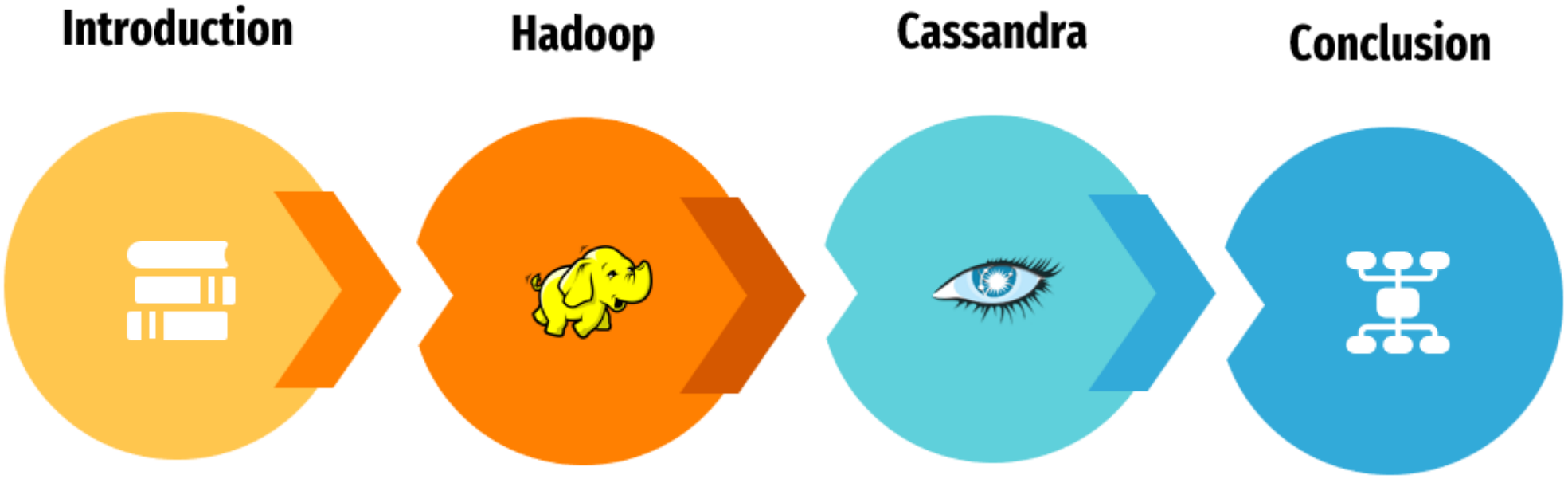
Hadoop, Cassandra

Guillaume SOL & Mathieu TRITZ

Sommaire



Introduction

Big Data - Qu’est-ce que c’est ?

Big Data - Exemples

De combien de données parlons-nous ?

- Facebook: **40 Po** de données | **100 To / jour**
- Yahoo : **60 Po** de données
- Twitter : **8 To / jour**
- eBay : **40 Po** de données | **50 To / jour**

Pour vous donner un ordre d'idée, chaque **minute** :

- Google est sollicité près de **4 millions de fois**
- **4,5 millions de vidéos** visionnées sur YouTube
- **188 millions d'emails** échangés

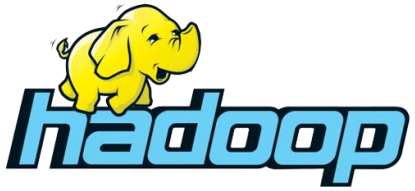
Big Data - Problématique

Problèmes du Big Data :

- Des volumes de données trop conséquents
  = Données non exploitables
- Difficile à stocker dans les SGBDR traditionnels
- Limite de performances
- Coût du stockage
- Données non structurées

Big Data - Ecosystème

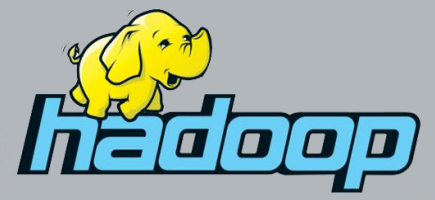
Construction d'un écosystème pour gérer ces masses de données

Hadoop ? 

