Année Universitaire 2024-2025

BIG DATA

Mhamdi Farouk

Big Data ?

- On entend beaucoup parler du Big Data dans tous les domaines de nos vies : nos navigations Web sont impactées par le Big Data, les réseaux sociaux, nos objets connectés (**75 milliards d'objets connectés avant 2025**)…
- Le big data, « grosses données » ou mégadonnées, parfois appelées données massives, désigne des ensembles de données devenus si volumineux qu'ils dépassent l'intuition et les capacités humaines d'analyse et même celles des outils informatiques classiques de gestion de base de données ou de l'information :

  - nous générons effectivement 2.5 trillions d’octets de données dans le monde (source : étude IBM, 2018).

  - les entreprises ont de plus en plus besoin de professionnels capables d’analyser ces données pour créer de la valeur.

  - le Big Data nécessite l’utilisation de technologies et de méthodes analytiques spécifiques.

Qu'est-ce qui a lancé l'ère du Big Data ?

Principalement, deux nouvelles opportunités qui ont contribué au lancement de l'ère du Big Data :

- Un torrent croissant de données :les données semblent venir continuellement et à un rythme rapide.

- Une capacité de stockage incroyable par rapport à toutes les formes précédentes de stockage et moins couteuse

Occasions d'effectuer des analyses de données nouvelles, dynamiques et évolutives,

Qu'est-ce qui a lancé l'ère du Big Data ?

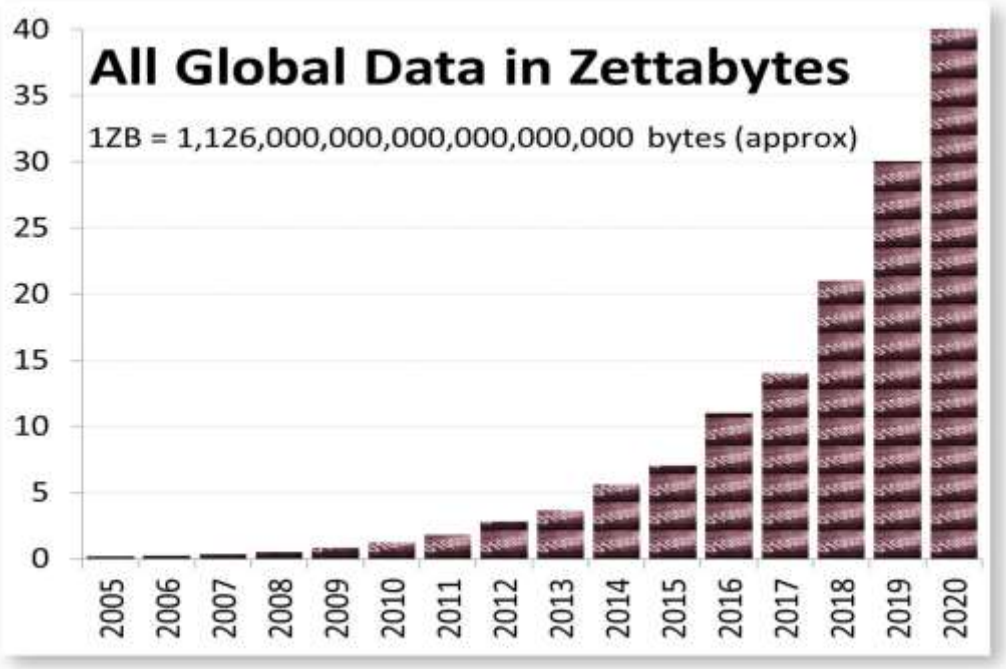
- Torrent croissant de données
- Continu
- Rythme rapide

- Une capacité de stockage incroyable
- Informatique parallèle, algorithmes distribués
- Bd NoSQL

Ère du Big Data

Contexte 1 : Torrent croissant de données, continu avec un rythme rapide (1/4)

- La quantités de données digitales produites double tous les 2 ans.
- En d'autres termes, on a produit autant de données digitales ces 2 dernières années que tout ce qui a été produit auparavant.
- Croissance de la quantité des données exponentielle



Contexte 1 : Torrent croissant de données, continu avec un rythme rapide (2/4)

Augmentation de 40 % de la quantité des données crées dans le monde (sur la période de 2013-2025)

Contexte 1 : Torrent croissant de données, continu avec un rythme rapide (3/4)

Contexte 1 : Torrent croissant de données, continu avec un rythme rapide (4/4)

Encore des chiffres :

compagnies	Données traitées par jour
eBay	100 PB
Google	100 PB
Facebook	30 PB
Twitter	100 TB
Spotify	64 TB

Année 2014

Question : Quelle est la quantité journalière des données analysées par Twitter pour faire des analyses de sentiments autour de leurs produits ?

Réponse : 12 TB

Les volumes à gérer sans précédents impliquent :

- Données **hétérogènes, complexes** et souvent liées
  - produites par des applications parfois différentes,
  - par des utilisateurs différents,
  - avec des liens explicites (par exemple citations, url, etc) ou implicites (à extraire)
- **Nombreux serveurs/clusters**
  - un serveur unique ne peut stocker cette quantité d'information, garantir des temps d'accès pour grand nombre d'utilisateur, faire des calculs rapides, etc
- Besoin de **distribuer les calculs et les données**
  - comme plusieurs **serveurs/clusters**, **besoin d'algorithmes permettant le calcul et la distribution** des données à large échelle

Data centers de quelques grands acteurs du Big Data :

- Google Data Center :
  - 70000 servers/data center et 16 data centers, ~1M de serveurs
- Facebook :
  - 5 data centers
- Amazon :
  - 7 data centers, 450 000 severs
- Microsoft :
  - environ 1M serveurs

Disciplines participant au Big Data

Le Big Data est un domaine pluridisciplinaire pour lequel on peut identifier 4 parties, parfois elles-mêmes basées sur plusieurs disciplines :

- Partie **Math-Info** comprend :

  - mathématiques statistiques et probabilistes sur lesquelles sont fondés des algorithmes d'apprentissage numérique (ou machine learning), ainsi que des algorithmes de fouille de données et de graphes.
  - Cette partie du Big Data est celle qui est souvent identifiée comme le Data science. C'est en tous cas le coeur mathématique du Big Data.

Disciplines participant au Big Data

- Partie **d'informatique distribué** pour l'analyse de données large échelle :
  - forme d'algorithmique distribuée récente (apparue en 2009), visant à amener les traitements sur les machines où sont stockées les données.
  - permet des traitements de données à large échelle (sur des données très volumineuses), voire à l'échelle du web (webscale).
  - Une première mise en oeuvre de cette approche utilisait le schéma Map-Reduce (schéma de calcul distribué)

Disciplines participant au Big Data

- Partie d'informatique parallèle à haute performance pour le data analytics et le machine learning :
  - visant à accélérer les calculs sur des machines parallèles
  - Par exemple, en utilisant un cluster de PC multi-coeurs (ensemble de PC dédiés aux calculs et reliés par un réseau local rapide) ou un cluster de GPU (réseau de cartes graphiques détournées pour du calcul scientifique), pour entraîner des réseaux de neurones profonds (deep learning).

Disciplines participant au Big Data

- Partie essentielle du Big Data :

  - réside dans la conception et l'exploitation de bases de données "not only SQL" (NoSQL).
  - stocker des données structurées complexes, ou au contraire de simples fichiers textes que l'on devra analyser en détail.
  - certaines BdD NoSQL ont été conçues pour un stockage distribué à très large échelle, d'autres pour favoriser la vitesse d'interrogation sur des données plus restreintes.

Exemples d'application du Big Data (1/4)

Parmi les applications du Big Data on peut citer trois grands classiques :

1. **Analyse de fichiers de traces pour en déduire des comportements passés et futurs :**
   - traces de transactions commerciales, traces d'activités de personnes mobiles, traces d'accès à des serveurs web. . ., et déductions de comportements de consommateurs, de voyageurs, d'utilisation d'Internet.
   - aboutit généralement à un système de recommandation pour mieux cibler des clients potentiels ou pour augmenter les performances de l'entreprise
   - Ce type d'analyse se faisait initialement off line sur de très gros volumes de données, mais on observe une demande croissante de traitement on line de flux continus de données.

Exemples d'application du Big Data (2/4)

Les traces que nous laissons sur Internet (volontairement ou non), vont constituer notre Identité Numérique

Réputation
Ce que l'on dis sur moi

Publication
Ce que je partage

Opinion
Ce que je pense

Préférences
Ce que j'aime

Réseau
Qui je connais

Coordonnées
Où et Comment me joindre

Achats
Ce que j'achète

Profession
Mon métier, où je travaille

Exemples d'application du Big Data (3/4)

2. **Analyse de signaux de sorties d'une foule de capteurs sur une installation industrielle**, couplée à l'accès à des bases de données de situations passées et de spécifications techniques,

- afin d'identifier les prémisses d'une défaillance future.

- doit en général se faire en temps réel à partir des signaux captés sur l'installation en cours d'utilisation

Exemples d’application du Big Data (4/4)

3. Analyse de réseaux sociaux, c’est-à-dire de **l’analyse de graphes,**

- afin d’en déduire par exemple des relations/influences entre individus ou populations d’individus.
- L’analyse de graphes de grandes tailles constitue une partie spécifique du Big Data souvent citée comme exemple, et qui intéresse particulièrement quelques géants du web.

Big Data : types de données

- **Données structurées** : organisées dans un référentiel formaté qui est généralement une base de données. Ce type concerne toutes les données pouvant être stockées dans une base de données SQL dans une table avec des lignes et des colonnes. Exemple : données relationnelles.

ID	Name	Age
1	Alex	22
2	Bob	24
3	Emily	32

- **Données semi-structurées** : sont des informations qui ne résident pas dans une base de données relationnelles, mais qui possèdent des propriétés organisationnelles facilitant leur analyse. Avec certains processus, vous pouvez les stocker dans la base de données relationnelle. Exemple: données XML

- **Données non structurées** : sont des données non organisées de manière prédéfinie ou qui ne possèdent pas de modèle de données prédéfini. Elles ne conviennent donc pas à une base de données relationnelle traditionnelle. Ilexiste des plates-formes alternatives pour le stockage et la gestion de ces de données. Exemple: PDF, texte.

Big Data : sources de données

Le volume de données numériques ne cesse d'augmenter. Cette prolifération des données est due à la numérisation croissante de tous les domaines du web et de l'économie. C'est dans ce contexte que le Big Data est né, au moyen de la fusion de diverses sources de données, structurées ou non structurées, telles que :

- L'utilisation d'internet sur les mobiles
- Les réseaux sociaux
- La géolocalisation
- Le cloud
- les données collectées des capteurs d'IOT
- Le streaming des médias

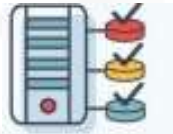
Caractéristiques de Big data

Le big data est caractérisé par les **5 V** suivants :

Le volume : correspond à la masse d'informations produite chaque seconde.

La vélocité :équivaut à la rapidité de l'élaboration et du déploiement des nouvelles données.

La variété : 20% des données sont structurées et 80% qui restent sont non-structurées (images, des vidéos, des textes, des voix, et bien d'autres encore…)



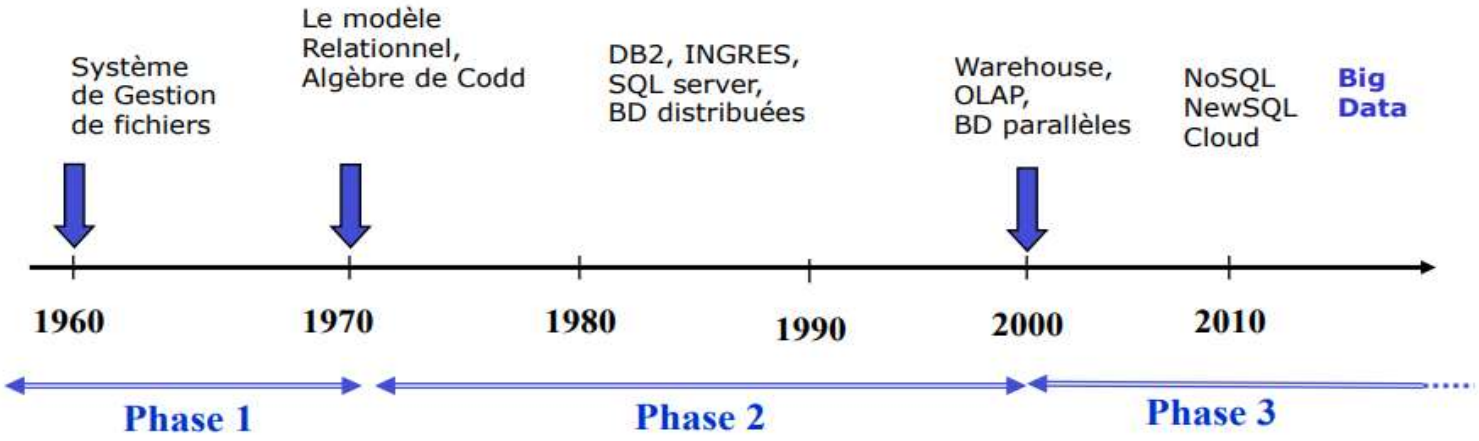
La véracité :concerne la fiabilité et la crédibilité des informations collectées.

La valeur : correspond au profit qu'on puisse tirer de l'usage du Big Data. Ce sont généralement les entreprises qui commencent à obtenir des avantages incroyables de leurs Big Data.

