ayant la meme clé pour obtenir le résultat final .

Exemple d'utilisation

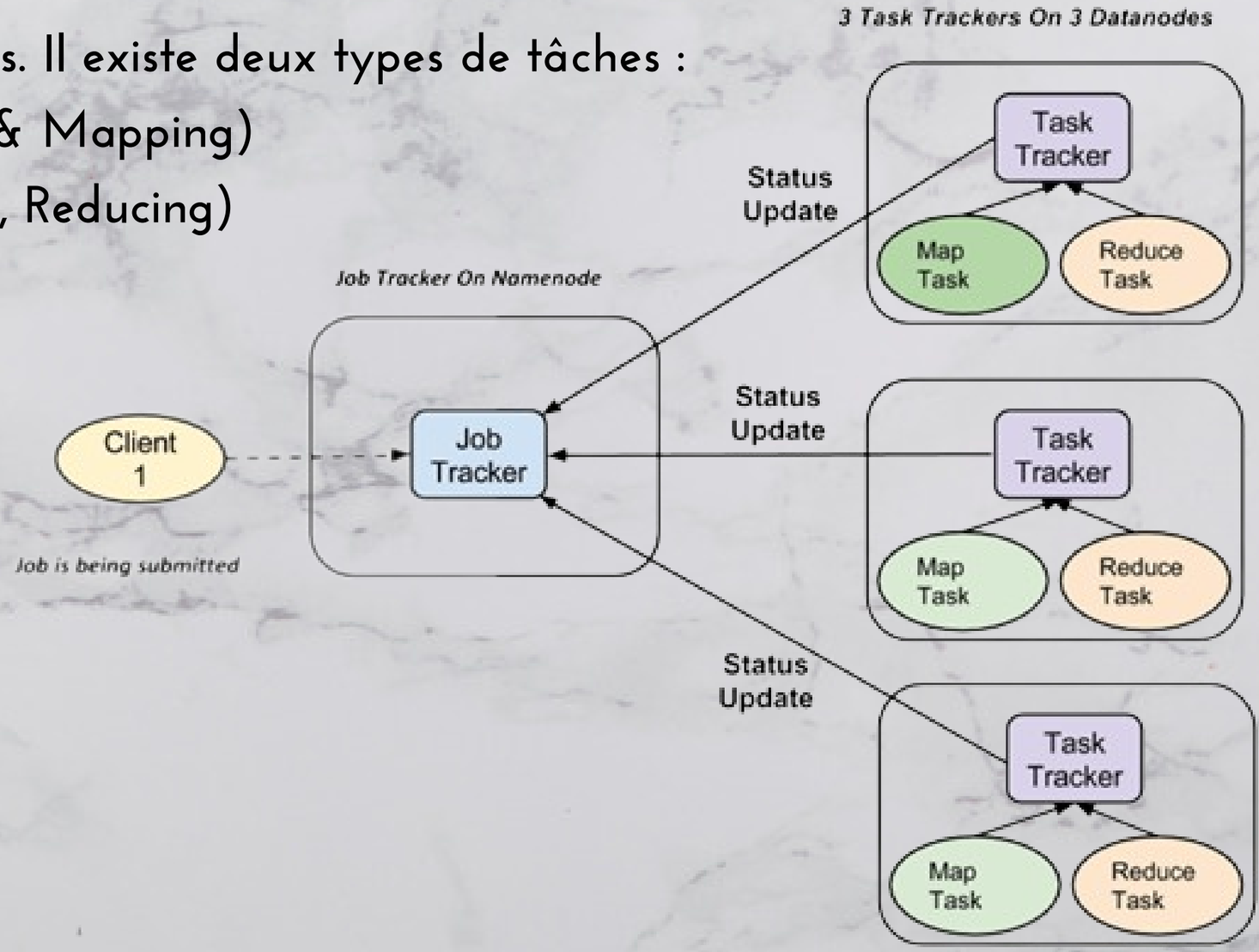
Le problème ici sera de compter le nombre d'occurrence de chaque lettre . Comme données en entrée nous allons avoir des fichiers texte. La fonction Map aura pour but de décomposer les lettres en couple de caractères clé-valeur. La fonction Reduce va prendre les couples avec la même clé et compter la totalité des occurrences pour ne faire qu'une seul paire clé-valeur par lettre.



