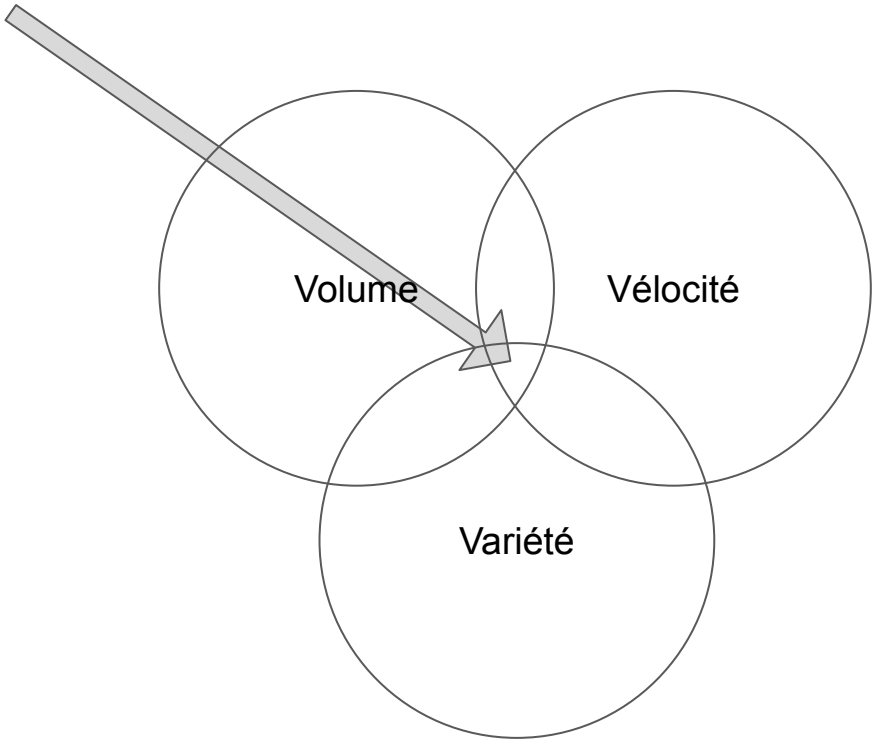
Introduction

Introduction

Nous générons de plus en plus de données:

- Transactions financières
- Événement d'équipement réseaux
- IOT
- Log serveur
- Click Stream (Navigation Web)
- E-mail et formulaire web
- Données issues des réseaux sociaux

Le problème



Le problème : Volume

- Finances
  - Presques 4 milliards d'actions échangées par jour à la bourse de New York
- Facebook
  - 2013 = 10 To /Jour
- Twitter
  - 400 000 tweets écrits par minute
- IoT
  - Tesla a capturé des informations issues de plus d'1 milliard de km

Le problème : Variété

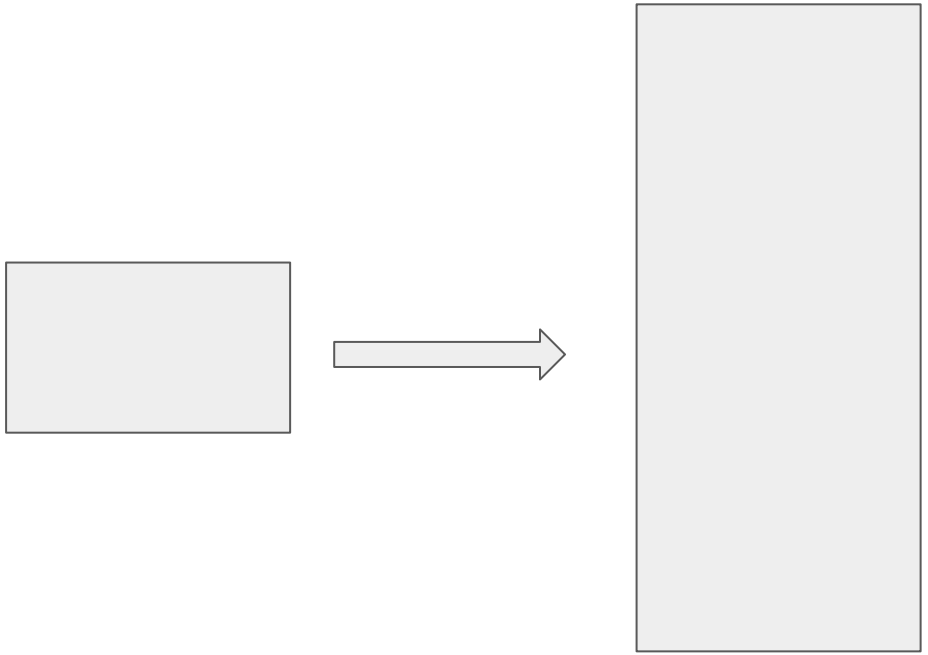
- Donnée Structurée
  - BDD
- Donnée Non Structurée
  - Page Web
  - Log
  - Image
  - Vidéo

Le problème : Vélocité

- Fréquence de mise à jour
- Temps Réel

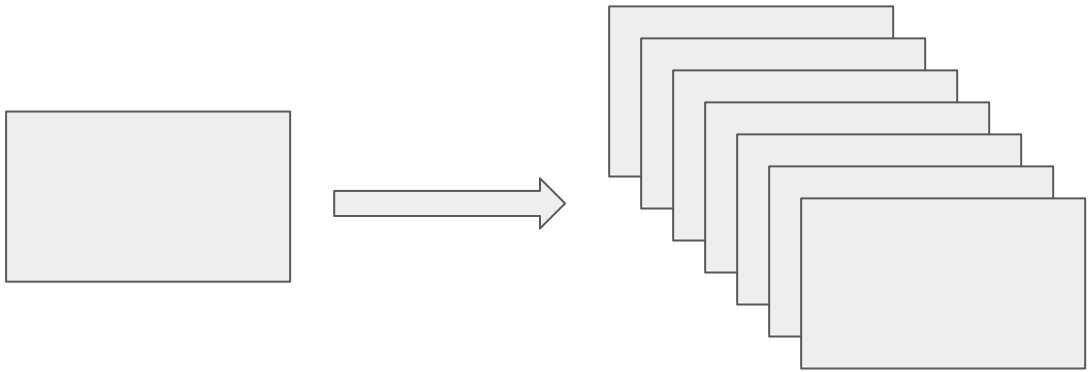
Approche Vertical

- CPU plus rapide
- Plus de mémoire
- Programmation Simple
- Limité par le matériel
- Faible volume