Les phases de la gestion des données

La gestion des données a passé par les phases schématisées par la figure, et qui sont :

- Phase 1 : période préhistorique 96-69
- Phase 2 : période phare 70- 2000
- Phase 3 : période nouvelle 2000-maintenant



Hadoop :

- ✓ Hadoop est un framework open source développé en java,
- ✓Hadoop est destiné à faciliter le développement d'applications distribuées et scalables, permettant la gestion de milliers de noeuds ainsi que des pétaoctets de données.
- ✓Hadoop s'inspire de deux produits, le premier est le « Google File System », un système de fichiers distribués. Le deuxième est le modèle de Programmation Map-Reduce,

HDFS (HadoopDistributed File System)

Dans Hadoop, les différents types de données sont stockées à l'aide du HDFS. Le HDFS va prendre les données en entrée et va ensuite les partitionner en plusieurs blocs de données. Par défaut les données sont répliquées sur trois noeuds différents pour rééquilibrer les données, deux sur le même support et un sur un support différent.

Typologies d’un cluster Hadoop

Hadoop repose sur un schéma dit « maître-esclave » et peut être décomposé en cinq éléments.

Le nom du noeud (Name Node) : Le « Name Node » est la pièce centrale dans le HDFS, il maintient une arborescence de tous les fichiers du système et gère l’espace de nommage.

Le gestionnaire de tâches (Job Tracker) : Il s’occupe de la coordination des tâches sur les différents clusters.

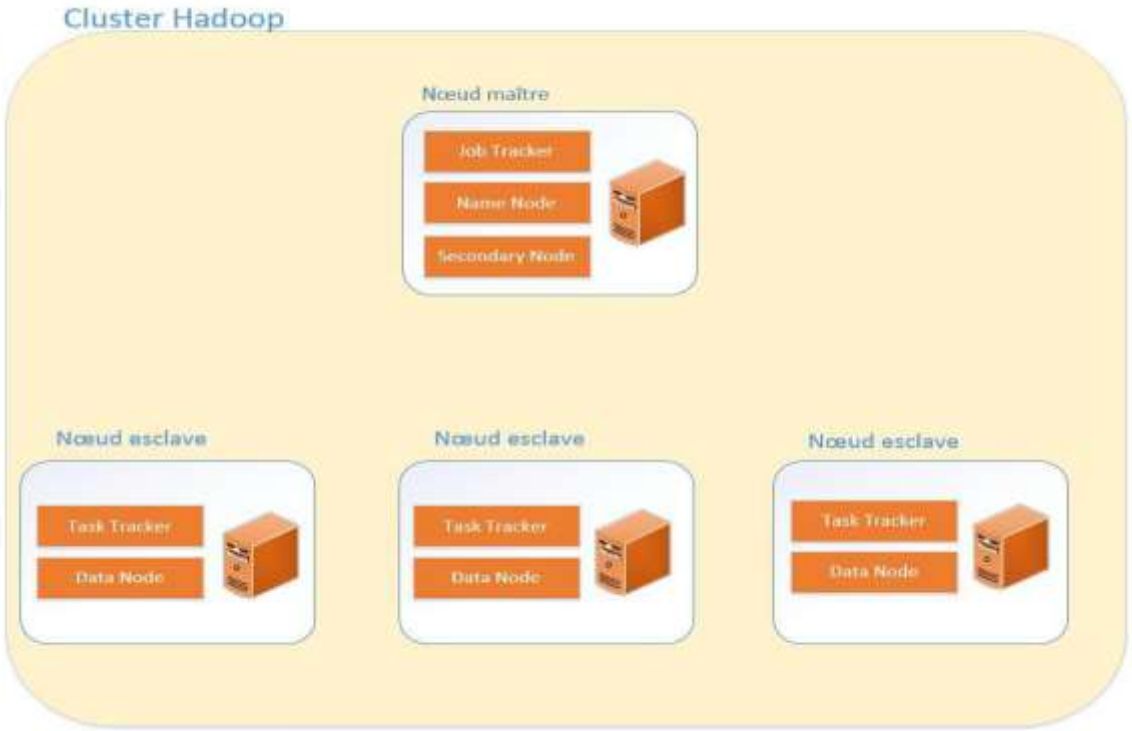
Le moniteur de tâches (Tasktracker) : Il permet l’exécution des ordres de map-Reduce,

Le noeud secondaire (Secondarynode) : faire périodiquement une copie des données du « Name Node » afin de pouvoir prendre la relève en cas de panne de ce dernier.

Le noeud de données (Data Node) : stockage des blocs de données.

Un cluster Hadoop

Un cluster Hadoop peut-être constitué de machines hétérogènes, que ce soit au niveau du hardware comme au niveau software (système d'exploitation). Cependant, il est bien plus simple d'administrer un cluster de type homogène.



Map-Reduce

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 ? Map-Reduce répond à ces problèmes.

Map-Reduce permet de traiter une grande quantité de données de manière parallèle, en les distribuant sur divers noeuds d'un cluster.

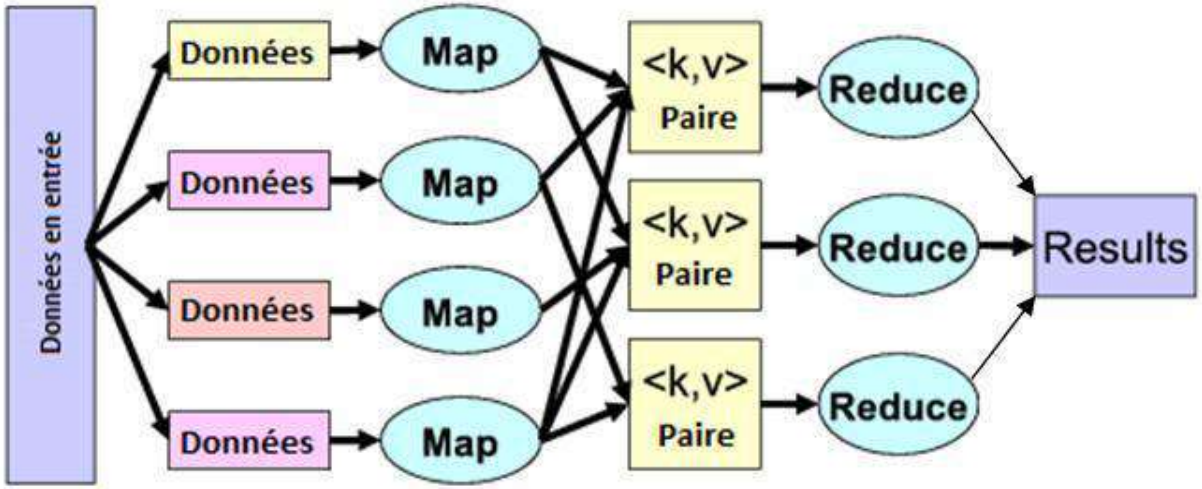
Le principe de Map-Reduce est simple: il s'agit de découper une tâche manipulant un gros volume de données en plusieurs tâches traitant chacune un sous-ensemble de ces données.

Le fonctionnement de MapReduce

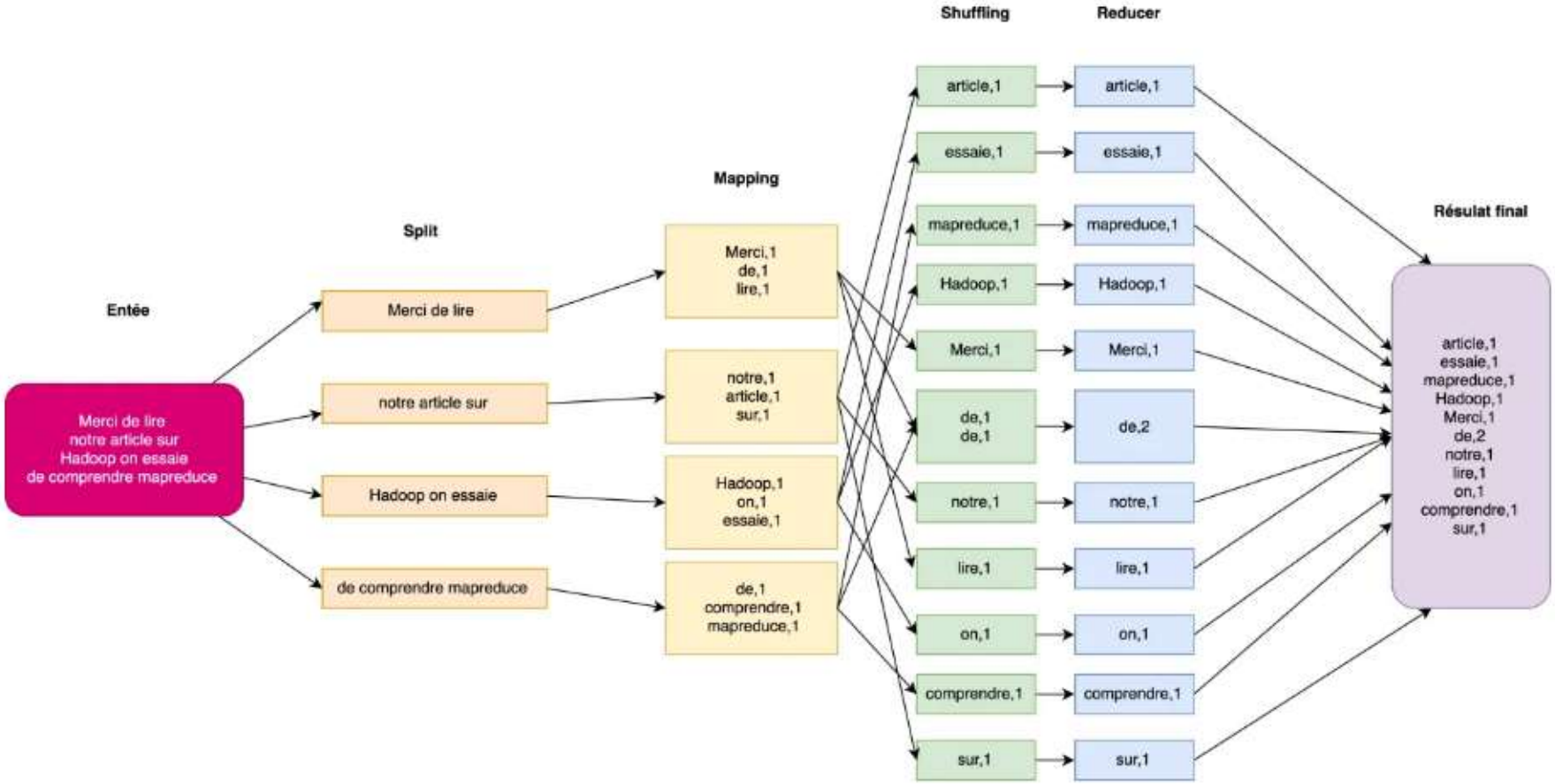
Le modèle MapReduce consiste en deux étapes, représentées toutes deux par des fonctions :

La fonction Map : faire une analyse du problème et le séparer en sous-tâches, pour pouvoir les redistribuer sur les noeuds du cluster.

La fonction Reduce : faire remonter tous les résultats à leur noeud parents respectif.



Exemple d'application de MapReduce



L’écosystème de Hadoop

Il y a beaucoup de de solutions qui gravitent autour de Hadoop et qui forme son écosystème complexe. Nous illustrons une liste non-exhaustive des principales solutions.