Fonctionnement de MapReduce avec Hadoop

Hadoop divise le travail en tâches. Il existe deux types de tâches :

1. Tâches de mappage (Splits & Mapping)
2. Réduire les tâches (Shuffling, Reducing)



Le processus d'exécution complet est contrôlé par deux types d'entités appelées :

- Job tracker : Agit comme un maître

-> divise le travail sur les mapper et reducers , s'executant sur les différents noeuds . .

- Task Trackers : Agit comme des esclaves

-> s'execute sur chacun des noeuds pour appliquer les vraies taches de map reduce

Job

Une execution du programme MapReduce sur un ensemble de données

Task

Execution d'un mapper ou d'un reducer sur une tranche de données

- L'utilisateur soumet une tâche MapReduce au système Hadoop.
- Le Job Tracker alloue des tâches Map et Reduce aux Task Trackers disponibles pour le traitement des données.

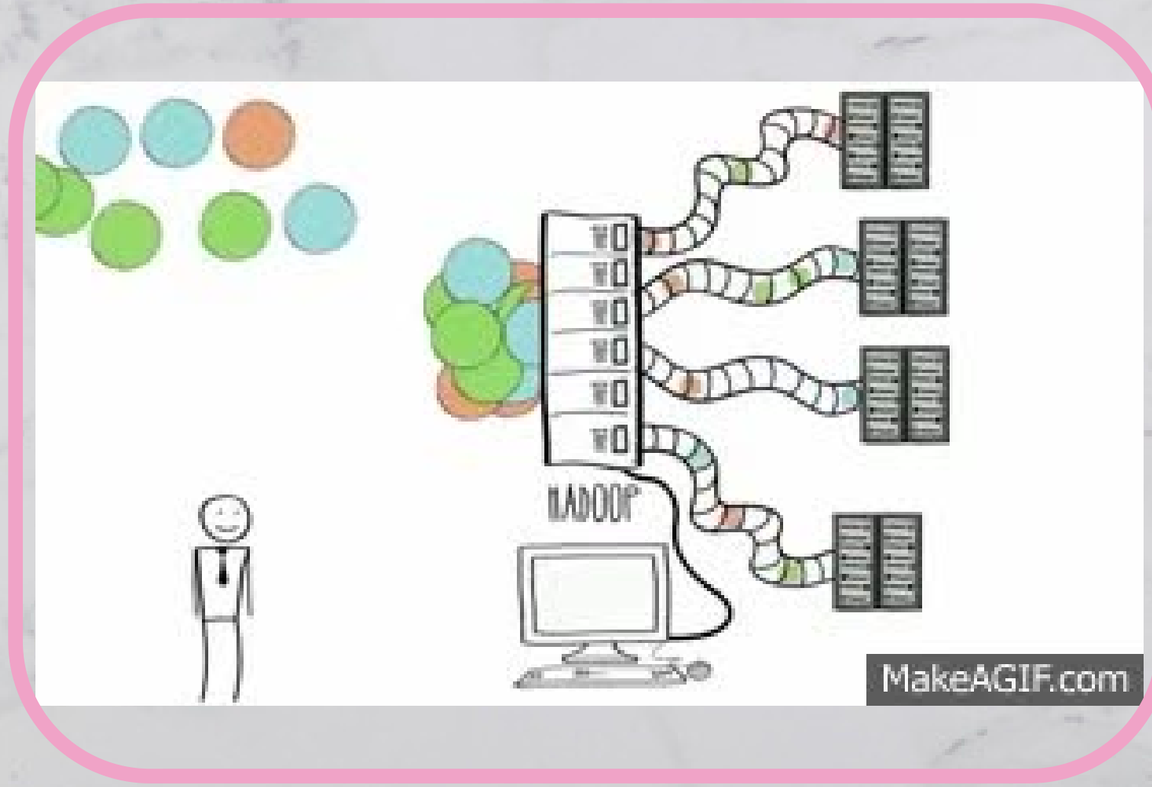
-> Les Task Trackers exécutent ces tâches Map et Reduce en parallèle sur les données qui leur sont assignées.