Approche Horizontal

- Gros volume
- Plusieurs machines
- Programmation complexe
  - Gestion des crashs
  - Distribution des calculs



Problème de la donnée

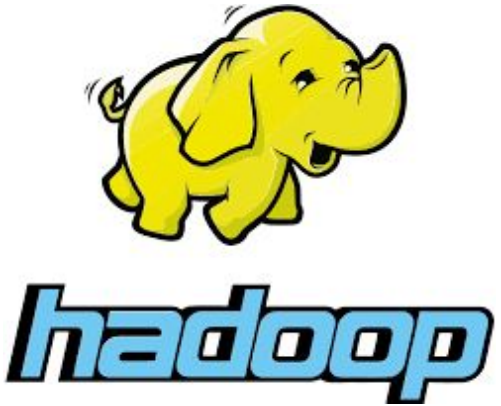
- Traditionnellement centralisé
- Transfert de donnée pour les traitements
- Bande passante réseau limité

Le besoin

- Facilement scalable
- Tolérant à la panne
- Rentable (hardware peu coûteux)

Naissance de Hadoop

- GFS (2003)
- MapReduce (2004)
- Hadoop (2006)
  - HDFS
  - Hadoop MapReduce



Qu’est ce que Hadoop

- Plateforme de Stockage et de Calcul Distribué
  - grand volume de donnée de manière résiliente
  - permet de se concentrer sur les problèmes business et non infra
- Différent cas d’usage possible
  - Extract Transform Load
  - Business Intelligence
  - Stockage
  - Machine Learning
  - ...

Hadoop est scalable

- ajout facile de noeud
- augmentation de ressource = augmentation de performance
- gestion des échecs
  - le système continue de fonctionner
  - la tâche est attribuée à un autre noeud
  - pas de perte de donnée car réplication

Hadoop écosystème

- riche
- open source
- grandissant
  - Spark (2014)
  - Kafka (2012)

L’écosystème Hadoop

HDFS - Hive - YARN - Oozie

HDFS

Hadoop Distributed File System

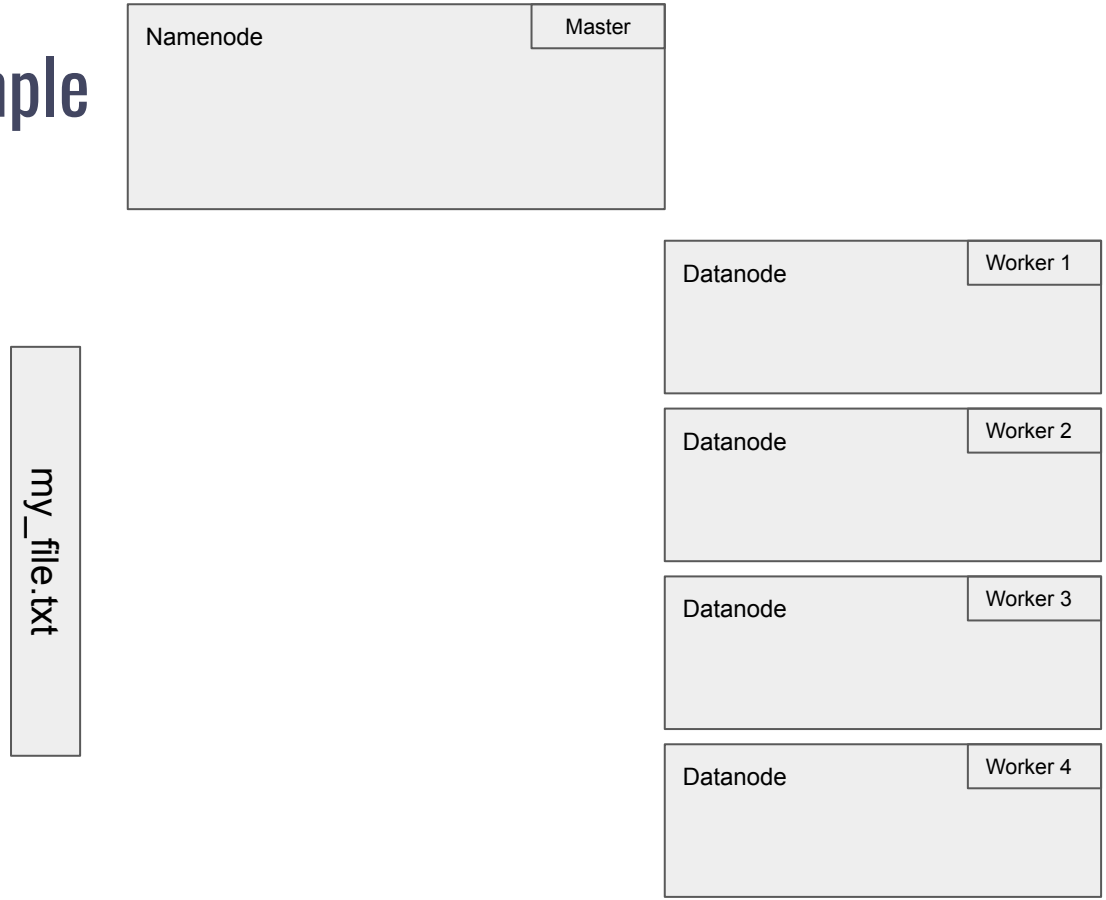
HDFS

- Comme un système de fichier classique
- distribué
- haute disponibilité
- résilience

