Université Mohammed V- Rabat
Ecole Mohammadia d'Ingénieurs
Département Génie Informatique
Filière Génie Informatique et Digitalisation

Traitement Big Data

Pr. N. EL FADDOULI
nfaddouli@gmail.com

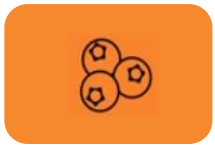
2023-2024

Introduction générale

- Problèmes du Big Data
- La solution Hadoop
- Limites de Hadoop.

Introduction: Problèmes Big Data

❑ Besoin de traiter des données qui sont caractérisées par ***(problème Big Data)***:



Variété ***(Structurées, Semi-Structurées, Non Structurées)***

Volume ***(TéraOctet, PétaOctet, ExaOctet, ZettaOctet, YottaOctet, RonnaOctet)***

Vélocité ***(génération rapide + traitement rapide: streaming, …)***

Introduction: Approche Monolithique

❑ Approches des solutions Big Data:

- ➢ Monolithique
- ➢ Distribuée

❑ Monolithique:

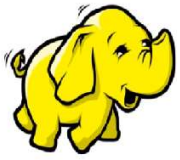
- ➢ Ressources massives *(CPU, RAM, Disque)*
- ➢ Exemples: **Teradata** et **Exadata** pours des données structurées et semi-structurées.

Introduction: Approche Distribuée

- **Approche distribuée**:
  - Utiliser un cluster de machines pas très chères.
  - Distribution des données et des traitements.
  - Exemples: Hadoop, Databricks, Google Cloud Dataprep, …

- Critères de de choix:
  - **Evolutivité**: Scalabilité verticale ou horizontale, coût et temps de mise à l'échelle.
  - **Tolérance aux pannes**: Haute disponibilité des données et des applications.
  - **Rentabilité**: Rapport coût/rendement.

La solution Hadoop



Stockage et traitement par lot distribués

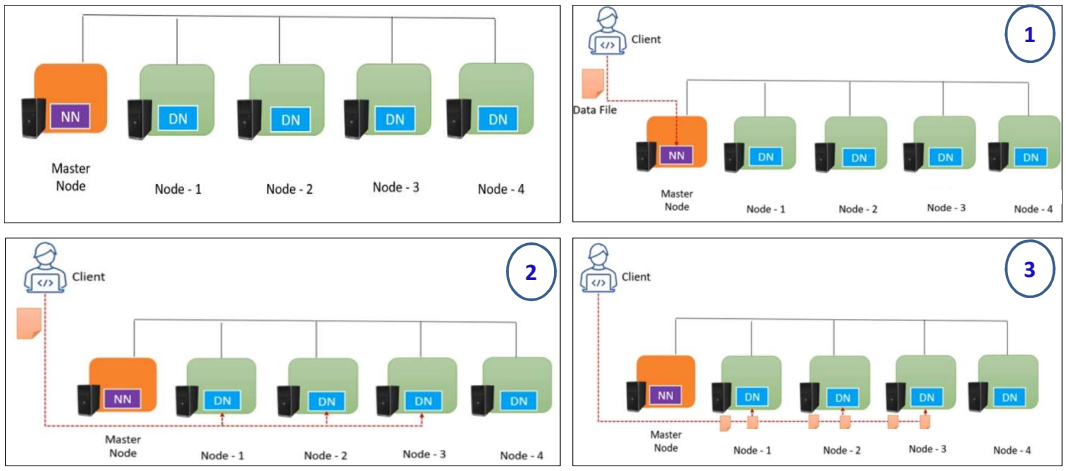
Map/Reduce – Framework et modèle de Traitement distribué

YARN – Gestionnaire des ressources (RAM, CPU, Disque) du cluster

HDFS – Stockage distribué des données sur les nœuds du cluster.

Hadoop:HDFS (Hadoop Distributed File System)

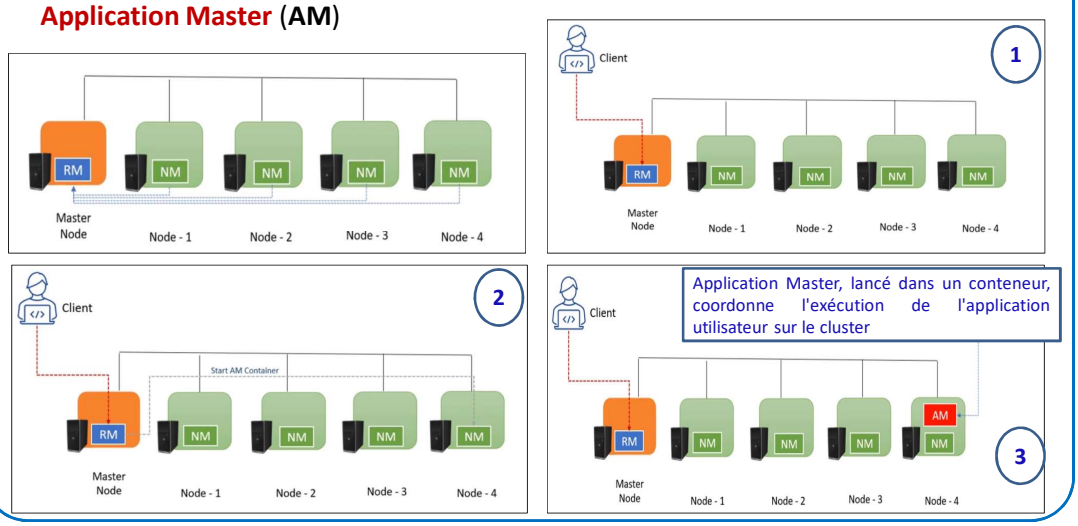
- Architecture ***Maitre/Esclaves*** : **NameNode/DataNodes**