- Pas une technologie Big Data mais des technologies Big Data
  - Hadoop est une technologie Big Data
  - mais le Big Data ne se résume pas à Hadoop et son éco-système

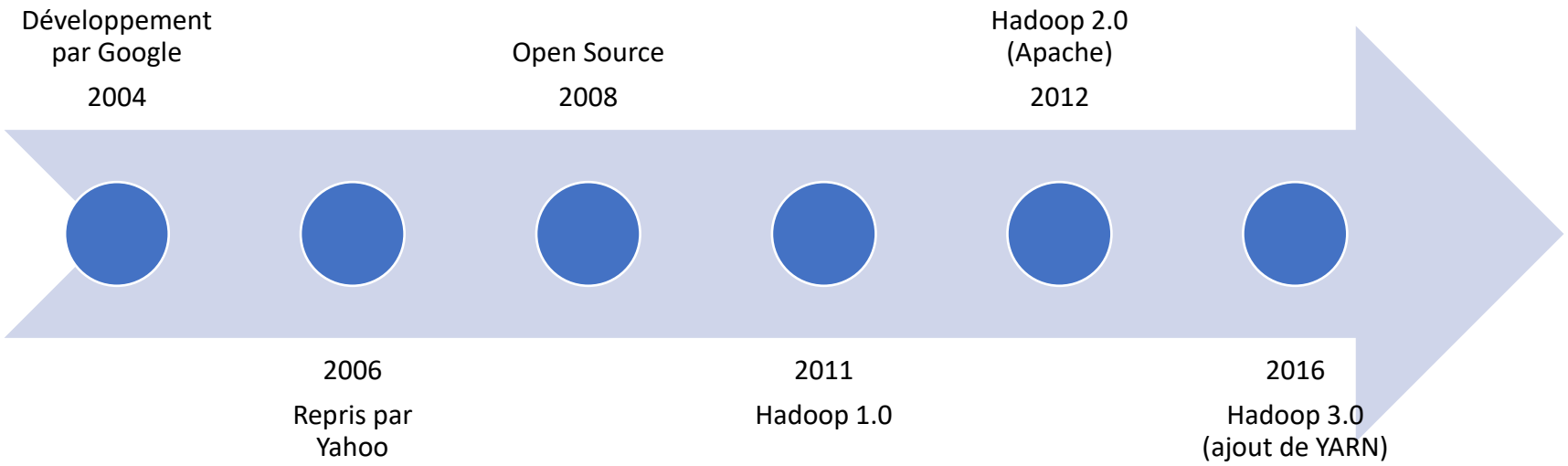
- Des outils spécifiques pour des cas d'usages précis

Hadoop



Hadoop - Qu’est-ce que c’est ?

- Framework Java Open Source
- Permet de développer des applications distribués et échelonnables