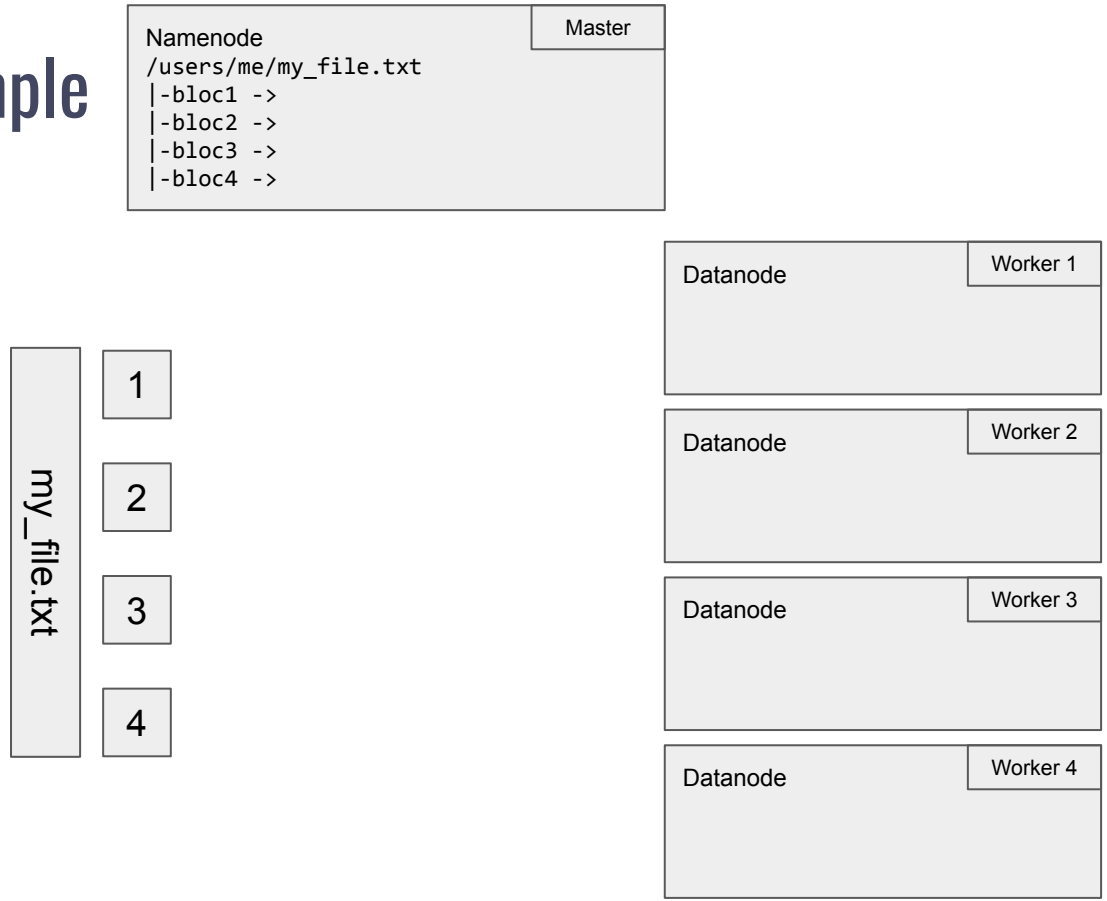
HDFS utilisation et limitation

- Gros fichier (bloc de 128 Mo)
- Pas de modification des fichiers (append accepté)
- Optimisé pour la lecture séquentielle de fichier

