




# Chapitre 4 –Mapreduce





# 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

- MapReduce1
- Fonctionnement
- Exemple
- Composants de MapReduce
- Limites de MPR1
- Nouveautés Hadoop2
  - Yarn
  - Gestion des ressources



# MapReduce

- Patron d'architecture de développement
- Permet de traiter des données volumineuses de manière parallèle et distribuée
- Les langages utilisés : Java, Python ou Ruby
- Au lieu de parcourir le fichier séquentiellement (beaucoup de temps), il est divisé en morceaux qui sont parcourus en parallèle.

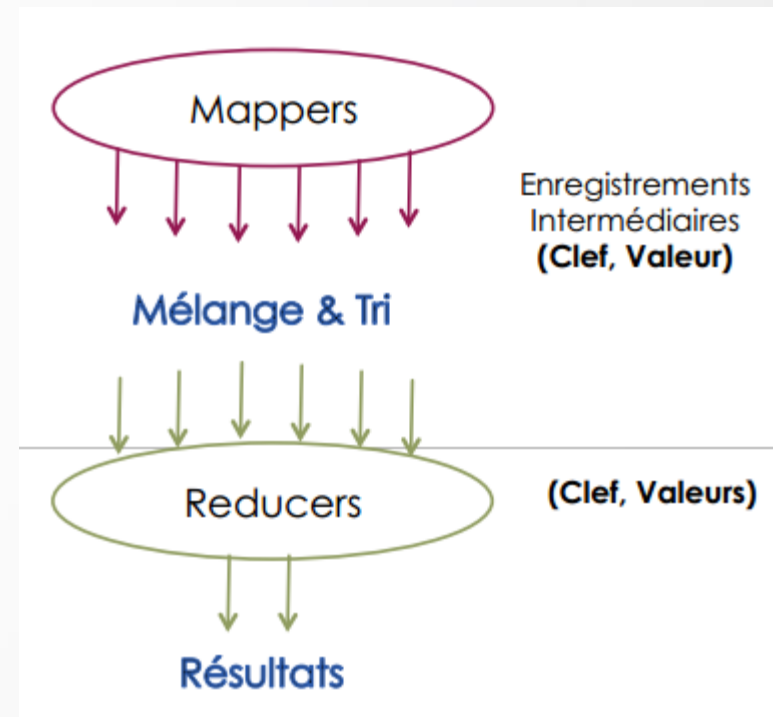


# MapReduce

- Ce modèle se base sur 2 étapes :
  - **Mapping** (map tasks) : le développeur définit une fonction de mappage dont le but sera d'analyser les données brutes contenues dans les fichiers stockés sur HDFS pour en sortir les données correctement formatées.
  - **Réduction** (reduce tasks) : cette tâche récupère les données construites dans l'étape du mappage et s'occupe de les analyser dans le but d'en extraire les informations les plus importantes.