Hadoop – Solution

Créer pour répondre à la problématique du **Big Data**

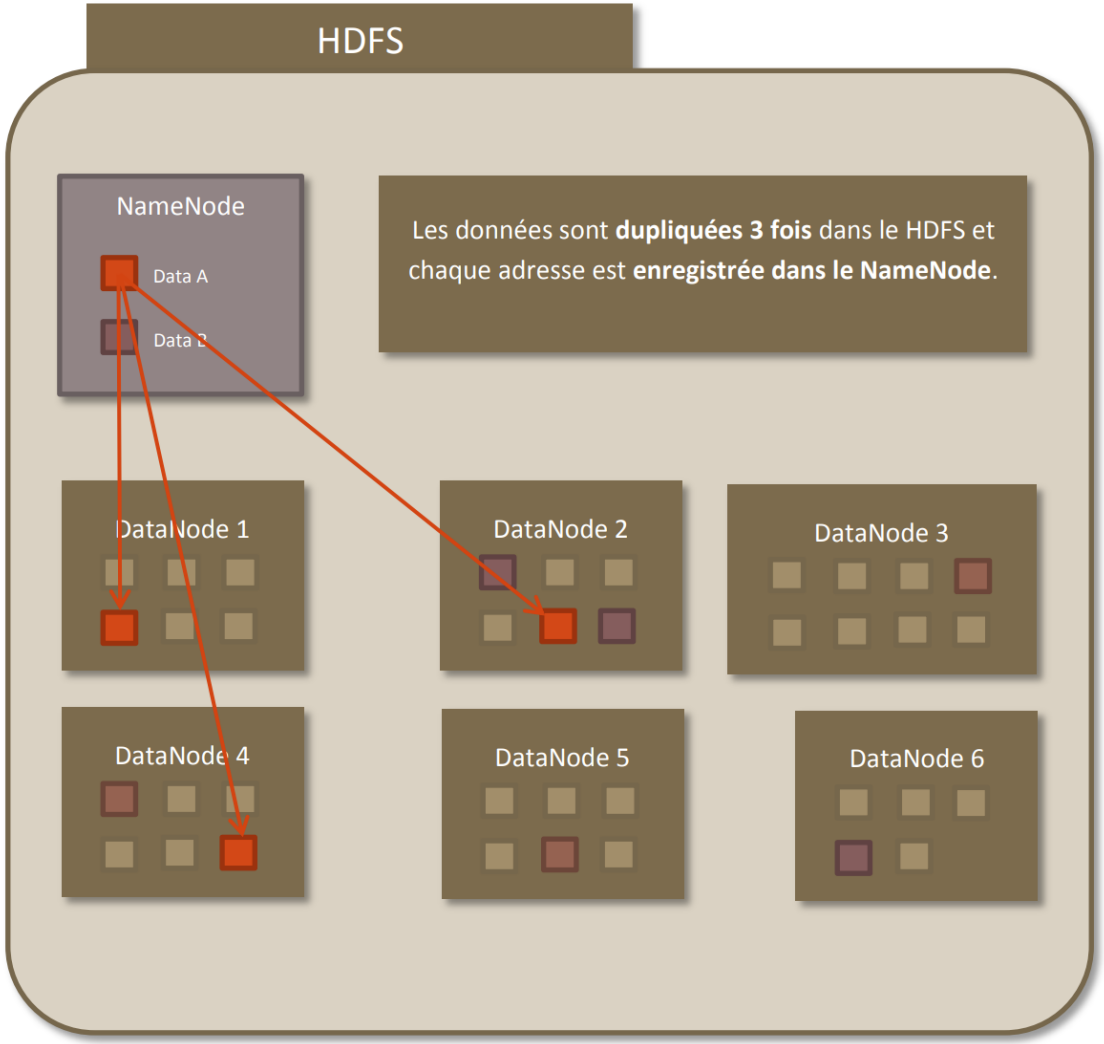
- Repose sur des modèles de programmation simples
- Traitement d'immenses volumes de données
- Rend les données disponibles sur des machines locales.

Particularité d'Hadoop :