Hadoop a été crée par Yahoo en 2005

Plus de Framework Big Data ajoutés

kafka

Apache Solr

AVRO

YARN

MESOS

Spark

mahout

HIVE

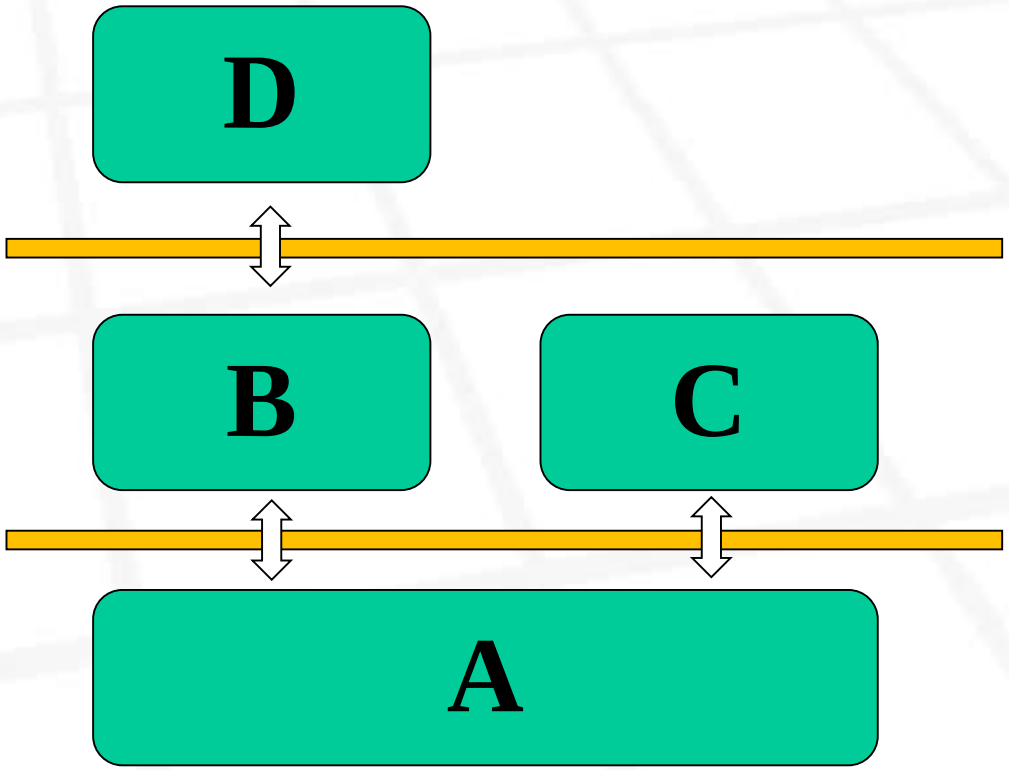
APACHE HBASE

APACHE GIRAPH

hadoop

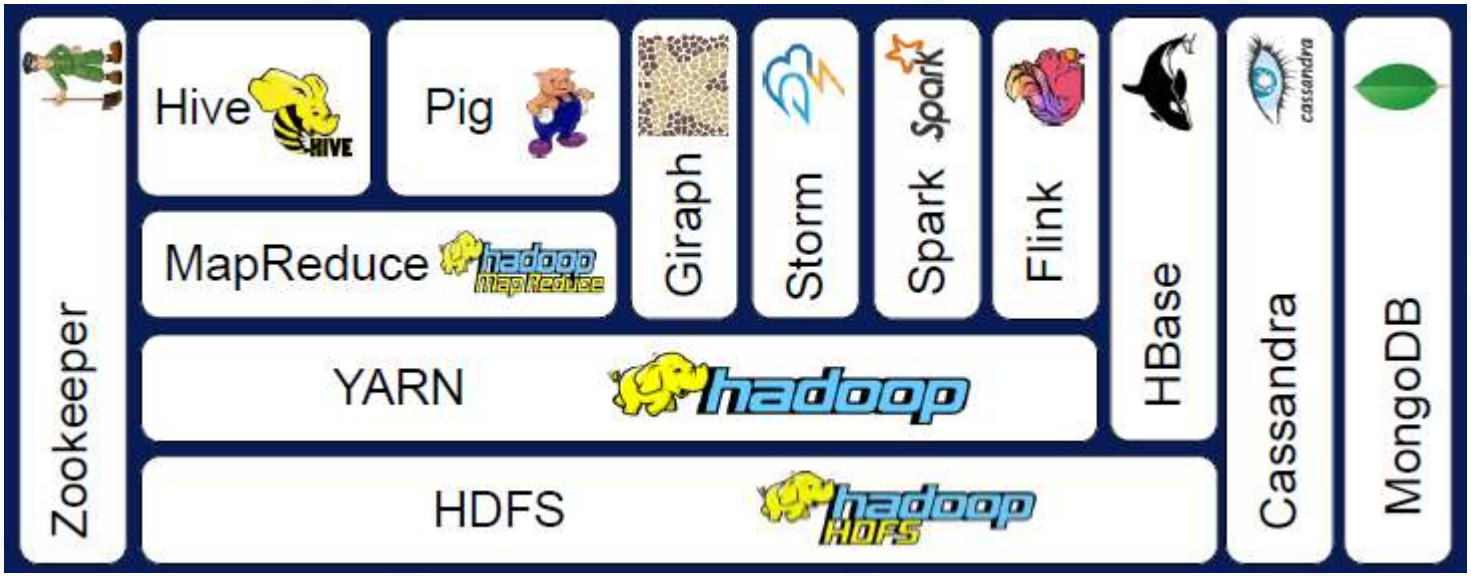
cassandra

Diagramme de couches



Un Diagramme de couches possible pour Hadoop

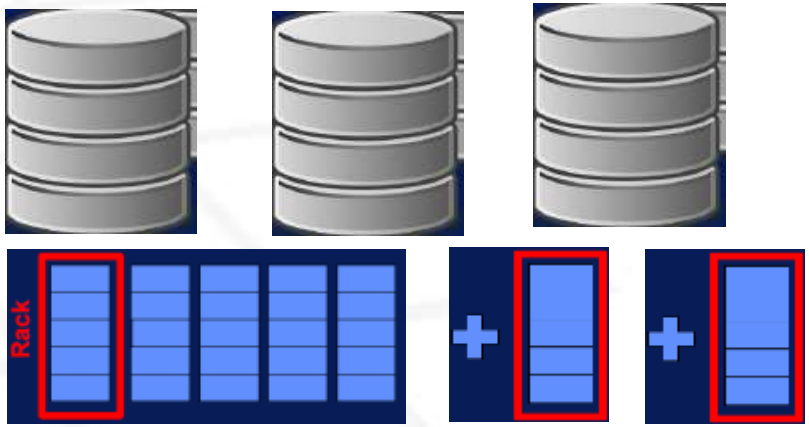
Niveaux plus hauts: interactivité

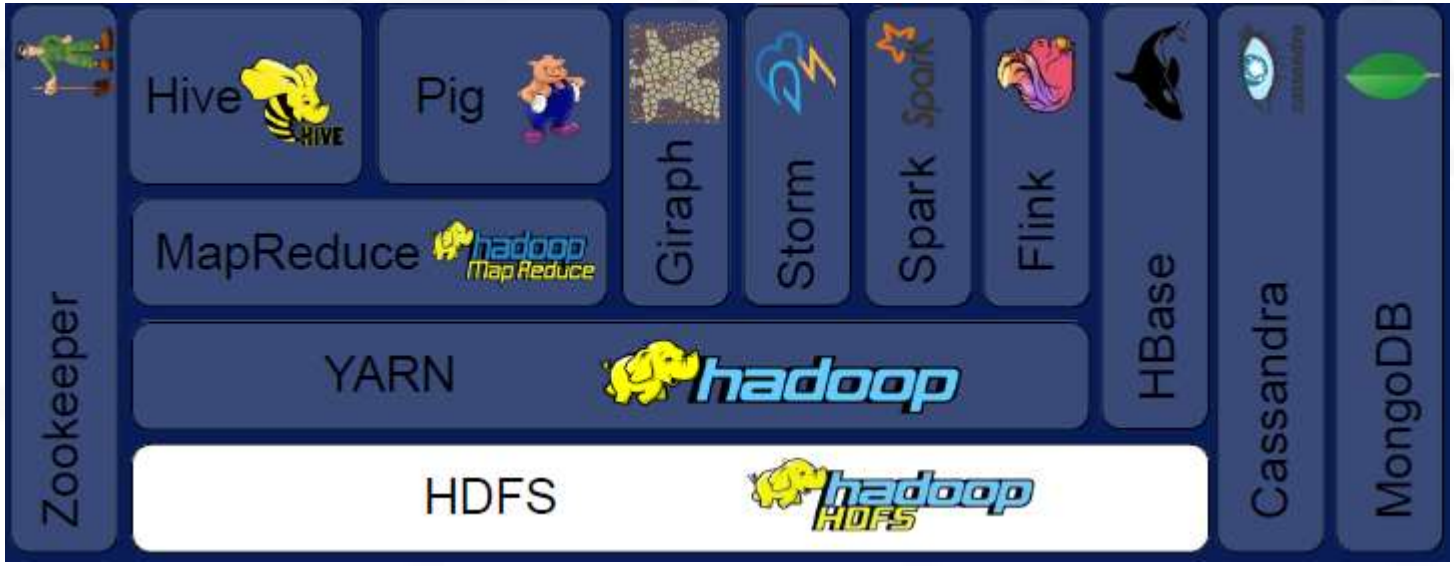


Niveaux inférieurs : Stockage et planification

Basé sur un Système de fichiers distribué (HDFS)

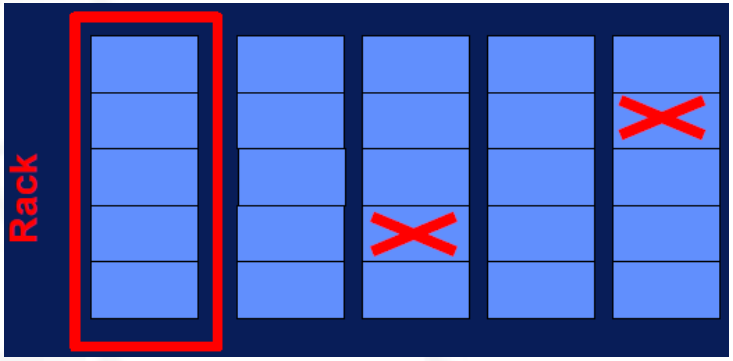
Stockage évolutif

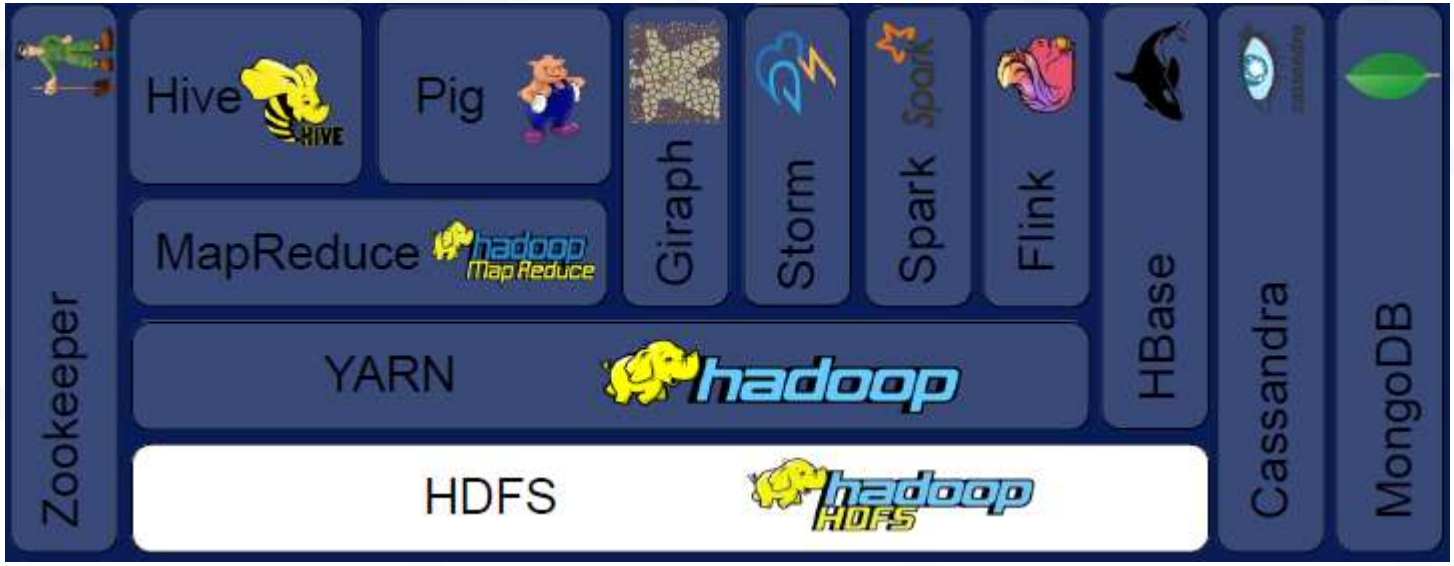




Basé sur un Système de fichiers distribué (HDFS)

Tolérance aux pannes





HDFS : le système de fichiers distribué de Hadoop

Un système de stockage pour le Big Data

Elément de base pour l'écosystème Hadoop

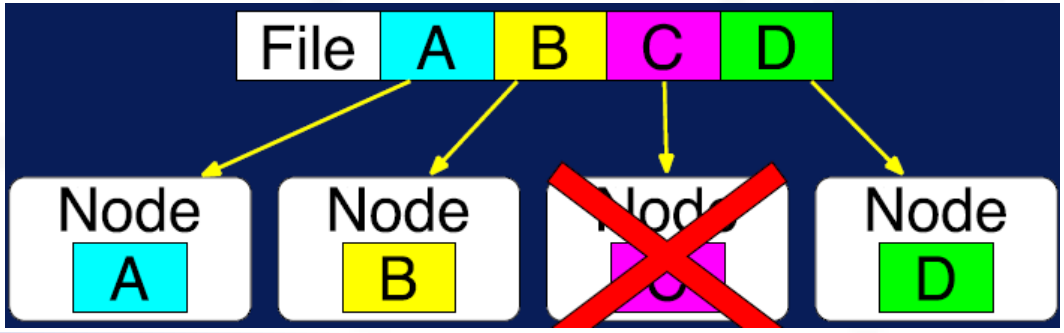
Les données sont réparties sur tout le cluster de machines

Divise les fichiers entre les nœuds pour un accès parallèle

Evolutivité

Fiabilité

Distribué

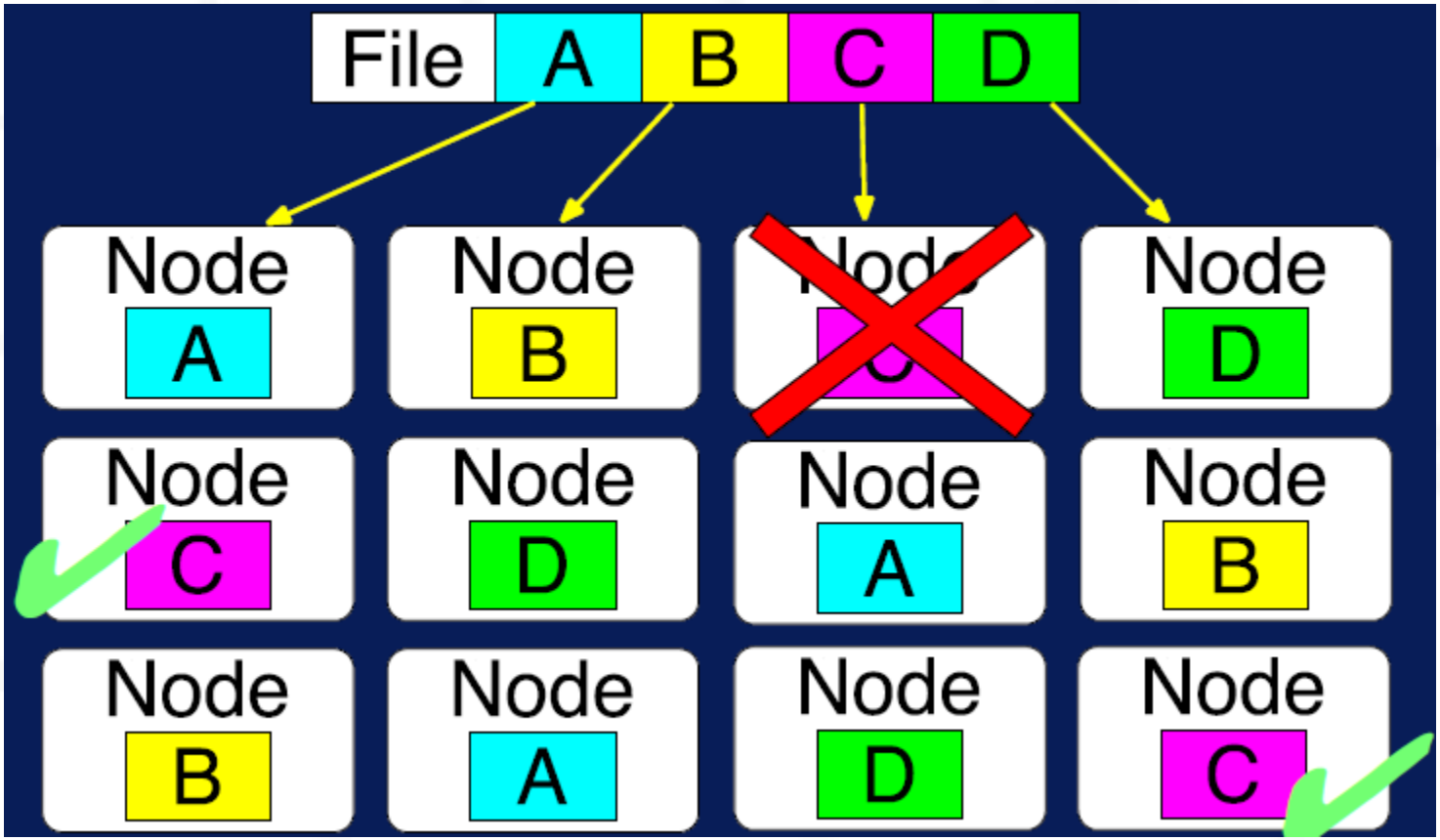


?