# ► MapReduce: Fonctionnement

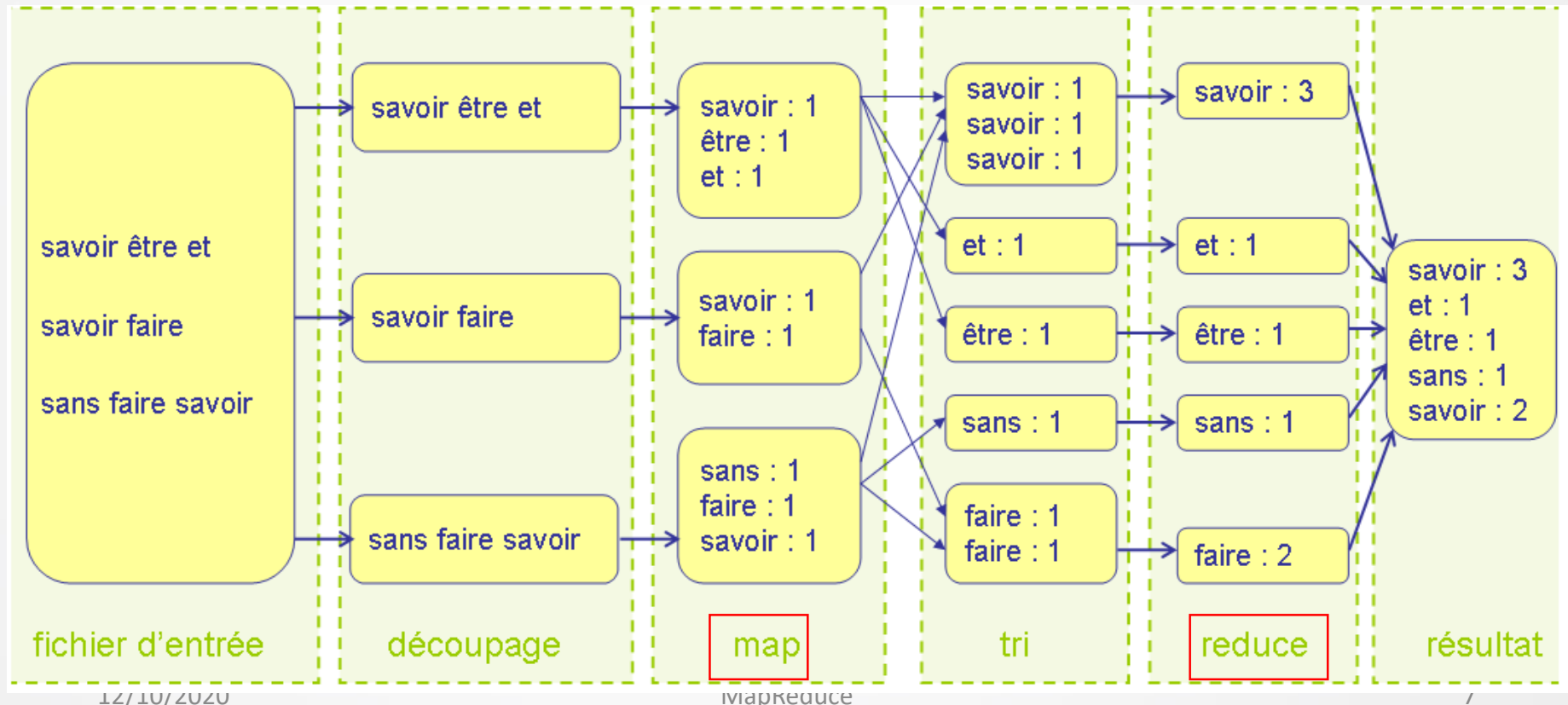
- Les **Mappers** sont de petits programmes qui commencent par traiter chacun une petite partie des données
- Ils fonctionnent en parallèle
- Leurs sorties représentent les enregistrements intermédiaires : sous forme d'un couple (clef, valeur)
- Une étape de Mélange et Tri s'ensuit
  - Mélange : Sélection des piles de fiches à partir des Mappers
  - Tri: Rangement des piles par ordre au niveau de chaque Reducer
- Chaque **Reducer** traite un ensemble d'enregistrements à la fois, pour générer les résultats finaux



# ► MapReduce: Exemple



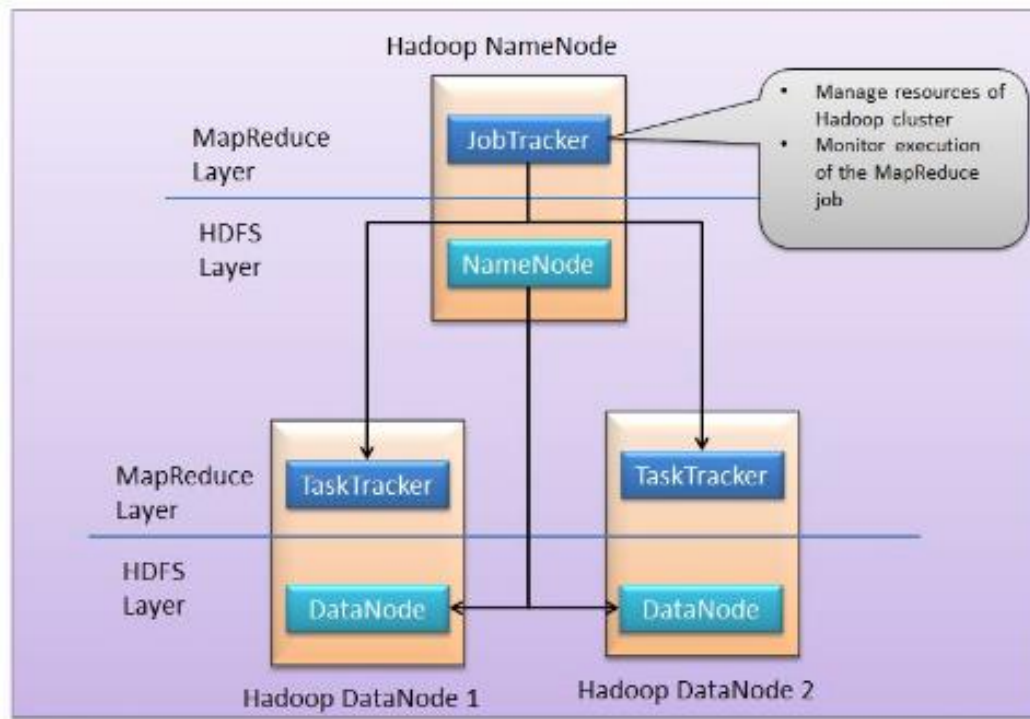
- But : Calculer le nombre d'occurrences des mots constituant le texte.
- L'ensemble du processus est schématisé ci-dessous