- Pendant l'exécution des tâches Map, chaque Task Tracker utilise la fonction Map personnalisée pour produire une nouvelle paire clé-valeur. Ces paires clé-valeur sont stockées en mémoire sur le nœud Task Tracker.

Une fois que toutes les tâches Map sont terminées, le Job Tracker commence à planifier les tâches Reduce en fonction des paires clé-valeur générées par les tâches Map. Les paires clé-valeur sont triées et regroupées par clé avant d'être envoyées aux nœuds Reduce correspondants.

- Les tâches Reduce utilisent une fonction Reduce personnalisée pour agréger les valeurs associées à chaque clé unique. Les résultats agrégés sont stockés en mémoire sur le nœud Task Tracker.

- Une fois que toutes les tâches Reduce sont terminées , les résultats finaux sont écrits dans le système de fichiers Hadoop et sont disponibles pour être utilisés par d'autres applications.



Exemple d'utilisation

le traitement de données de logs Web.

"Les entreprises peuvent générer des millions de lignes de logs chaque jour à partir de leurs sites Web, ce qui peut nécessiter des analyses pour améliorer l'expérience utilisateur ou détecter les problèmes de sécurité."

1. Division des données : Les fichiers de logs Web sont divisés en blocs plus petits, chacun de taille par défaut de 128 Mo, pour faciliter le traitement.

2. Étape Map : Les nœuds Map exécutent une fonction Map personnalisée sur chaque ligne de log pour extraire les informations clés telles que l'adresse IP du client, le type de requête, le code de réponse et d'autres informations pertinentes. Ces informations sont transformées en paires clé-valeur, par exemple, la clé pourrait être l'adresse IP du client et la valeur pourrait être le code de réponse.

3. Shuffle et sort : Les paires clé-valeur produites par les nœuds Map sont triées et regroupées par clé. Cette étape regroupe toutes les paires clé-valeur avec la même clé et les envoie au nœud Reduce.

4. Étape Reduce : Les nœuds Reduce agrègent les informations pour chaque clé unique, par exemple, en comptant le nombre de requêtes réussies et échouées pour chaque adresse IP. Les résultats agrégés sont stockés en mémoire sur le nœud Task Tracker.

5. Écriture des résultats : Les résultats agrégés sont écrits dans le système de fichiers Hadoop, généralement dans HDFS.

6. Fin de l'exécution : Une fois que tous les nœuds Reduce ont terminé leur travail, le programme MapReduce se termine et le résultat final est disponible dans le système de fichiers Hadoop.

Voici un exemple de code Python utilisant mrjob pour effectuer un MapReduce sur des fichiers de logs afin de compter le nombre de vues de chaque page d'un site Web :

```
from mrjob.job import MRJob
import re

class MRPageViews(MRJob):

def mapper(self, _, line):
# extraire l'URL de la page à partir de la ligne de log
url = re.search(r'"GET (.*?) HTTP', line).group(1)
yield (url, 1)

def reducer(self, url, counts):
yield (url, sum(counts))

if __name__ == '__main__':
MRPageViews.run()
```

- La méthode mapper extrait l'URL de la page à partir de chaque ligne de log et renvoie un tuple (url, 1) pour chaque page visitée.
- La méthode reducer agrège les tuples en comptant le nombre total de vues pour chaque page.

Pour exécuter ce job MapReduce sur un cluster Hadoop, nous pouvons utiliser la commande suivante :

```
python page_views.py -r hadoop access_log.txt -o output
```

Où :

Page_views.py: le nom du fichier contenant le code ci-dessus.
Access_log.txt : le nom du fichier de logs d'entrée .
Output : le nom du répertoire de sortie.

Les domaines d'application de MapReduce :

Analyse de données : MapReduce est utilisé pour effectuer des analyses sur des ensembles de données massives, y compris l'exploration de données, l'analyse de texte, l'analyse de sentiment, la classification, la segmentation de données, et l'analyse de tendances.

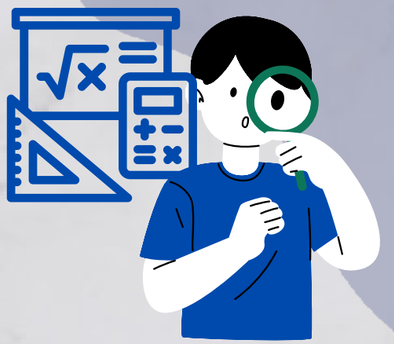
Exemple: Une entreprise de commerce électronique utilise MapReduce pour analyser les habitudes d'achat de ses clients, afin d'identifier les produits les plus populaires et les tendances d'achat. La société peut utiliser ces informations pour optimiser son inventaire, son marketing et ses offres promotionnelles.