HDFS : le système de fichiers distribué de Hadoop

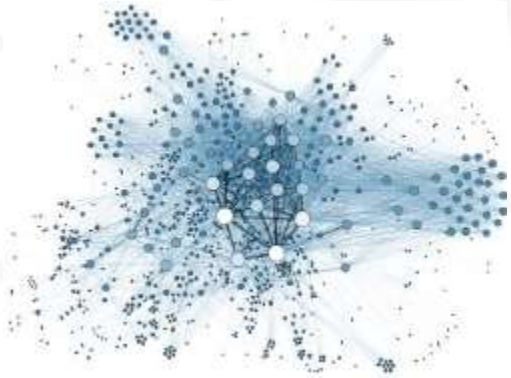
Réplication pour la tolérance aux pannes

Réplication optimisée



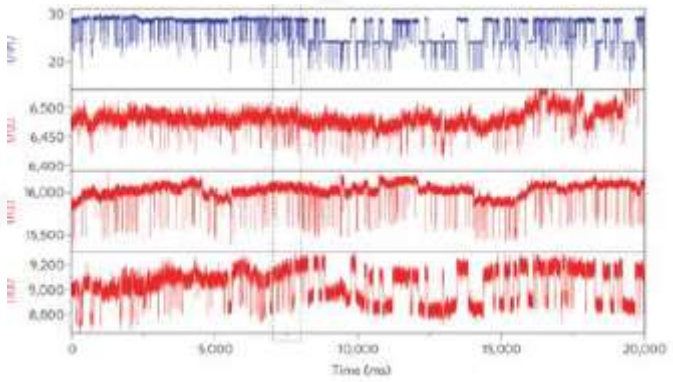
HDFS : le système de fichiers distribué de Hadoop

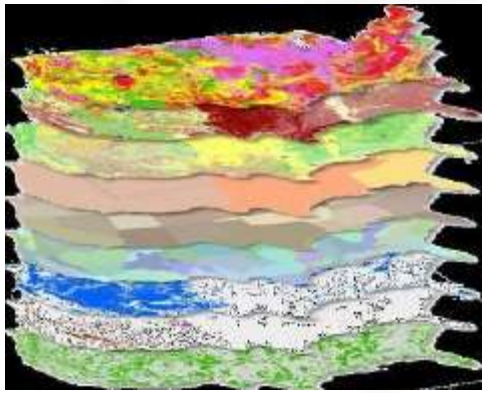
Lecture personnalisée pour gérer plusieurs types de fichiers



Cars marketplace

vendor	Model	Price	Mileage	VIN Code
Chevrolet	Corvette	37226	25965.0	ILLAKAWA2D2
Chevrolet	Corvette	34229	46429.0	RCPNSPYGXOF
Chevrolet	Corvette	27982	56209.0	NWLGGZVEHG
Chevrolet	Corvette	51826	72998.0	NGVZSCDGSN
Chevrolet	Corvette	52846	34364.0	PSDRUYYOUG
Chevrolet	Malibu	37874	37273.0	VLFPQPWNEF
Chevrolet	Malibu	35600	71441.0	EXLJGDWOZS
Chevrolet	Malibu	52447	46700.0	NUNGUZANBP
Chevrolet	Malibu	27129	36254.0	OPFUENLEHS
Chevrolet	Malibu	28846	77362.0	WRCOOFREZU
Chevrolet	Malibu	46165	60990.0	HUFTHQHSFF
Chevrolet	Malibu	38263	37290.0	JHNAPSHVD





HDFS : le système de fichiers distribué de Hadoop

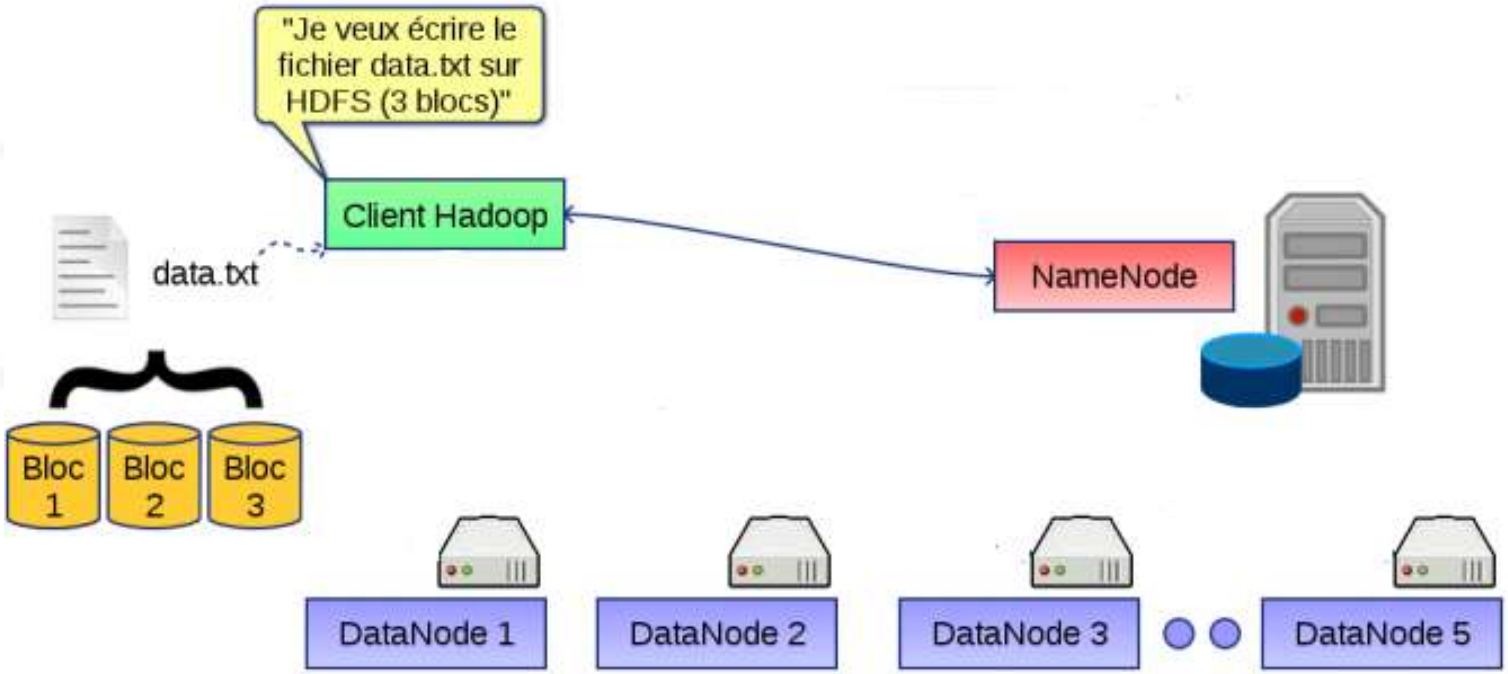
HDFS repose sur deux serveurs (des daemons):

- Le NameNode, qui stocke les informations relatives aux noms de fichiers.
- Il y a un seul NameNode dans tout le cluster Hadoop.

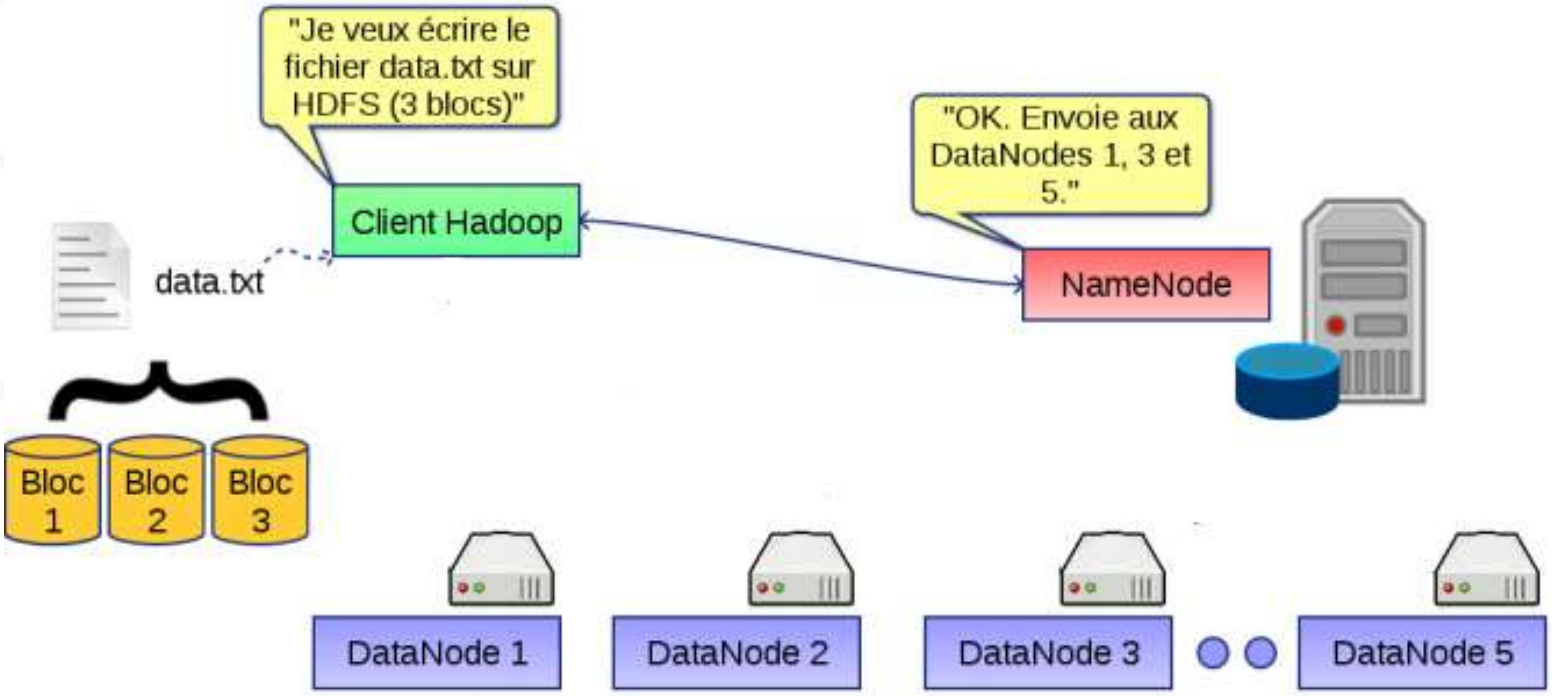
- Le DataNode, qui stocke les blocs de données eux-mêmes.
- Il y a un DataNode pour chaque machine au sein du cluster,
- ils sont en communication constante avec le NameNode pour recevoir de nouveaux blocs, indiquer quels blocs sont contenus sur le DataNode, signaler des erreurs, etc...

- Par défaut, les données sont divisées en blocs de 64MB (configurable).