# ► MapReduce



- Deux processus JobTracker et TaskTracker







# MapReduce

- **JobTracker**

- Planifie les tâches,
- Affecte les tâches aux **TaskTrackers**.
- Gère les jobs MapReduce et surveille les progrès réalisés
- Récupère les erreurs, et redémarre les tâches lentes et les tâches qui ont échoué

- **TaskTracker**

- Notifie périodiquement le **JobTracker** du niveau de progression d'une tâche ou bien le notifie en cas d'erreur afin que celui-ci puisse reprogrammer et assigner une nouvelle tâche.
- S'exécute sur chacun des nœuds pour exécuter les vraies tâches de **MapReduce**
- Choisit en général de traiter (Map ou Reduce) un bloc sur la même machine que lui



# Gestion des ressources

- MapReduce a un modèle de gestion de mémoire inflexible basé sur les slot.
  - Chaque **TaskTracker** est configuré au démarrage pour avoir un nombre bien déterminé de slots (map slot, reduce slot) pour l'exécution des tâches
  - Une tâche est exécutée dans un seul slot
  - Les slots sont configurés au démarrage pour avoir un maximum d'espace mémoire



# ► Limites de MPR1

- **Scalabilité limitée** : Le **JobTracker** s'exécute sur une seule machine.  
Ses rôles sont :
  - Gestion des ressources
  - Ordonnancement et suivi des Job
- **Problème de disponibilité** : Le **JobTracker** est un SPOF. S'il est endommagé, tous les jobs doivent être redémarrés
- **Problème d'utilisation des ressources** : Il y a un nombre prédéfini de map slots et reduce slots pour chaque **TaskTrackers**.
- **Utilisation des applications non-MapReduce** : Le **JobTracker** est intégré à MapReduce et ne supporte que les applications utilisant le framework de programmation MapReduce

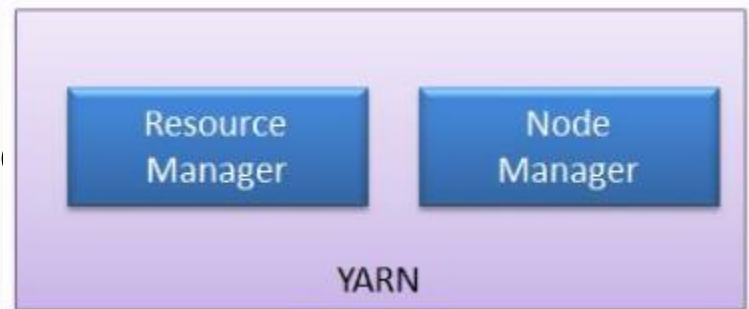
# ► Nouveautés Hadoop 2



- **YARN** – Possibilité de traitement de Terabytes et Petabytes de données existants dans HDFS en utilisant des application Non-MapReduce
- **Resource Manager** – Séparation des deux fonctionnalités essentielles du jobtracker (gestion des ressources et ordonnancement et suivi des jobs) en deux composants séparés :
  - Gestionnaire de Resource
  - Gestionnaire d'application