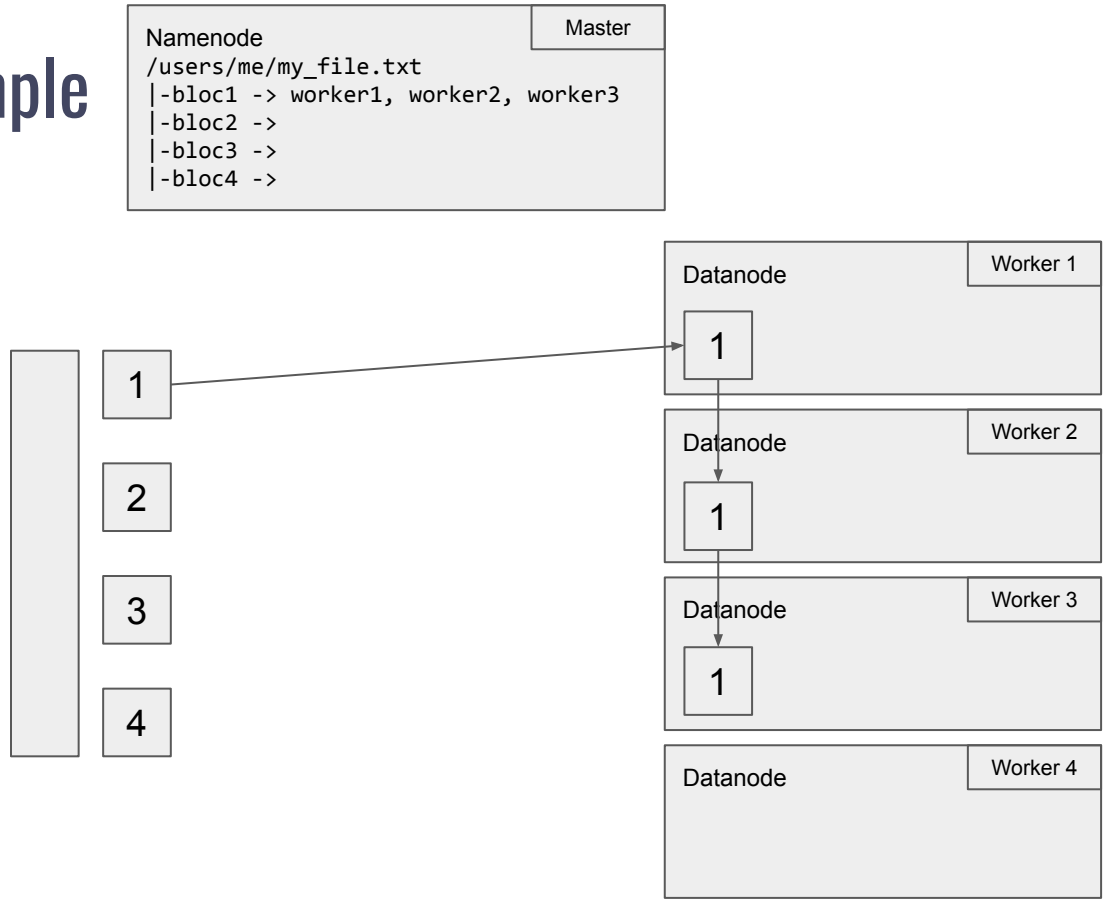
HDFS exemple



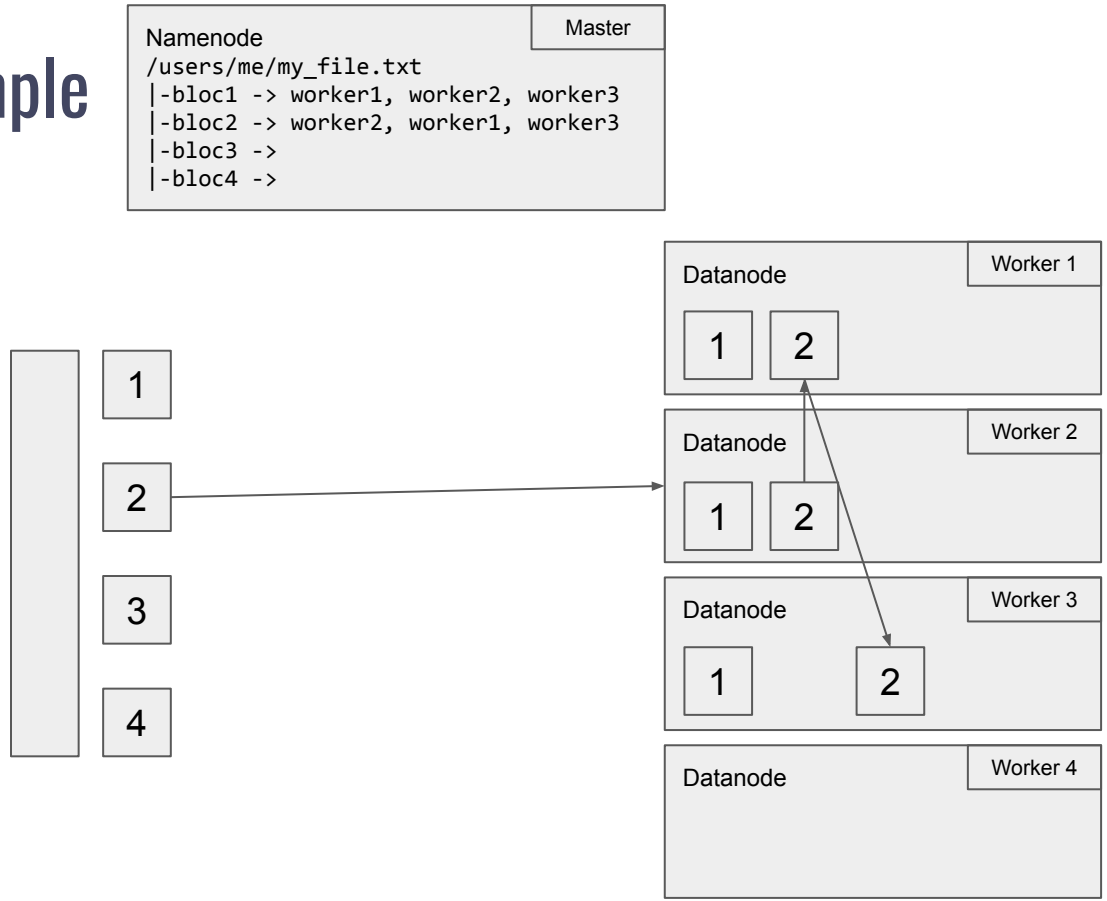
HDFS exemple



HDFS exemple



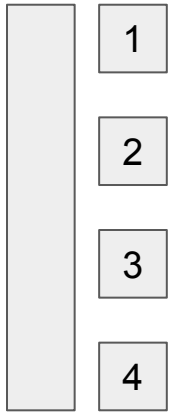
HDFS exemple



HDFS exemple

Namenode — Master

```
/users/me/my_file.txt
|-bloc1 -> worker1, worker2, worker3
|-bloc2 -> worker2, worker1, worker3
|-bloc3 -> worker2, worker3, worker4
|-bloc4 -> worker1, worker3, worker4
```



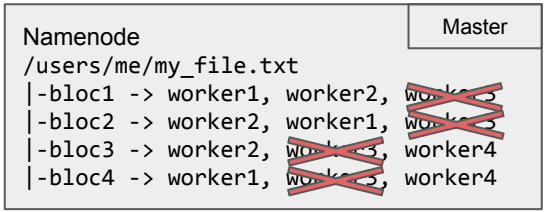
Datanode — Worker 1: 1 2 4

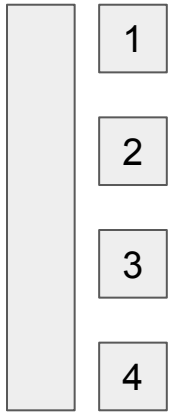
Datanode — Worker 2: 2 1 3

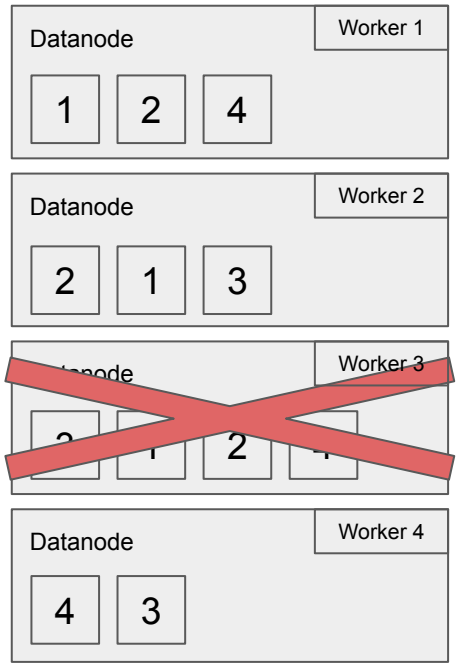
Datanode — Worker 3: 3 1 2 4

Datanode — Worker 4: 4 3

HDFS exemple







Hive

Hive

- Vue SQL sur les fichiers du HDFS
- Data Warehouse
- HiveQL ~= SQL
- JDBC

Hive

- Génère du code spark ou mapreduce
- Donnée Structurée (Parquet, Avro, ...)
- Pas besoin de programmer (SQL)

Hive vs RDBMS

	RDBMS	Hive
Langage de requête	SQL	SQL
Update	Oui	Non
Latence	Faible	Élevé
Scalable	Difficile	Facile
Coût	Élevé	Faible