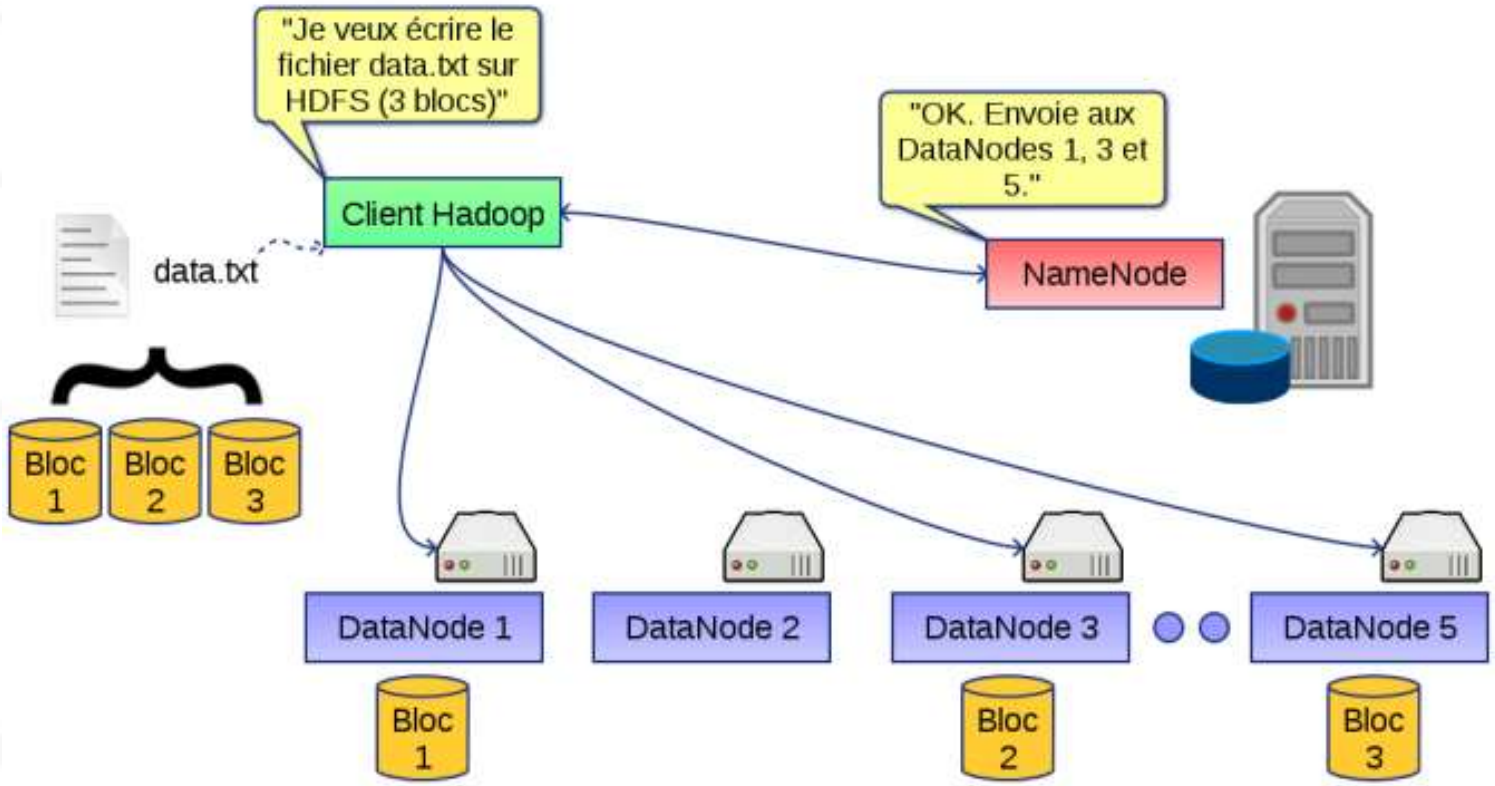
Exemple 1 : écriture d'un fichier HDFS (en 3 blocs)



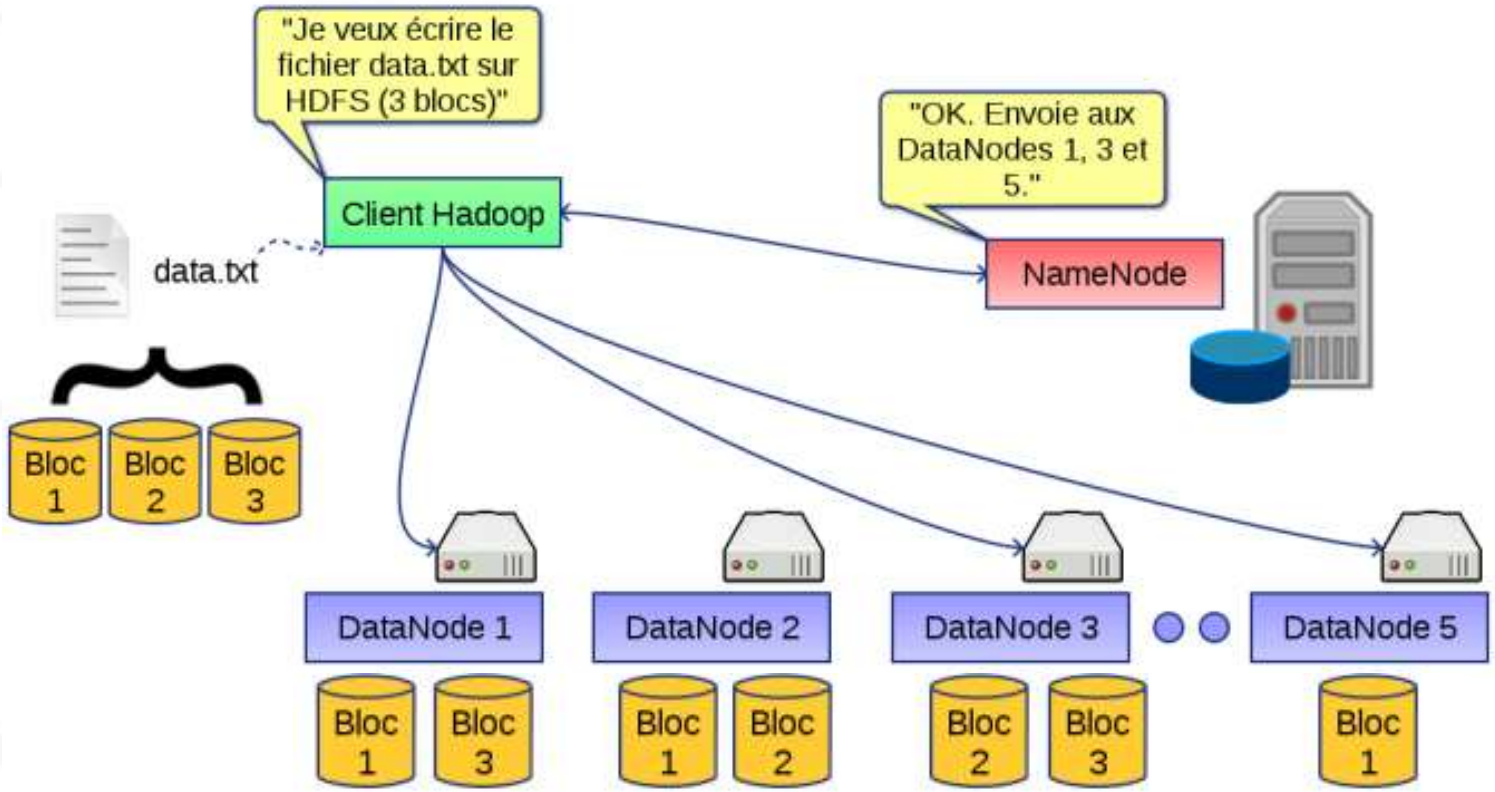
Exemple 1 : écriture d'un fichier HDFS (en 3 blocs)



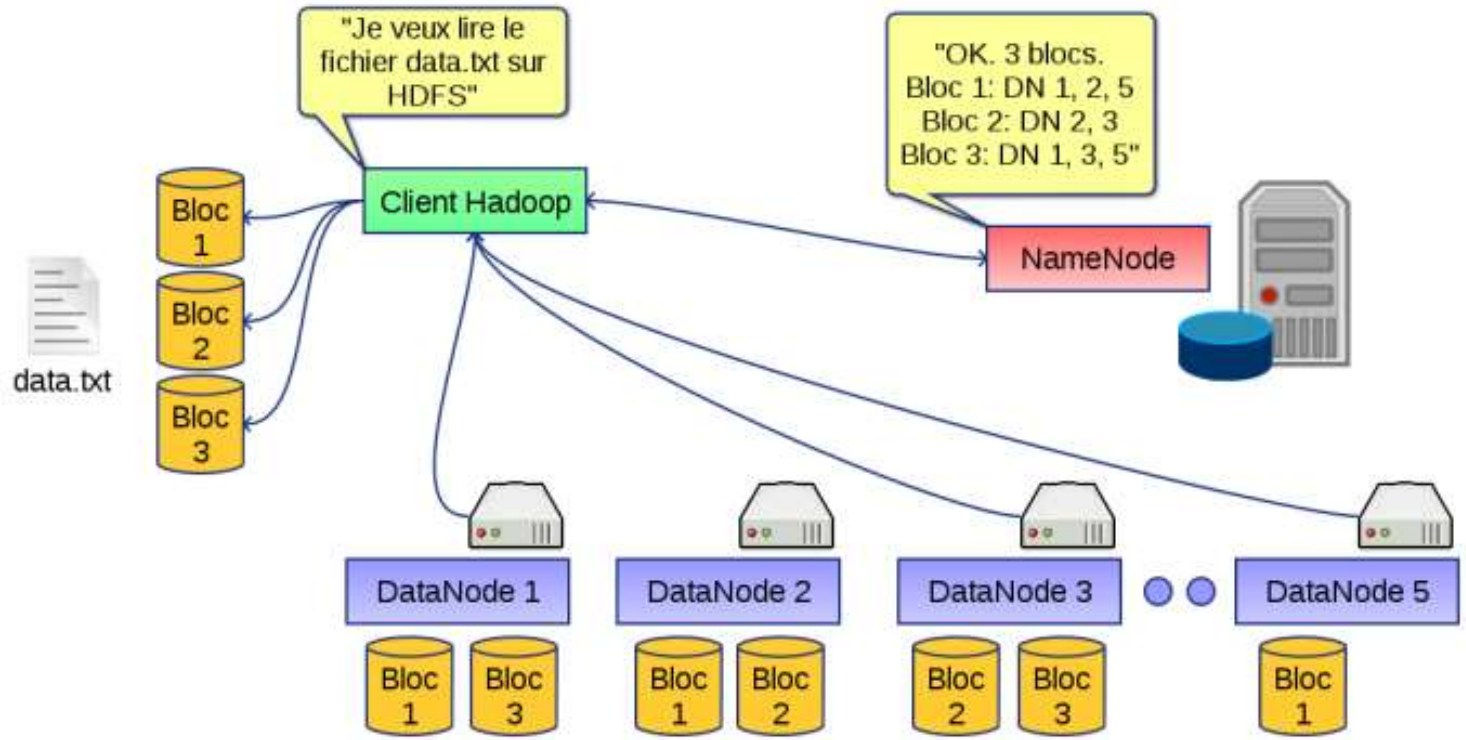
Exemple 1 : écriture d'un fichier HDFS (en 3 blocs)



Exemple 1 : écriture d'un fichier HDFS (en 3 blocs)



Exemple 2 : lecture d'un fichier HDFS (en 3 blocs)



HDFS : le système de fichiers distribué de Hadoop

La commande hadoop fs : la commande permettant de stocker ou extraire des fichiers de HDFS est l'utilitaire console hadoop, avec l'option fs

- Pour stocker le fichier livre.txt sur HDFS dans le répertoire /data_input.

hadoop fs -put livre.txt /data_input/livre.txt

- Pour obtenir le fichier /data_input/livre.txt de HDFS et le stocker dans le fichier local

hadoop fs -get /data_input/livre.txt livre.txt

- Pour créer le répertoire /data_input

hadoop fs -mkdir /data_input

HDFS : le système de fichiers distribué de Hadoop

La commande hadoop fs : la commande permettant de stocker ou extraire des fichiers de HDFS est l'utilitaire console hadoop, avec l'option fs

- Pour créer le répertoire /data_input

hadoop fs -mkdir /data_input

- Pour supprimer le fichier /data_input/livre.txt

hadoop fs -rm /data_input/livre.txt

D'autres commandes usuelles: -ls, -cp, -rmr, -du, etc...

HDFS : le système de fichiers distribué de Hadoop

Inconvénients :

- Le NameNode est unique : on ne peut pas en avoir plusieurs. En conséquence, si la machine hébergeant le NameNode tombe en panne, le cluster est incapable d'accéder aux fichiers le temps que le problème soit corrigé. Ce problème est partiellement mitigé dans les versions récentes (beta) de Hadoop. Une implémentation est en cours pour un basculement automatique.

- HDFS est optimisé pour des lecture concurrentes: écrire de manière concurrente est nettement moins performant.