- **NN**: Gère la distribution des fichiers et leurs métadonnées (*nom, taile, id blocs,...*)
- **DN**: Gère les blocs qu'il détient.

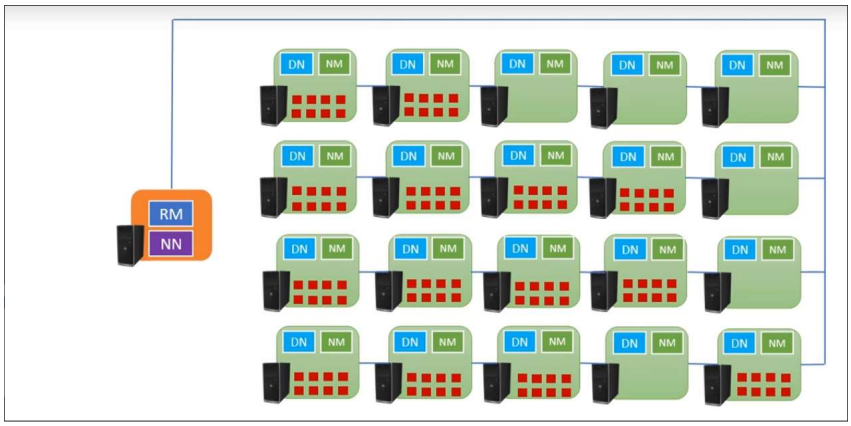
Hadoop: YARN - Yet Another Resource Negociator (Manager)

- Gestionnaire des ressources (*RAM, CPU, Disque*) réparties dans des **conteneurs**.
- Constitué de trois composants: **Resource Manager** (**RM**), **Node Manager** (**NM**) et **Application Master** (**AM**)



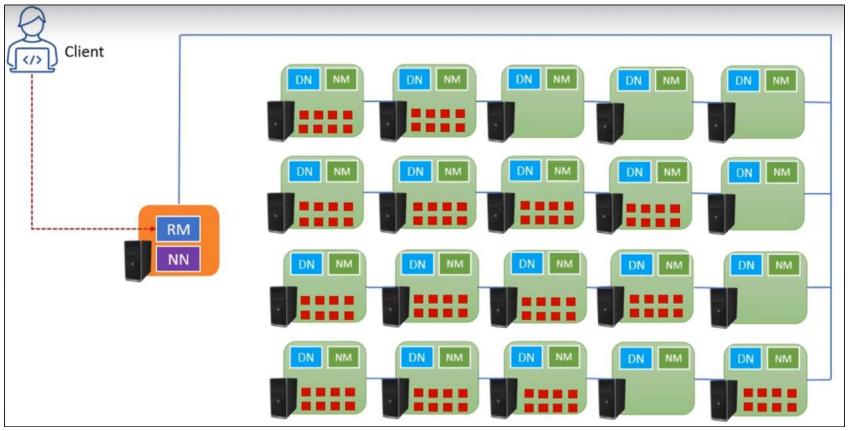
Hadoop: Map/Reduce

- Application utilisateur constituée de deux fonctions principales **Map** et **Reduce**.
- La tâche **Map** est exécutée en premier sur plusieurs nœuds du cluster
- La tâche **Reduce** est exécuté ensuite pour traiter les résultats du **Map**



Hadoop: Map/Reduce

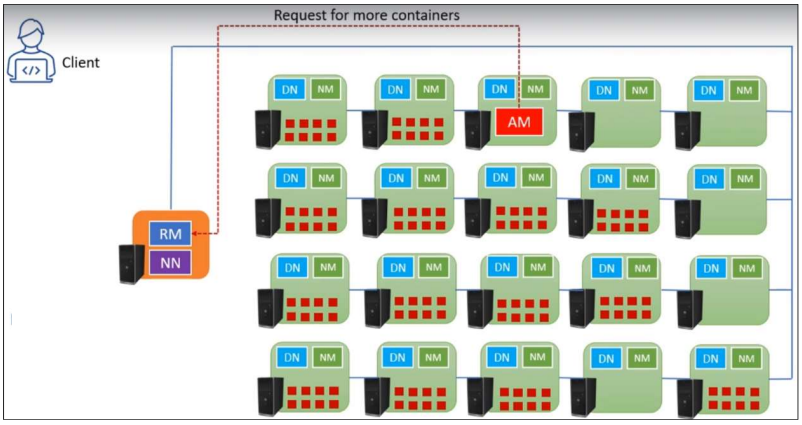
1. Demander au RM d'exécuter d'une application MapReduce