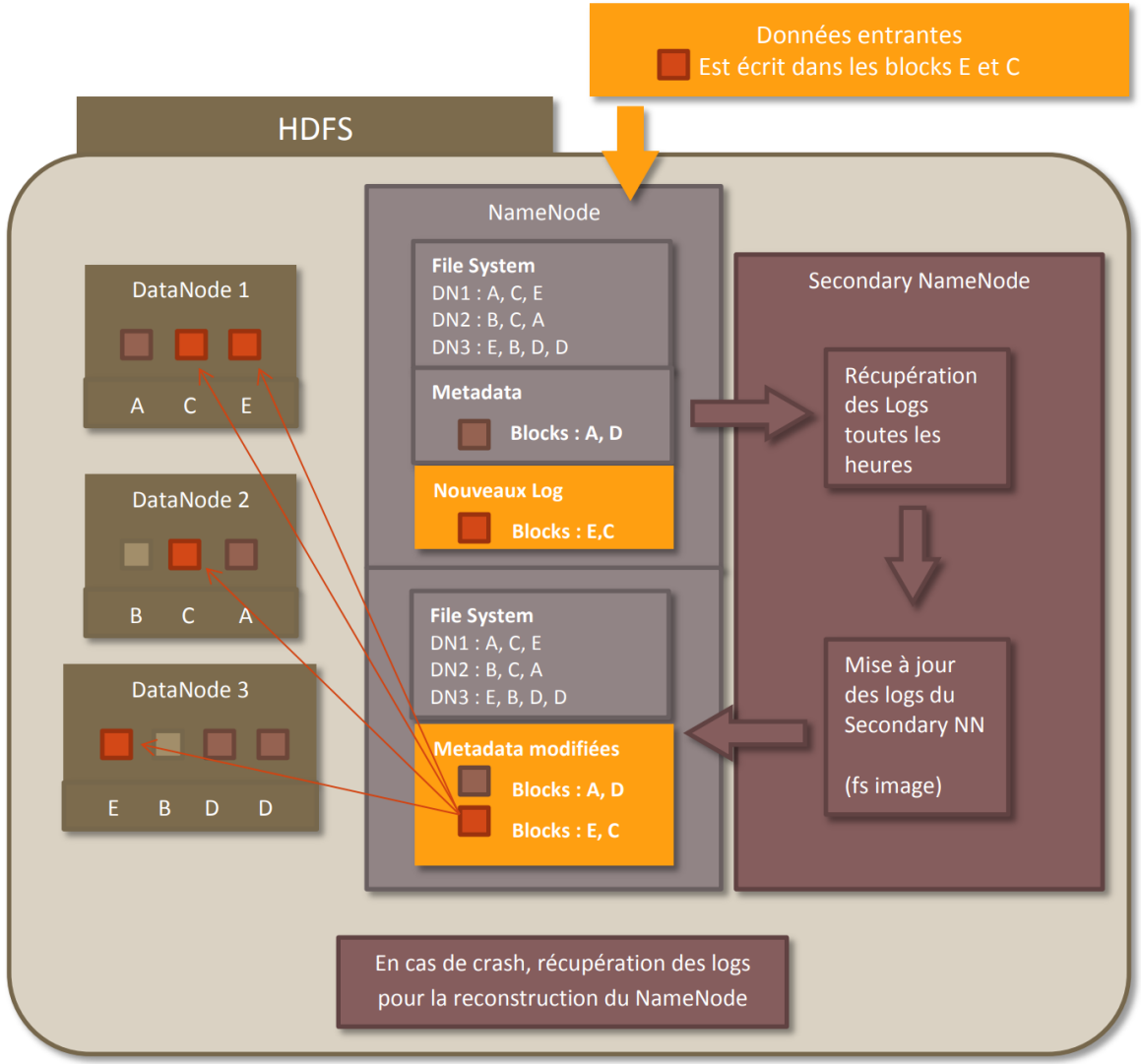
- **HDFS** (Hadoop Distributed File System) : modèle de stockage distribué.
- **MapReduce** : algorithme permettant d'effectuer des calculs parallèles et d'avoir des données fiables.

(2 composants principaux)