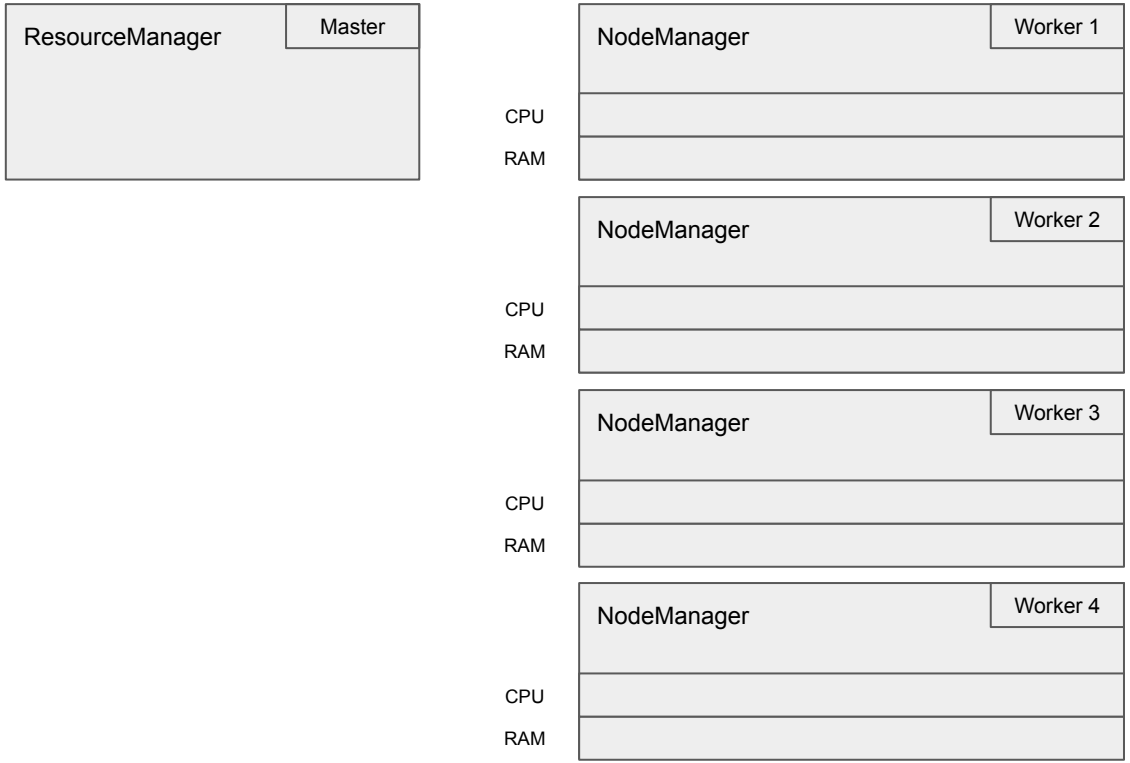
YARN

Yet Another Resource Negotiator

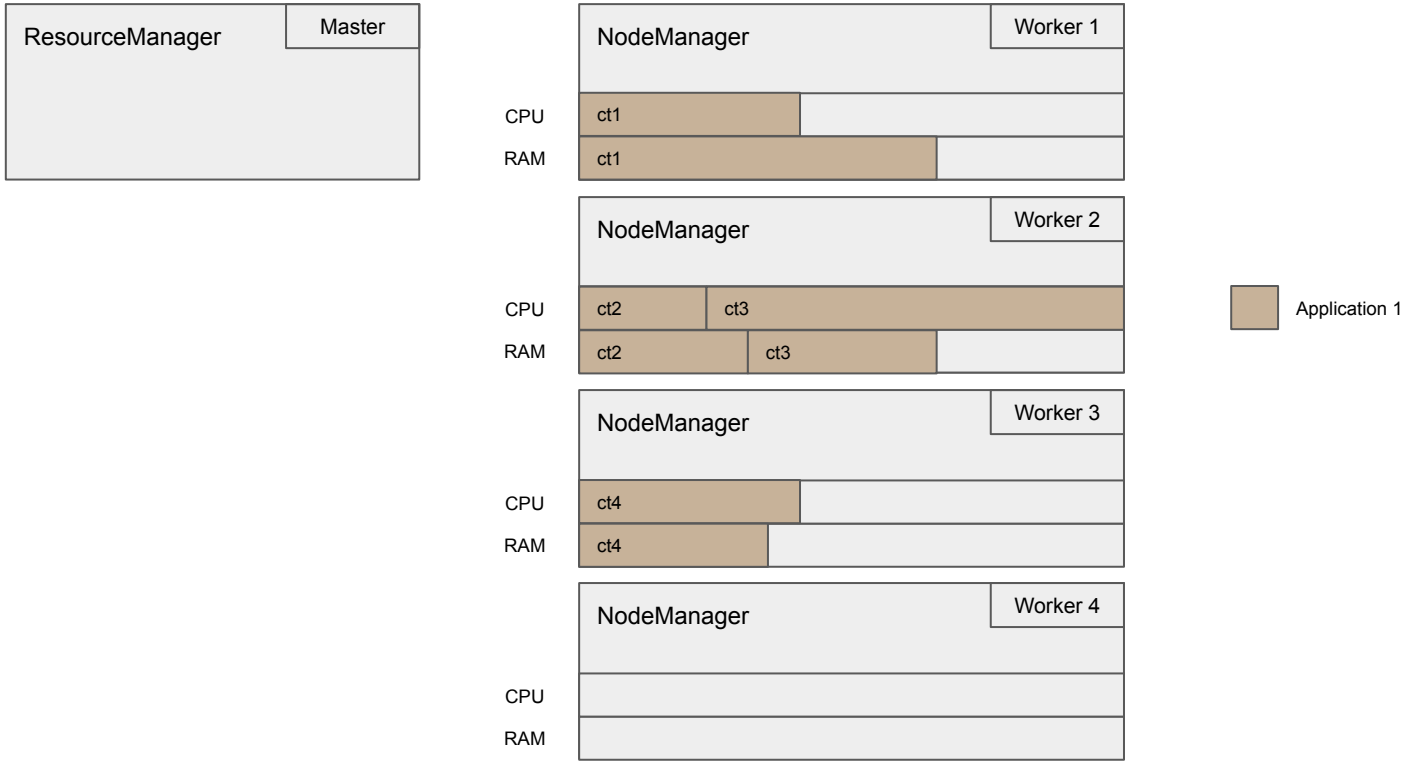
YARN

- Yet Another Resource Negotiator
- Gère la répartition des ressources de calcul
- Conteneur = CPU + RAM
- Queue pour la répartition équitable