Hadoop: Map/Reduce

2. Le **RM** lancera un **AM** pour coordonner l'exécution des tâches **Map** er **Reduce** de l'application.

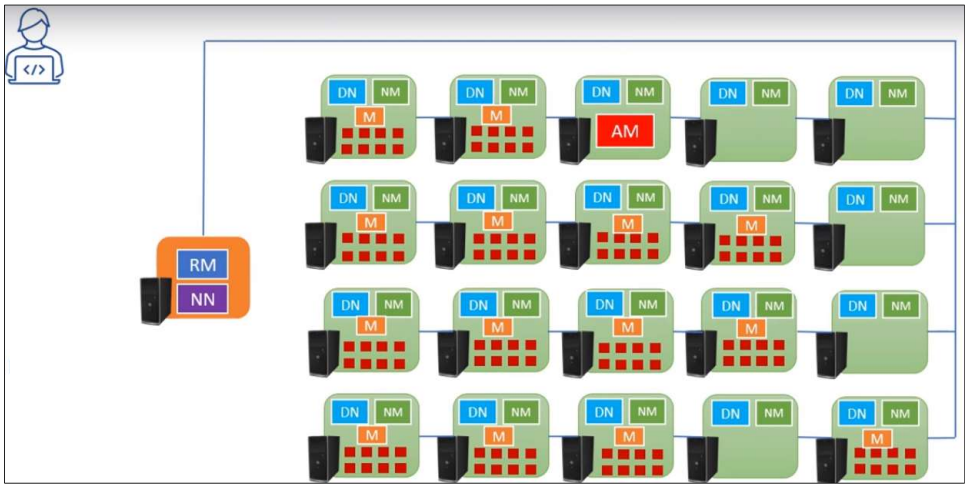
3. L'**AM** demandera au **RM** des **conteneurs** (ressources) nécessaires pour lancer les tâches de l'application



Hadoop: Map/Reduce

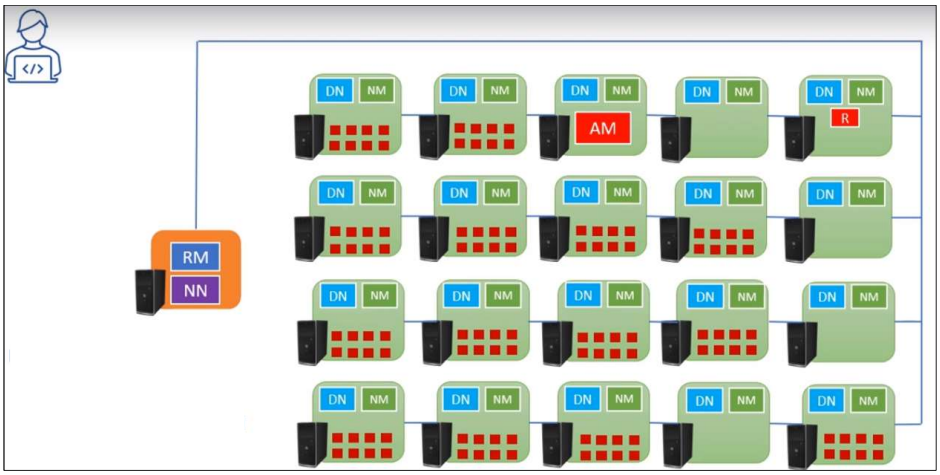
4. Le **RM** demandera aux **NMs** des conteneurs libres à allouer à l'**AM**

5. L'**AM** utilisera ces conteneurs pour lancer d'abord les tâches **Map**



Hadoop: Map/Reduce

6. À la fin de toutes les tâches **Map**, l'**AM** demandera des conteneurs pour lancer des tâche **Reduce** pour fusionner et agréger les résultats des tâches **Map**



Hadoop: Quelques Limites

- Inefficacité pour les petits fichiers:
  - Hadoop n'est pas recommandé pour traiter de petits fichiers (surcharge du NN)
- **Latence élevée** :
  - Hadoop est principalement adapté au traitement batch.
  - Plus de **90%** du temps d'exécution est pour les entrées/sorties disque.
  - Pas idéal pour les applications nécessitant un traitement en temps réel.
  - Pas idéal pour les traitement **itératifs**