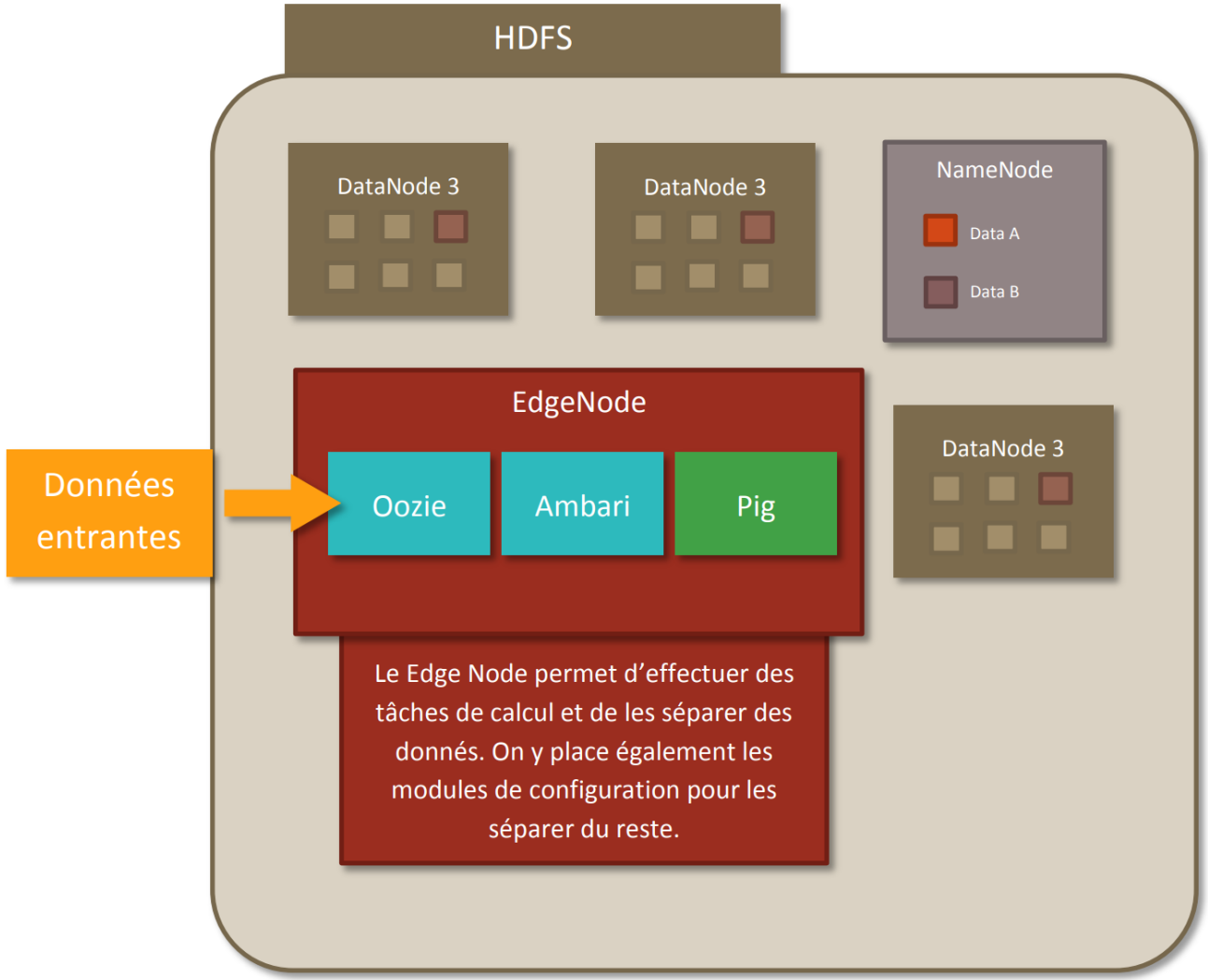
Hadoop - HDFS


HDFS
NameNode
Data A
Data B
Les données sont **dupliquées 3 fois** dans le HDFS et chaque adresse est **enregistrée dans le NameNode**.
DataNode 1
DataNode 2
DataNode 3
DataNode 4
DataNode 5
DataNode 6