Zookeeper
Hive
HIVE
Pig
MapReduce
hadoop
Map Reduce
YARN
hadoop
HDFS
hadoop
HDFS
Giraph
Storm
Spark
Spark
Flink
HBase
Cassandra
MongoDB

Programmation flexible et la gestion des ressources

YARN planifie les tâches sur
> 60 000 serveurs chez Yahoo

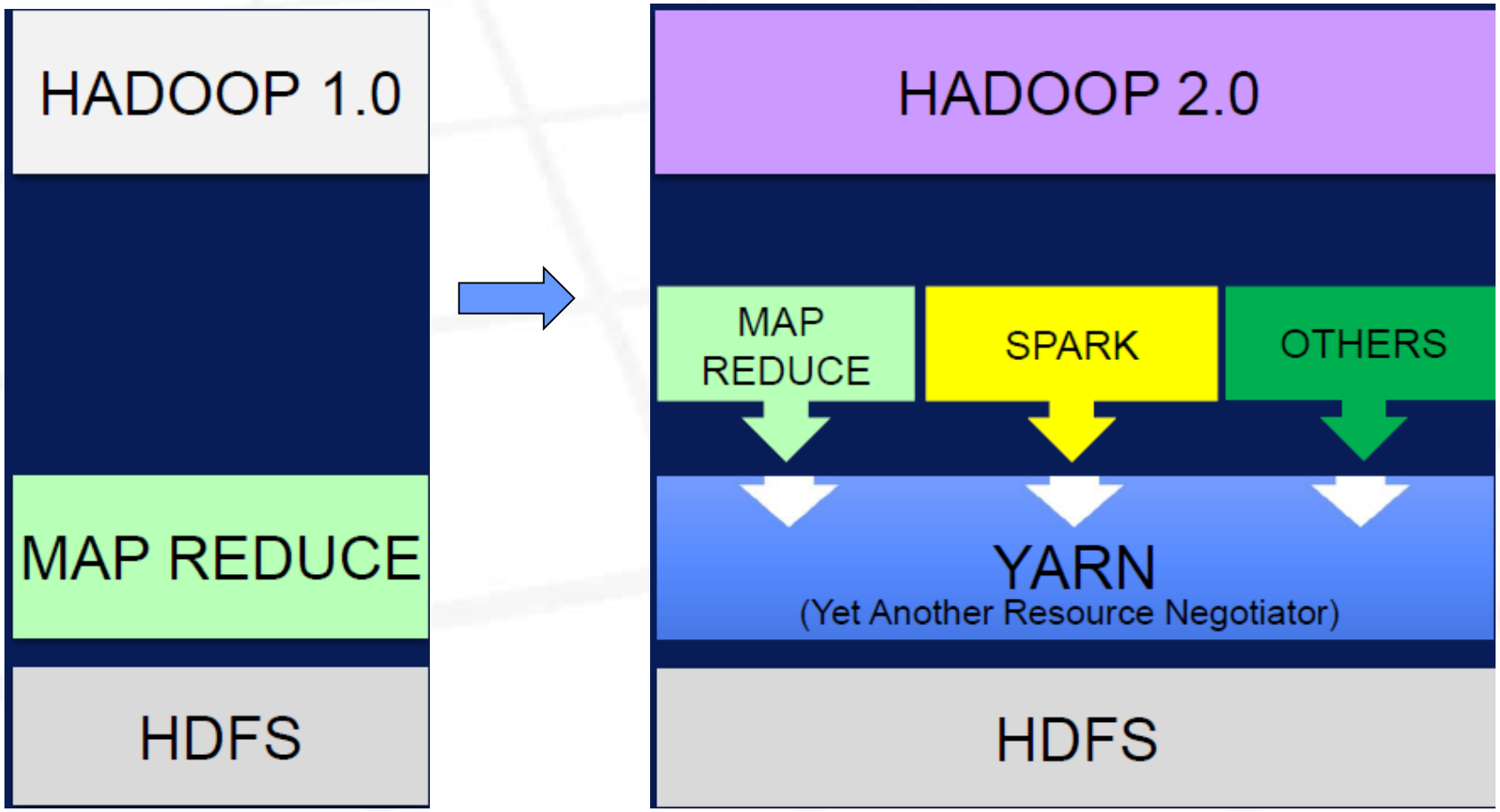
YARN : la gestion des ressources pour Hadoop

YARN fournit une gestion flexible des ressources pour le cluster Hadoop

Utilisation plus optimisée des ressources → Coûts plus faibles

Possibilité d'utiliser plusieurs applications sur le même jeu de données

YARN étend Hadoop pour permettre la connexion avec plusieurs frameworks tels que MapReduce, Giraph, Spark et Flink



YARN : vue générale

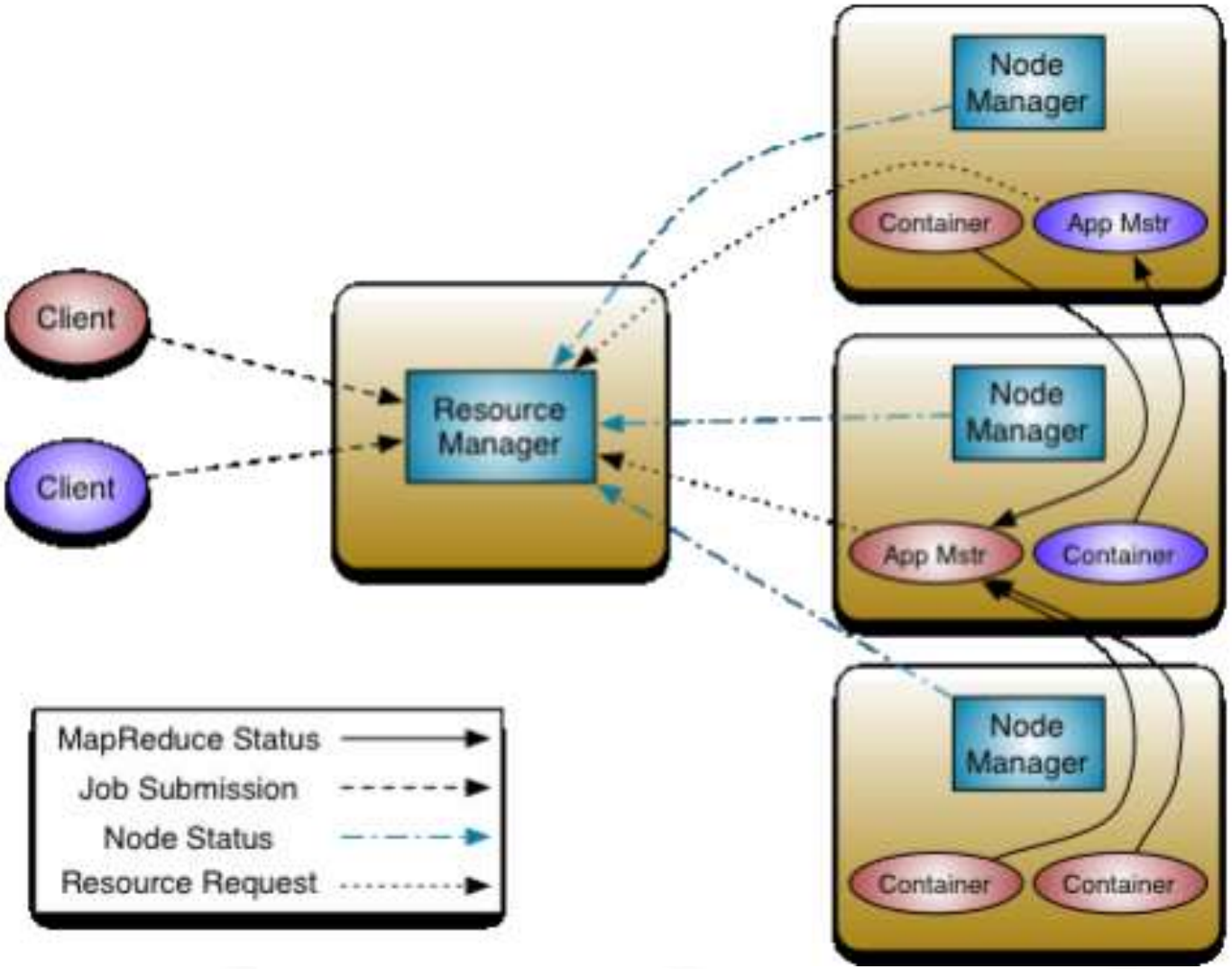
Le démon principal est Resource Manager;

- il est unique sur le cluster et gère le concept de « resources ».

- Le Resource Manager maintient une table de « resources » disponibles sur le cluster: machines, slots d'exécution, mémoire/CPU sur les machines, etc.

- Il est composé de deux principaux composants:
  - Le scheduler, en charge de distribuer des travaux de manière générique aux différents slots d'exécution du cluster, et en fonction des ressources disponibles et des règles liées à leur attribution.
  - L'Application Manager, à l'écoute de nouveaux programmes à exécuter proposés par des clients, et qui lancera une première tâche lors du début de l'exécution: la tâche Application Master, qui constituera le « chef d'orcheste » de l'exécution du programme et soumettra par elle-même de nouvelles tâches à effectuer au scheduler du ResourceManager directement.

YARN : vue générale



YARN : vue générale

NodeManager;

- Du coté des nœuds, c'est un second démon, NodeManager, qui tourne sur chaque machine

- Ce démon est en charge de maintenir les slots d'exécution locaux (désignés sous le terme générique de container d'exécution), et de recevoir et exécuter des tâches dans ces containers attribués aux applications par le scheduler.

- Ces tâches à exécuter peuvent être de simples opérations map ou reduce, mais aussi le coeur de l'application lui-même : la classe driver, main du programme, qui serait alors lancée au sein d'une tâche spéciale ApplicationMaster, lancée par l'ApplicationMaster du ResourceManager.

Exemple YARN : soumission d'un programme

Les étapes d'exécution d'un programme via Yarn:

1- Un client se connecte au ResourceManager, et annonce l'exécution du programme / fournit son code (jar).

2- L'Application Manager du ResourceManager prépare un container libre poury exécuter l'Application Master du programme (son main).

3- L'application Master du programme est lancé; il s'enregistre immédiatement auprés du ResourceManager et effectue ses demandes de ressources (containers pour exécuter tâches map et reduce, par exemple).

4- Il contacte alors directement les NodeManager correspondants aux containers qui lui ont été attribués pour leur soumettre les tâches.

Exemple YARN : soumission d'un programme

Les étapes d'exécution d'un programme via Yarn:

5- Pendant l'exécution des différentes tâches, les containers (par le biais des démons NodeManager) mettent à jour l'Application Master sur leur statut continuellement. En cas de problème, celui-ci peut demander de nouveaux containers au Resource Manager pour ré-exécuter une tâche.

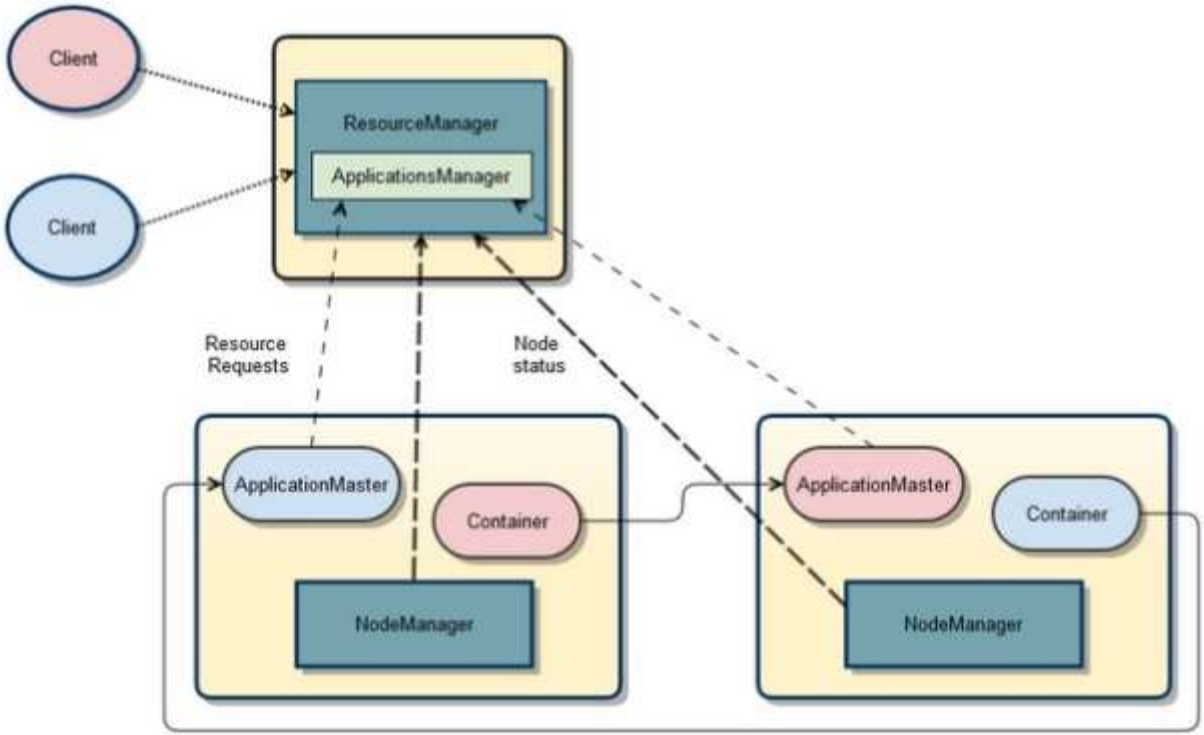
6- L'utilisateur peut par ailleurs obtenir une mise à jour sur le statut de l'exécution soit en contactant le ResourceManager, soit en contactant l'Application Master directement.

7- Une fois toutes les tâches exécutées, l'Application Master signale le ResourceManager et s'arrête.

8- Le ResourceManager libère alors le container occupé restant, qui exécutait le code de l'Application Master.

Exemple YARN : soumission d'un programme

Les étapes d'exécution d'un programme via Yarn:



Modèle simplifié de programmation

Google a utilisé MapReduce pour l'indexation de sites Web

Map → apply() : appliquer l'opération à tous les éléments

Reduce → Summarize() : résumer opération sur éléments

MapReduce

Programmation simple pour de grands résultats :

Pour exécuter un problème large de manière distribué, il faut pouvoir découper le problème en plusieurs problèmes de taille réduite à exécuter sur chaque machine du cluster (stratégie algorithmique dite diviser pour régner).

MapReduce est un paradigme (un modèle) visant à généraliser les approches existantes pour produire une approche unique applicable à tous les problèmes

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 clef/valeur.
- Elle va regrouper les données en les associant à des clefs, choisies de telle sorte que les couples clef/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.

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 clefs distinctes produite par l'opération MAP. Au terme de l'opération REDUCE, on aura un résultat pour chacune des clefs 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.

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 (clef ; valeur)..

- Grouper (shuffle) ces couples (clef ; valeur) par clef.

- Réduire (reduce) les groupes indexés par clef en une forme finale, avec une valeur pour chacune des clefs distinctes.

MapReduce

Pour résoudre un problème via la méthodologie MapReduce avec Hadoop, on devra donc :

- Choisir une manière de découper les données d'entrée de telle sorte que l'opération MAP soit parallélisable.

- Définir quelle CLEF utiliser pour notre problème.

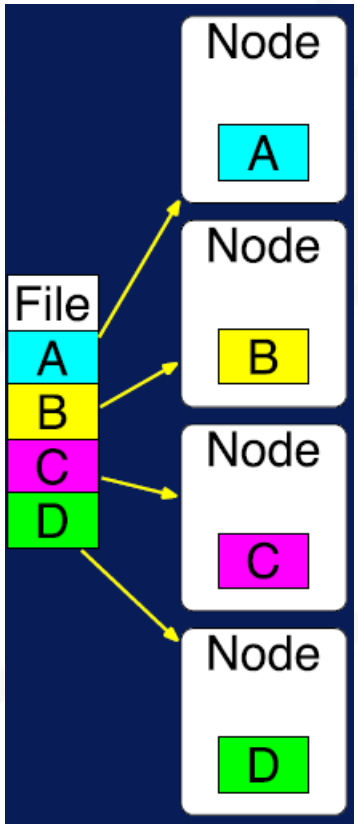
- Écrire le programme pour l'opération MAP.