Recherche d'informations : Les moteurs de recherche utilisent MapReduce pour effectuer des opérations de recherche et de traitement de requêtes en temps réel.

Exemple: Un moteur de recherche utilise MapReduce pour indexer des milliards de pages web et pour traiter des requêtes de recherche en temps réel. MapReduce permet au moteur de recherche de traiter les requêtes en parallèle sur plusieurs nœuds de traitement, ce qui permet d'accélérer les temps de réponse.

Traitement de données en temps réel : MapReduce peut être utilisé pour traiter des données en temps réel, par exemple dans les applications de surveillance de la santé ou de la finance.

Exemple: Une entreprise de services financiers utilise MapReduce pour surveiller les transactions financières en temps réel, afin de détecter les fraudes et les activités suspectes. MapReduce permet à l'entreprise de traiter les transactions en temps réel, même lorsque le volume de transactions est très élevé.

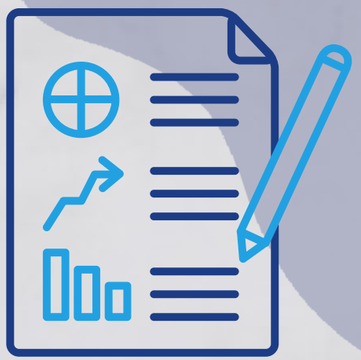


Calcul scientifique : MapReduce est utilisé pour des applications de calcul scientifique telles que la modélisation climatique, la modélisation de fluides et la simulation de molécules.

Exemple: Un centre de recherche utilise MapReduce pour effectuer des simulations de modélisation climatique. MapReduce permet au centre de recherche de traiter de grandes quantités de données météorologiques, de modéliser les effets des changements climatiques et de prédire les tendances à long terme.

Analyse de sécurité : Les applications de sécurité utilisent MapReduce pour analyser des ensembles de données massives afin de détecter les menaces et les activités suspectes.

Exemple: Une entreprise de cybersécurité utilise MapReduce pour analyser les données de journalisation de ses clients, afin de détecter les menaces et les activités suspectes. MapReduce permet à l'entreprise de traiter de grandes quantités de données de journalisation en temps réel, de générer des alertes de sécurité et de fournir des rapports d'analyse détaillés.



Traitement de données graphiques : Les applications de traitement de données graphiques, telles que la recommandation de produits, la recommandation de films et l'analyse de réseaux sociaux, utilisent MapReduce pour traiter des graphes massifs.

Exemple: Un site de commerce électronique utilise MapReduce pour recommander des produits à ses clients en fonction de leurs habitudes d'achat. MapReduce permet au site de commerce électronique de traiter de grandes quantités de données de transaction pour identifier les produits les plus populaires, les produits associés et les produits complémentaires.

Les limites

Le job tracker se charge de :

- Allouer les ressources aux différents taches .
- Coordonner l'execution des job Map reduce .
- Réserver et ordonner les slots , et gérer les fautes .

Un seul job tracker cause des problèmes :

1. Problème d'évolutivité
2. Problème d'utilisation des ressources.
3. Supporter que les applications map reduce .

Conclusion

MapReduce est une technologie largement utilisée dans le domaine du Big Data et a permis de démocratiser l'analyse de données à grande échelle en offrant une solution efficace et évolutive pour traiter les volumes de données massifs. Cependant, de nouvelles technologies de traitement de données en temps réel, telles que Apache Spark, ont émergé ces dernières années, offrant des performances encore meilleures pour certaines applications.

Références

https://www.talend.com/fr/resources/what-is-mapreduce/

https://stph.scenari-community.org/contribs/nos/Hadoop3/co/Deroulement_de_la_presentation.html

https://www.youtube.com/watch?v=nIT1wVTLVY4&t=3s

https://www.guru99.com/introduction-to-mapreduce.html

https://chat.openai.com/chat

Merci !