Hadoop - HDFS

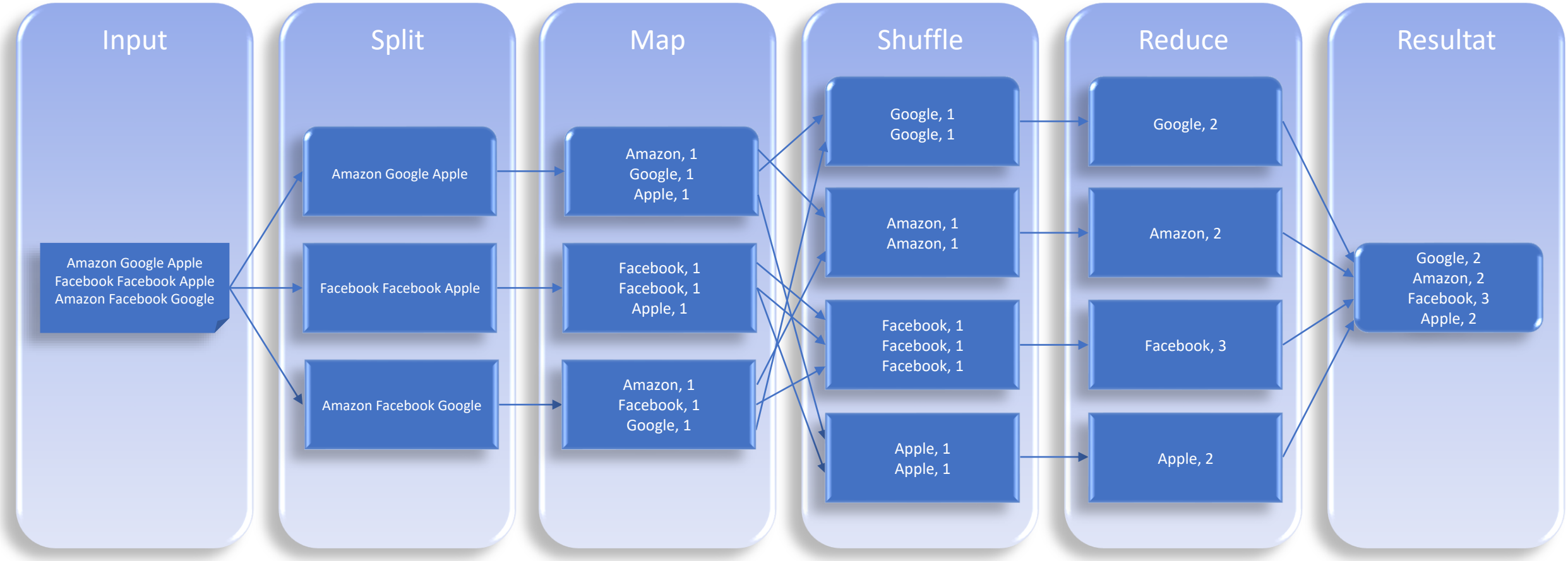

Données entrantes
Est écrit dans les blocks E et C
HDFS
NameNode
File System
DN1 : A, C, E
DN2 : B, C, A
DN3 : E, B, D, D
Metadata
Blocks : A, D
Nouveaux Log
Blocks : E,C
File System
DN1 : A, C, E
DN2 : B, C, A
DN3 : E, B, D, D
Metadata modifiées
Blocks : A, D
Blocks : E, C
DataNode 1
A C E
DataNode 2
B C A
DataNode 3
E B D D
Secondary NameNode
Récupération des Logs toutes les heures
Mise à jour des logs du Secondary NN
(fs image)
En cas de crash, récupération des logs pour la reconstruction du NameNode