- Ecrire le programme pour l'opération REDUCE.

… et Hadoop se chargera du reste (problématiques calcul distribué, groupement par clef distincte entre MAP et REDUCE, etc.).

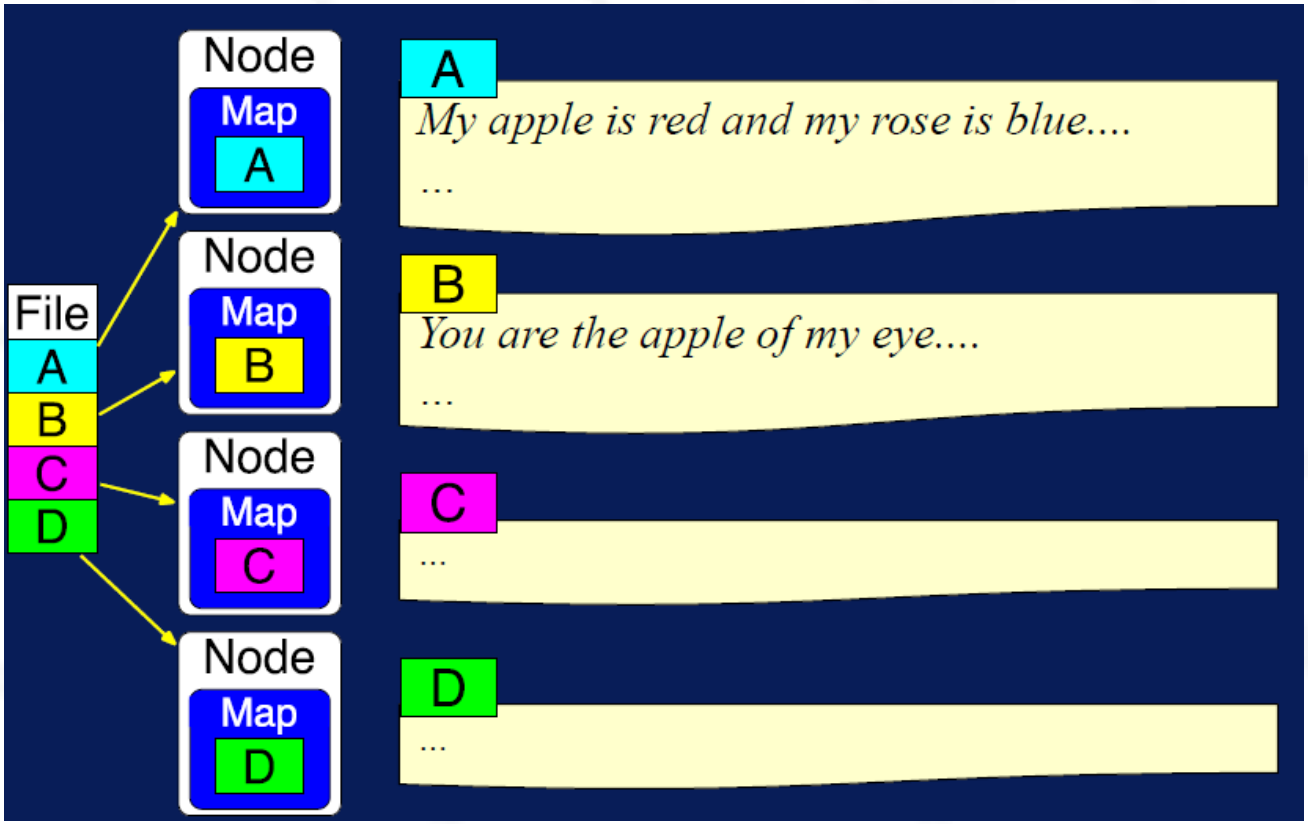
Exemple de programme MapReduce : WordCount

Etape 0 : enregistrement du fichier sur HDFS



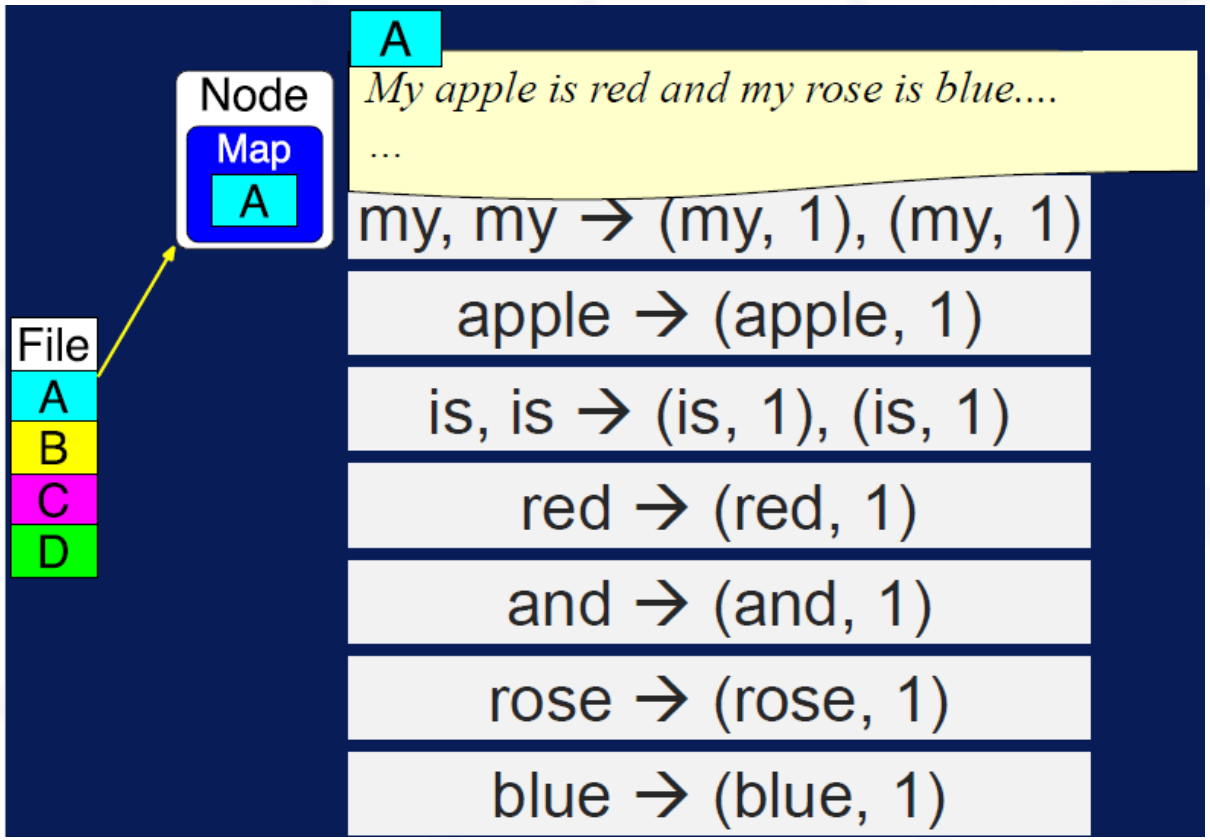
Exemple de programme MapReduce : WordCount

Etape 1 : appliquer Map sur chaque « Node »



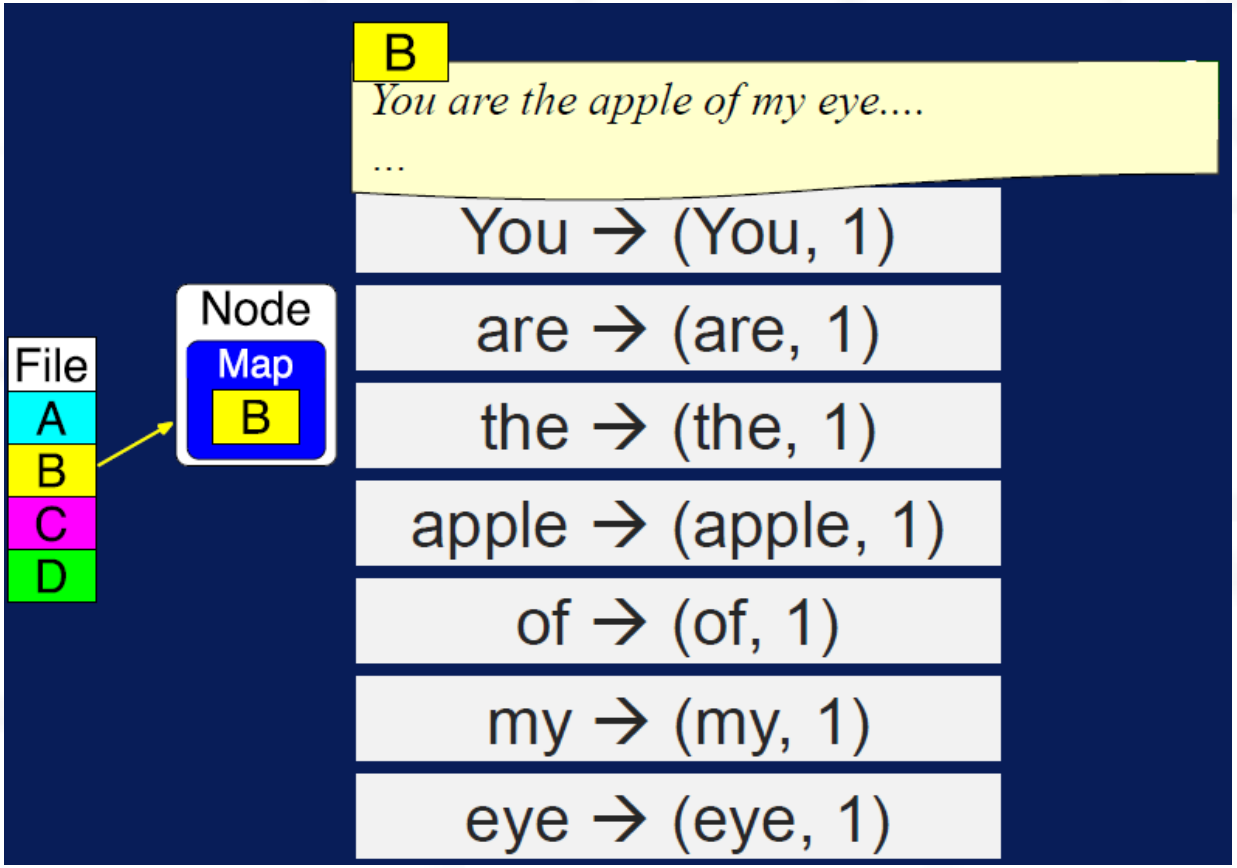
Exemple de programme MapReduce : WordCount

Map produit les paires (clef ; valeur)



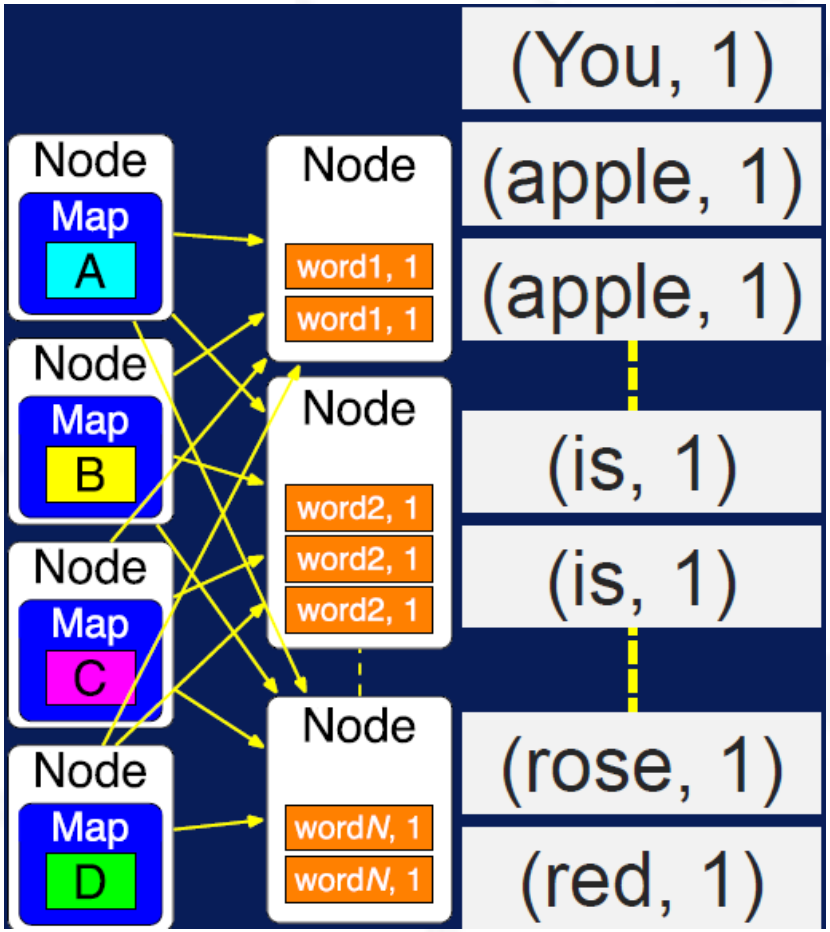
Exemple de programme MapReduce : WordCount

Map produit les paires (clef ; valeur)