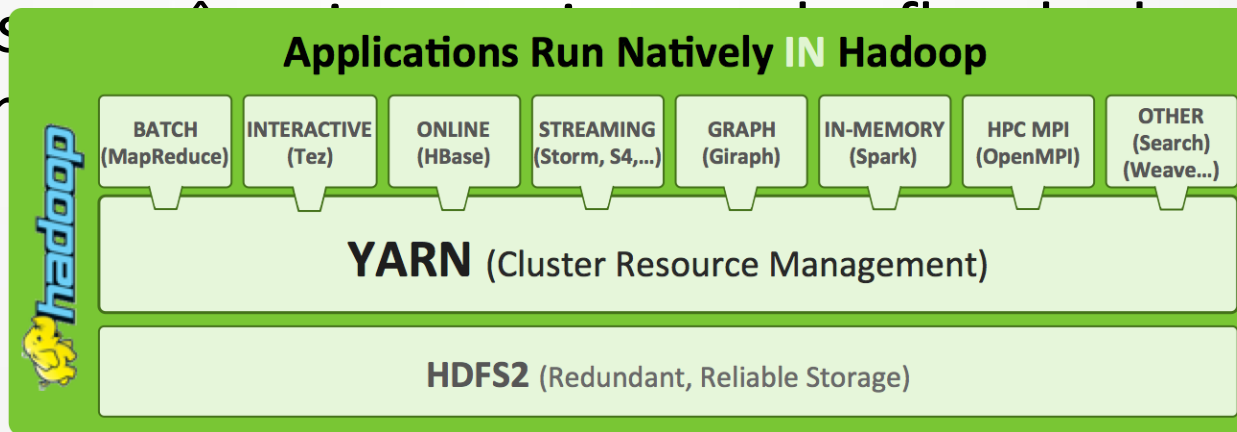
Jobtracker et Tasktrack  
n'existent plus.



# ► YARN



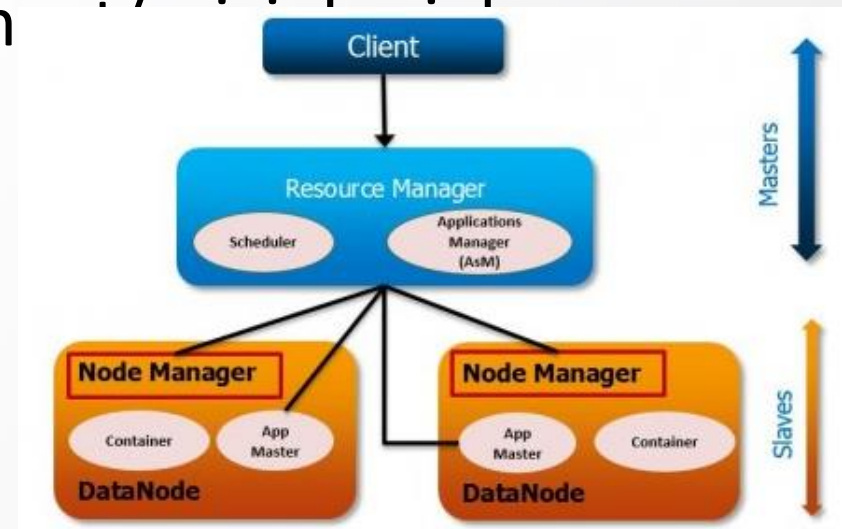
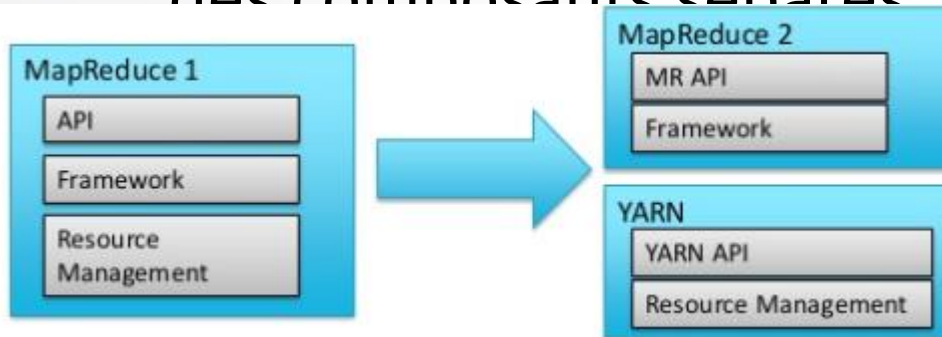
- YARN sépare la gestion des ressources/ordonnancement des job du traitement des données. Permettant a Hadoop de supporter d'autre applications de traitement de données.
- Par exemple, Hadoop peut maintenant exécuter des applications de traitement de données similaires à MapReduce.