Hadoop - HDFS

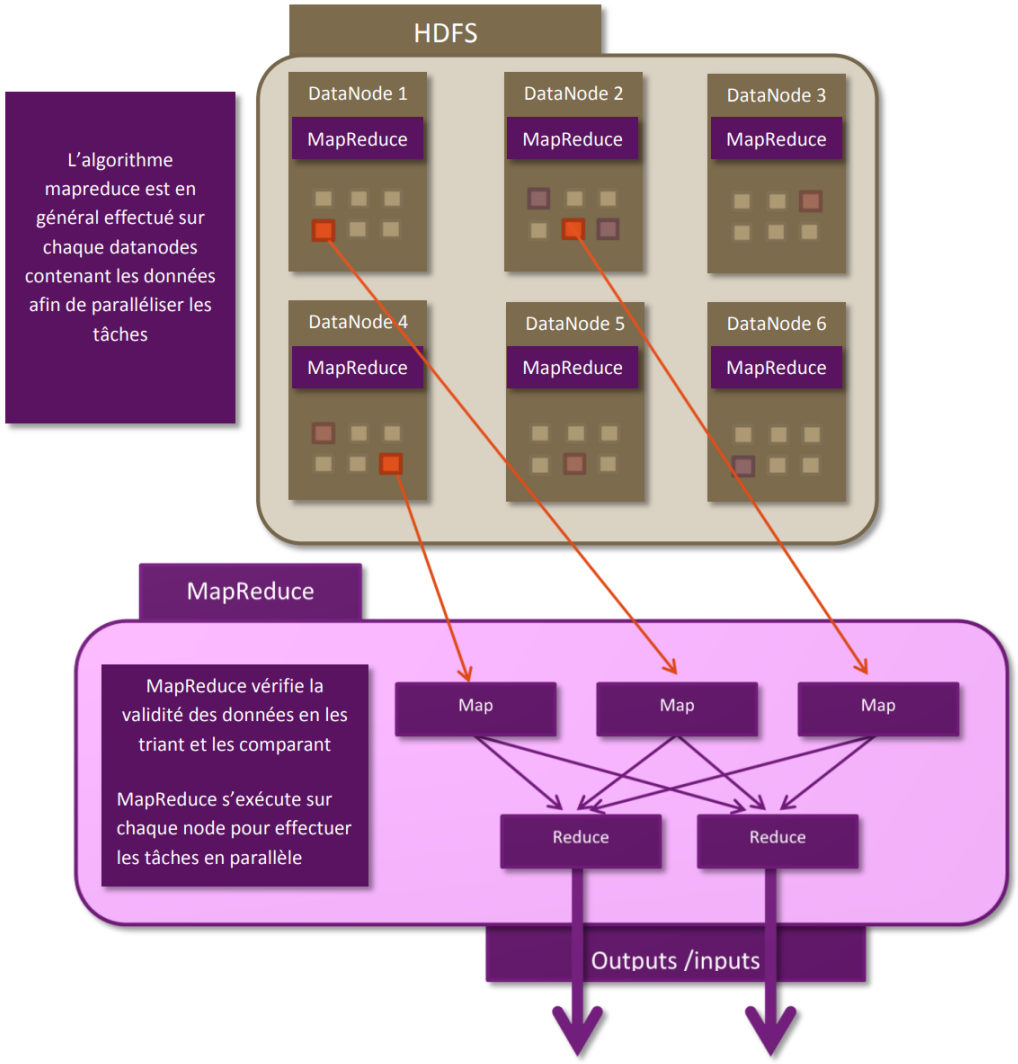

HDFS
DataNode 3
DataNode 3
NameNode
Data A
Data B
EdgeNode
Oozie
Ambari
Pig
DataNode 3
Données entrantes
Le Edge Node permet d'effectuer des tâches de calcul et de les séparer des donnés. On y place également les modules de configuration pour les séparer du reste.

Hadoop – MapReduce

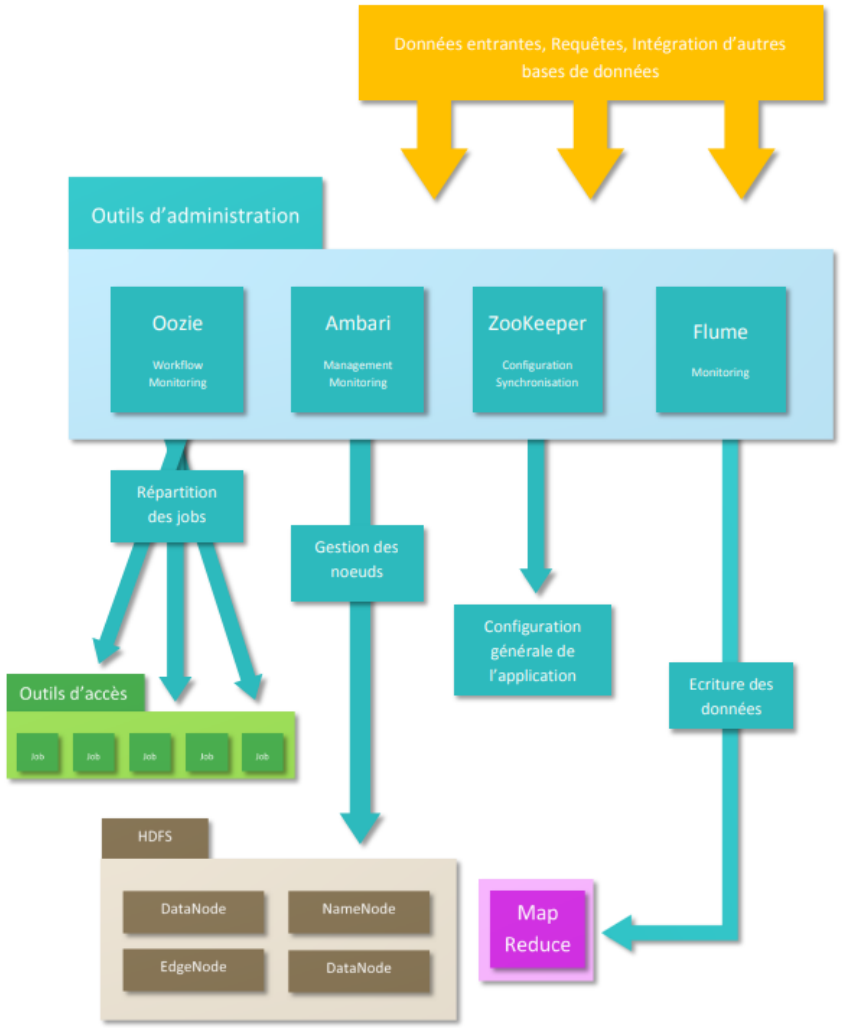
(Vision Wordcount)