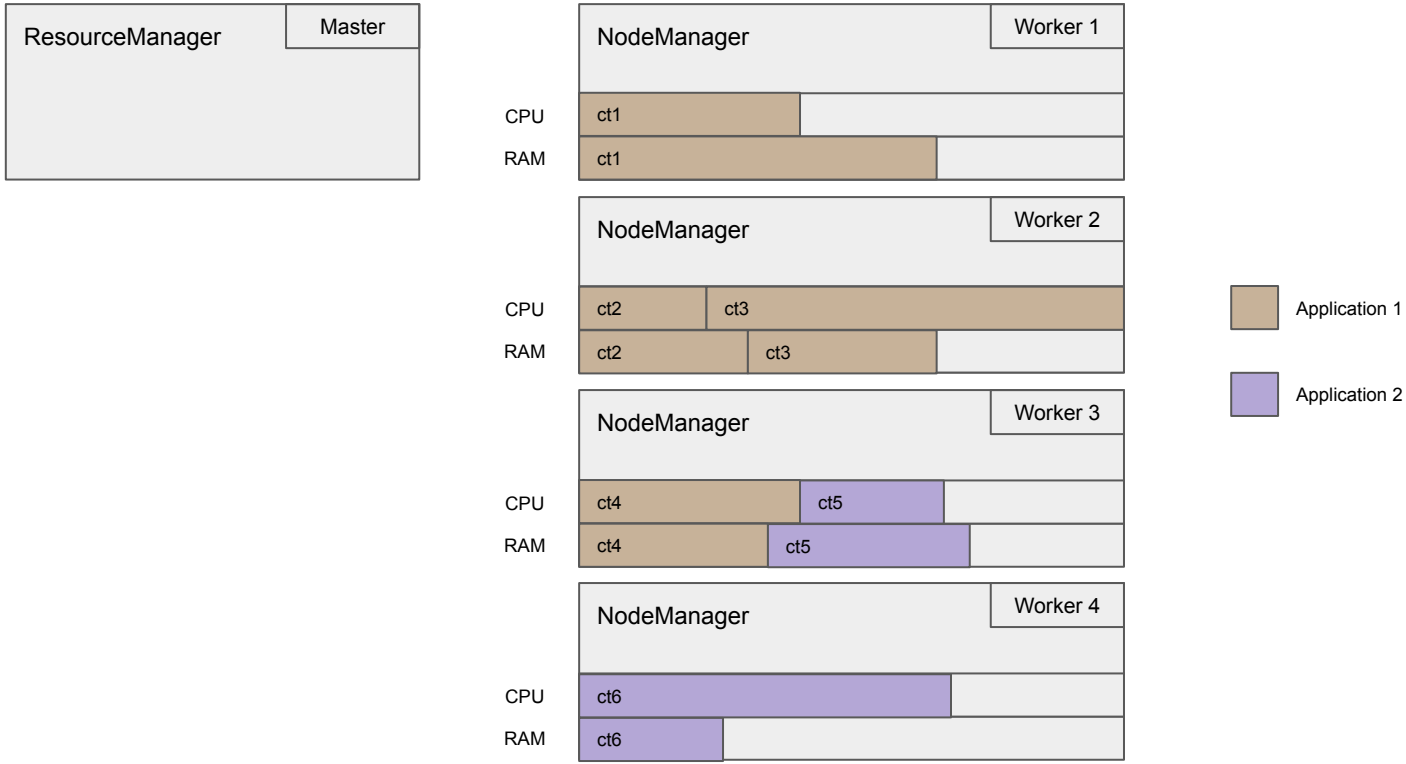
YARN


ResourceManager
Master
NodeManager
Worker 1
CPU
RAM
NodeManager
Worker 2
CPU
RAM
NodeManager
Worker 3
CPU
RAM
NodeManager
Worker 4
CPU
RAM

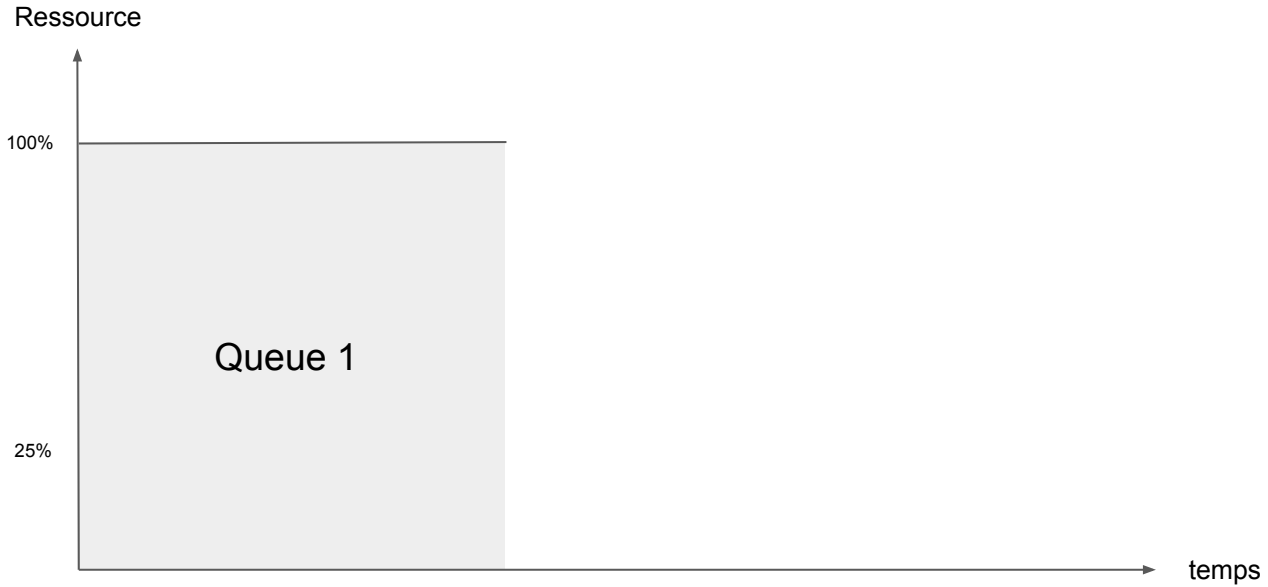
YARN


ResourceManager
Master
NodeManager
Worker 1
CPU
RAM
ct1
ct1
NodeManager
Worker 2
CPU
RAM
ct2
ct3
ct2
ct3
NodeManager
Worker 3
CPU
RAM
ct4
ct4
NodeManager
Worker 4
CPU
RAM
Application 1