Exemple de programme MapReduce : WordCount

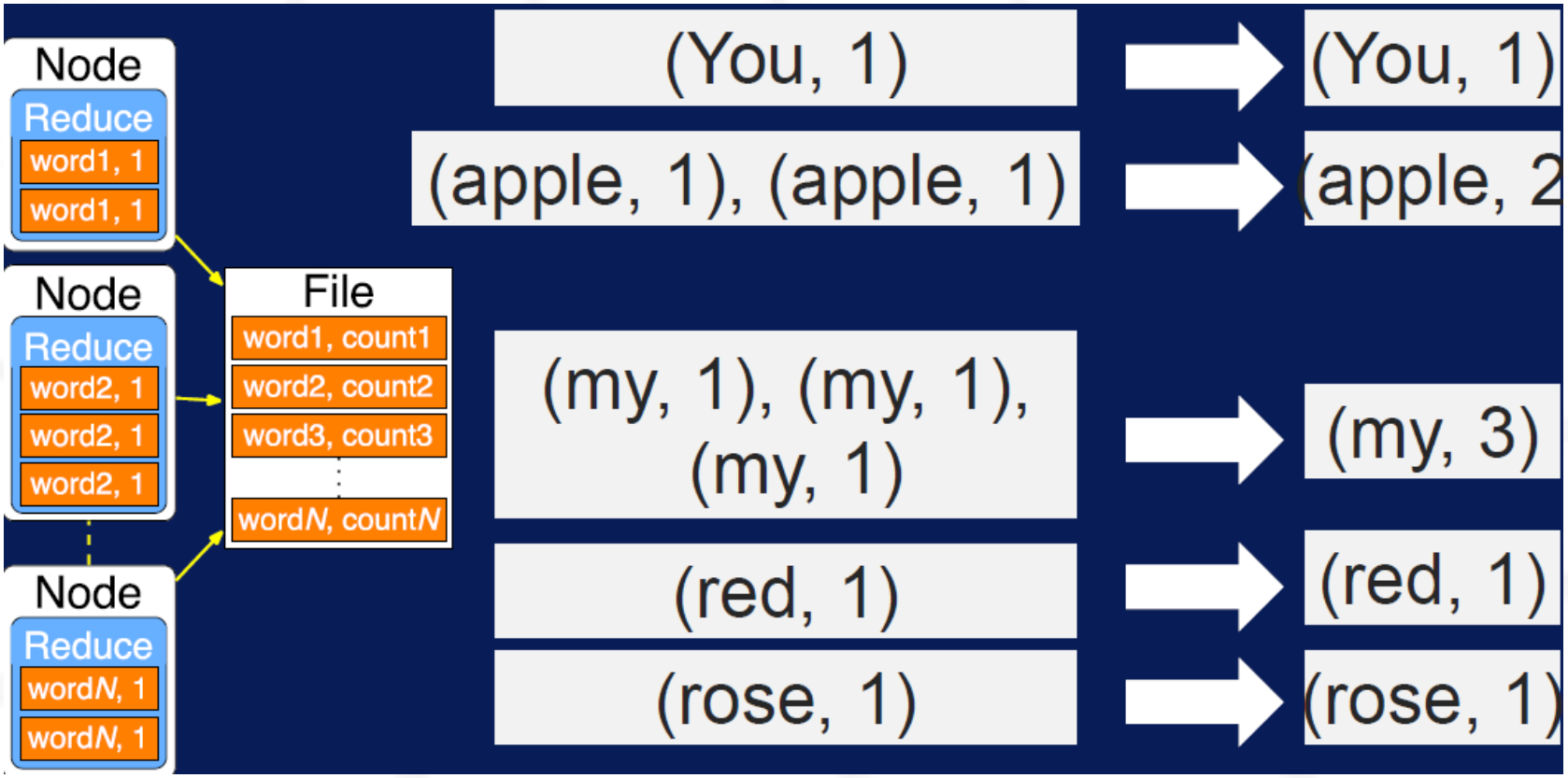
Etape 2 : trier et grouper



Les paires avec la même (clef ; valeur) seront envoyés au même « Node »

Exemple de programme MapReduce : WordCount

Etape 3 : réduire



Exercice : pour les exemples suivants :

Exemple 1 : on souhaite compter le nombre de visiteurs sur chacune des pages d'un site Internet.
On dispose des fichiers de logs sous la forme suivante

Exemple 2 : on administre un réseau social comportant des millions d'utilisateurs. Pour chaque utilisateur, on a dans notre base de données la liste des utilisateurs qui sont ses amis sur le réseau (via une requête SQL).
On souhaite afficher quand un utilisateur va sur la page d'un autre utilisateur une indication « Vous avez N amis en commun »

1- proposer des couples (clef ; valeur) avec lesquels le traitement soit parallélisable

2- Ecrire les squelettes des programmes de Map et Reduce

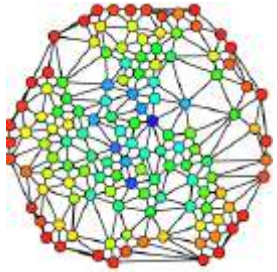
Modèles de programmation de niveau supérieur

Pig créé chez Yahoo, Hive créé sur Facebook

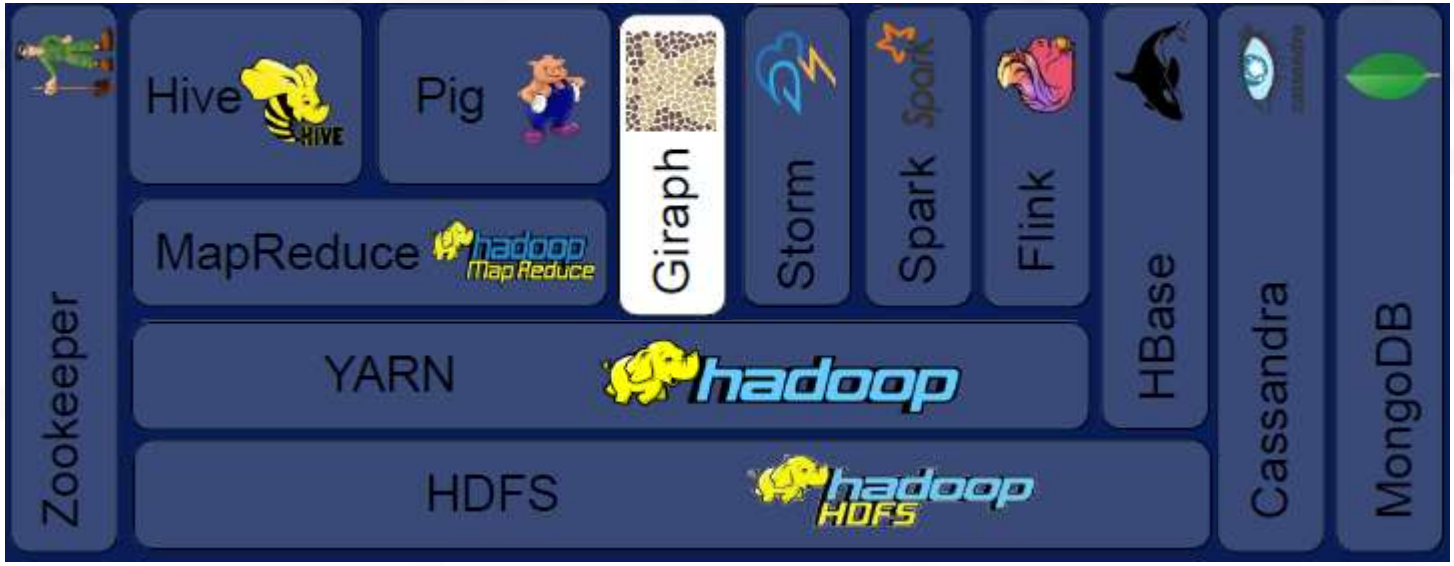
Hive = requêtes de type SQL

Pig = script de flux de données

Modèles spécialisés pour le traitement graphique

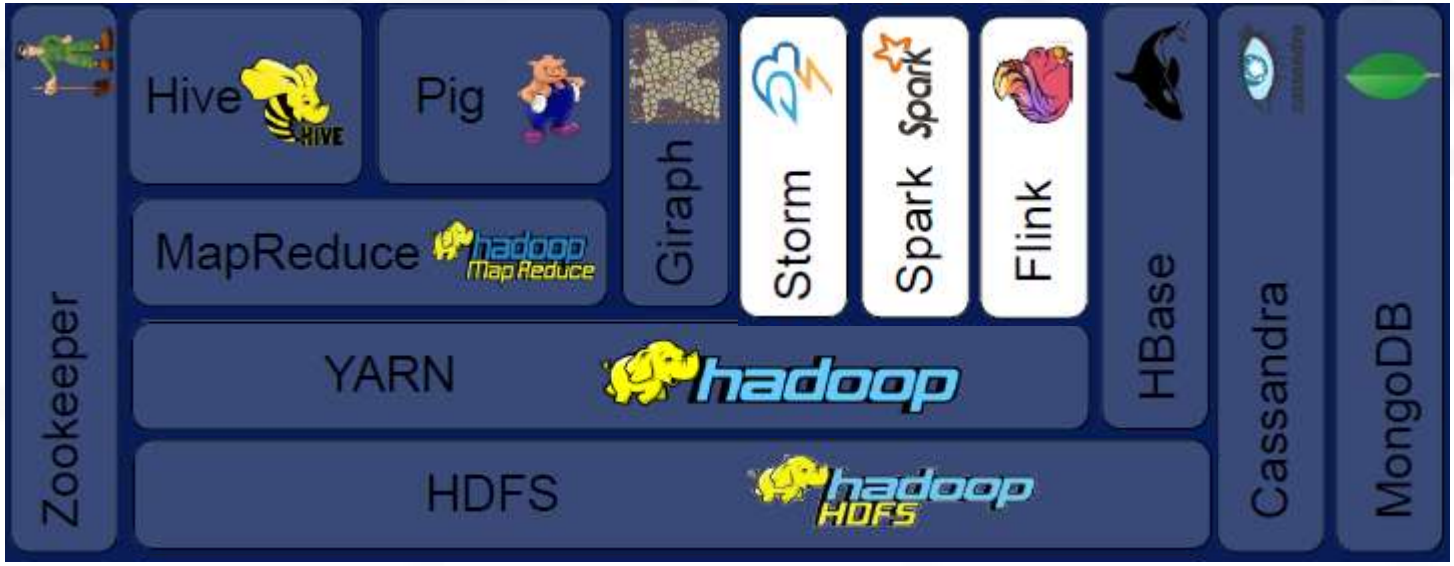


Giraph utilisé par Facebook
pour analyser les graphes sociaux



Traitement en temps réel et en mémoire

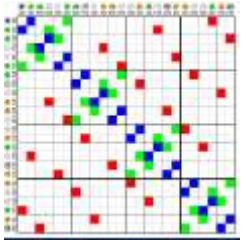
En mémoire → 100x plus rapide pour certaines taches



Bases de données de types NoSQL

HBase utilisée dans la plateforme de messagerie de Facebook

Clés-valeurs



Sparse tables

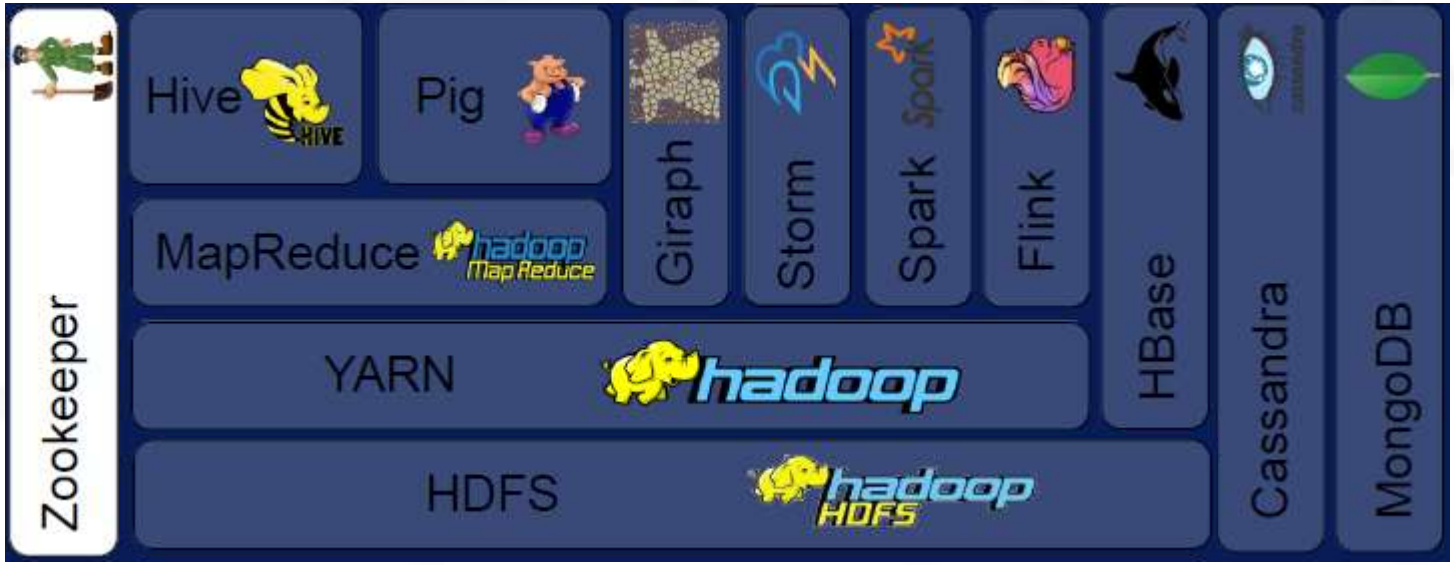
Zookeeper : logiciel de gestion de configuration pour systèmes distribué

ZooKeeper est un sous projet de Hadoop mais il est un projet top-level à part entière

Configuration

Synchronisation

Disponibilité haute



Tous ces outils sont open-source

Grande communauté

Télécharger séparément
ou partie d'une image pré-construite