# ► Gestion des ressources



- La nouvelle génération du framework MapReduce est une application qui s'exécute avec Yarn.
- Le nouveau framework MRv2 divise les deux fonctions majeurs du JobTracker, en gestion des ressources et ordonnancement des composants séparés

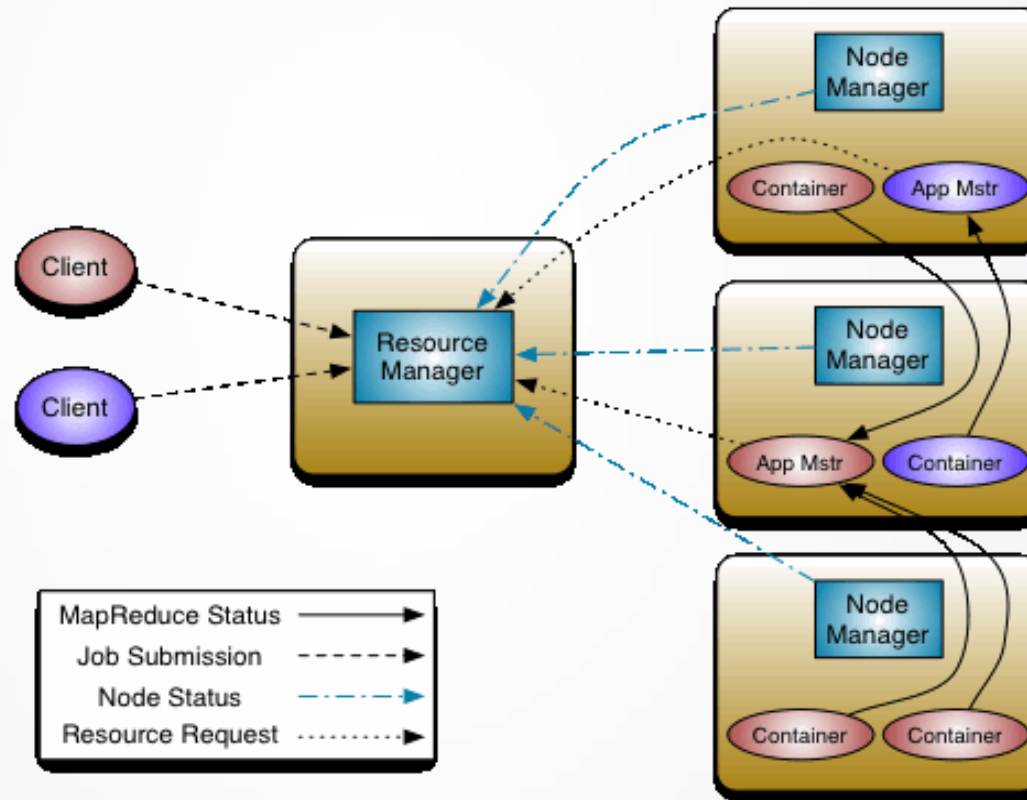




# Gestion des ressources

- Resource Manager (RM)
  - Tourne sur le nœud master
  - Ordonnanceur de ressources global
  - Permet l'arbitrage des ressources entre plusieurs applications
- Node Manager (NM)
  - S'exécute sur les nœuds esclaves
  - Communique avec le RM
- Application Master (AM)
  - Créé par le NM
  - Un seul AM par application
  - Demande plusieurs containers pour exécuter les tâches de l'application
- Container
  - Créé par l'AM
  - Se voit allouer des ressources sur le nœud esclave

# ► Exécution d'un job





# ► Fault Tolerance



- **Task (Container)** : L'application Master essaye de re-exécuter la tâche. Si au bout de 4 tentatives aucune réponse, la tâche est considérée comme *failed*.
- **Application Master** : si AM n'envoie plus de heartbeat, RM essaiera de re-exécuter la totalité de l'application (2 tentatives)

Si la valeur de l'option **Job recovery** du AppMaster est false, toutes les tâches seront re-exécutées. Sinon, l'AppMaster reprend l'état des tâches lors du démarrage et re-exécute les tâches incomplètes

- **NodeManager** : si les NM n'envoient plus de heartbeats au RM, il sera supprimé de la liste des nœuds. Les tâches seront considérées comme failed par l'AppMaster.
- **ResourceManager** : aucune application ni tâche ne peut être exécutée si le RM est non fonctionnel. (possibilité d'avoir un Resource Manager active et un Resource Manager (standby))