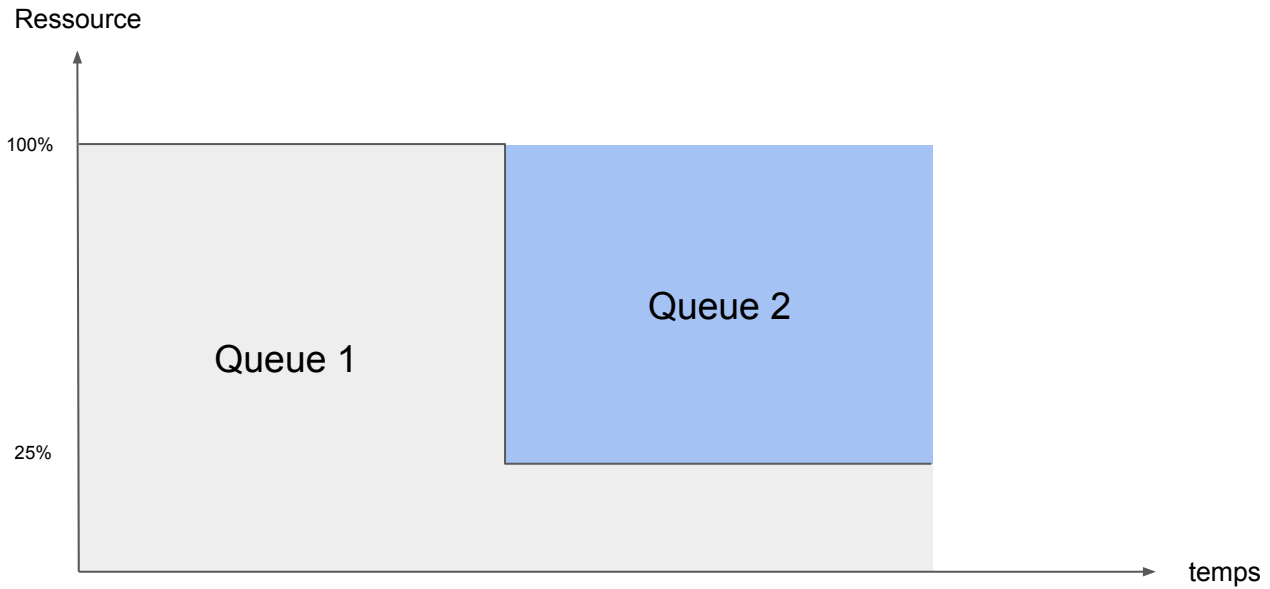
YARN


ResourceManager
Master
NodeManager
Worker 1
CPU
RAM
ct1
ct1
NodeManager
Worker 2
CPU
RAM
ct2
ct3
ct2
ct3
NodeManager
Worker 3
CPU
RAM
ct4
ct5
ct4
ct5
NodeManager
Worker 4
CPU
RAM
ct6
ct6
Application 1
Application 2

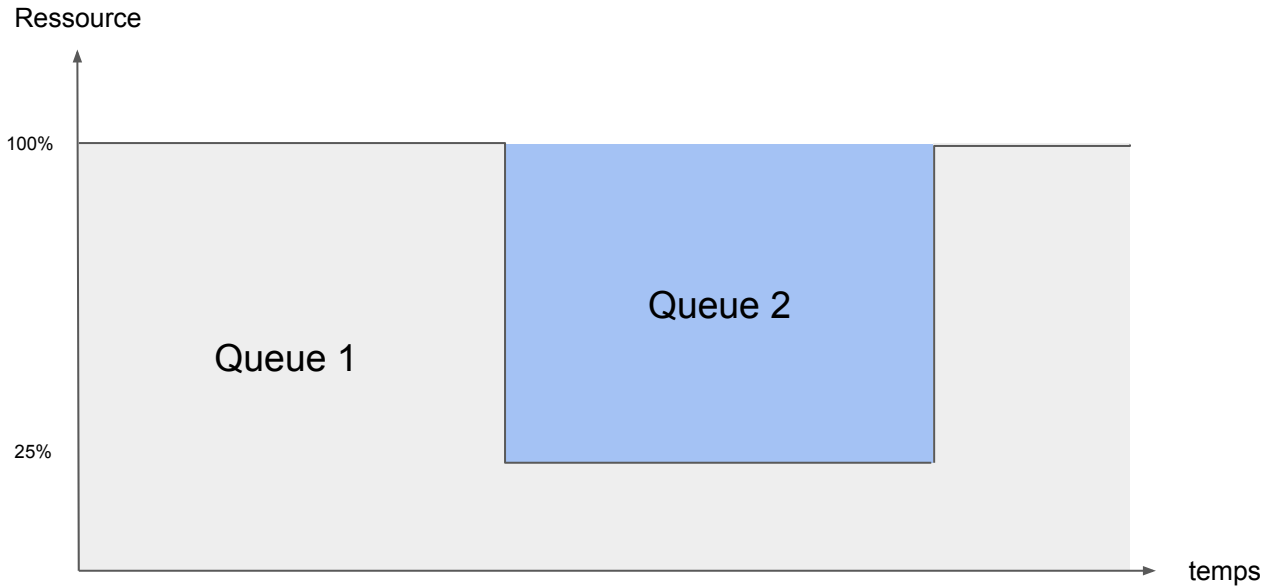
YARN Queue



YARN Queue



YARN Queue

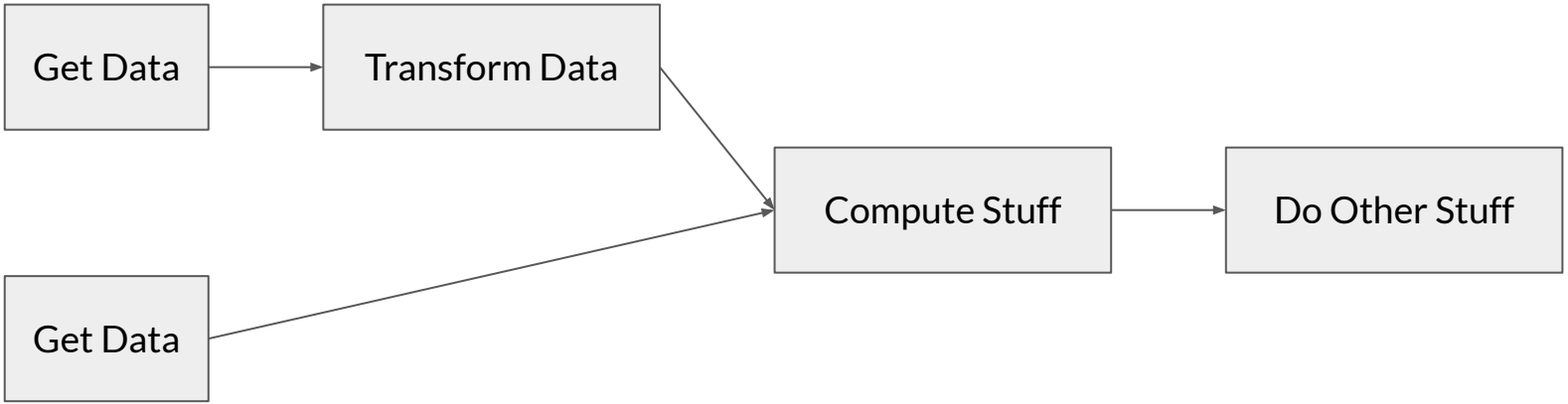


Oozie

Oozie

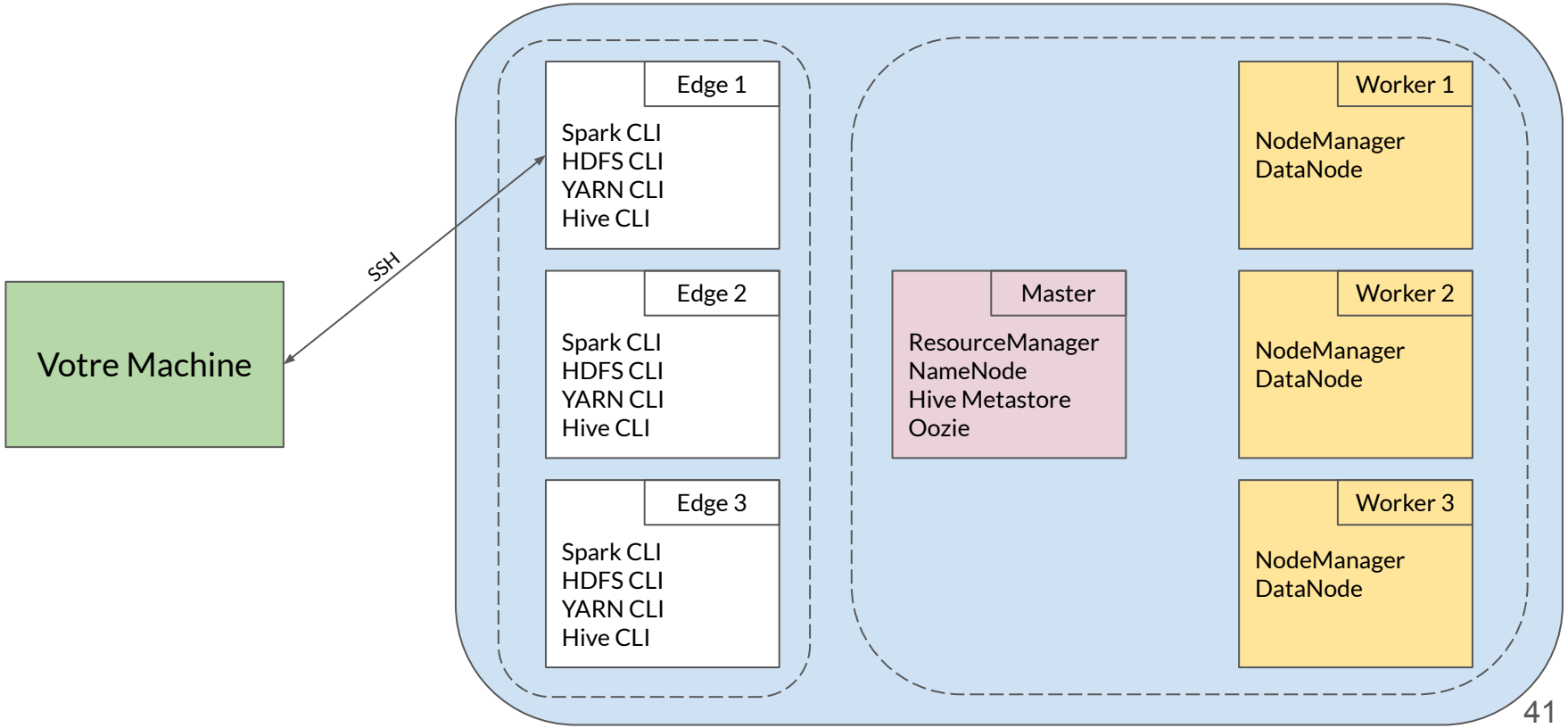
- Workflow
  - suite de job ayant des dépendances
- Coordinator
  - lancement de workflow à heure fixe
  - lancement suite à événement (arrivé d'un fichier)

Oozie : Workflow



Organisation d’un Cluster

Organisation d'un cluster



Map Reduce

Map Reduce

Modèle de programmation inventé en 2004 par Google. Il est utilisé pour paralléliser les calculs en tirant parti d'un système de fichier distribué.

- Map
  - Projection (SELECT)
  - Filtre (WHERE)
- Reduce
  - Aggregation (GROUP BY)
  - Jointure (JOIN)

Compter des Lego avec map reduce

Combien y a-t-il de brique 8 et 4 dans ce tas de cartes ?

1. distribuer des lego à chaque node
2. chaque node filtre les 8 et 4
3. chaque node compte les 8 et 4
4. tous les comptes de 8 sont envoyés à un node
5. tous les comptes de 4 sont envoyés à un node
6. Le “Node 8” ajoute les différents comptes
7. Le “Node 4” ajoute les différents comptes