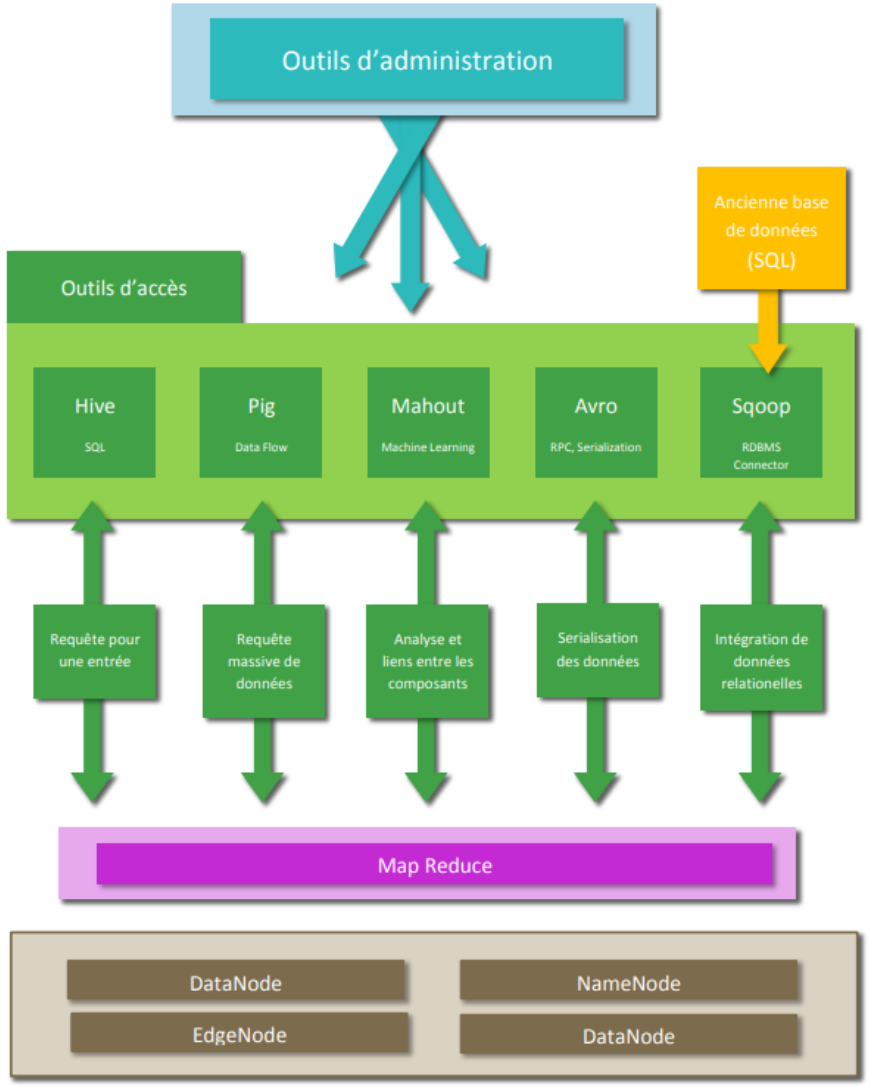
Hadoop - MapReduce



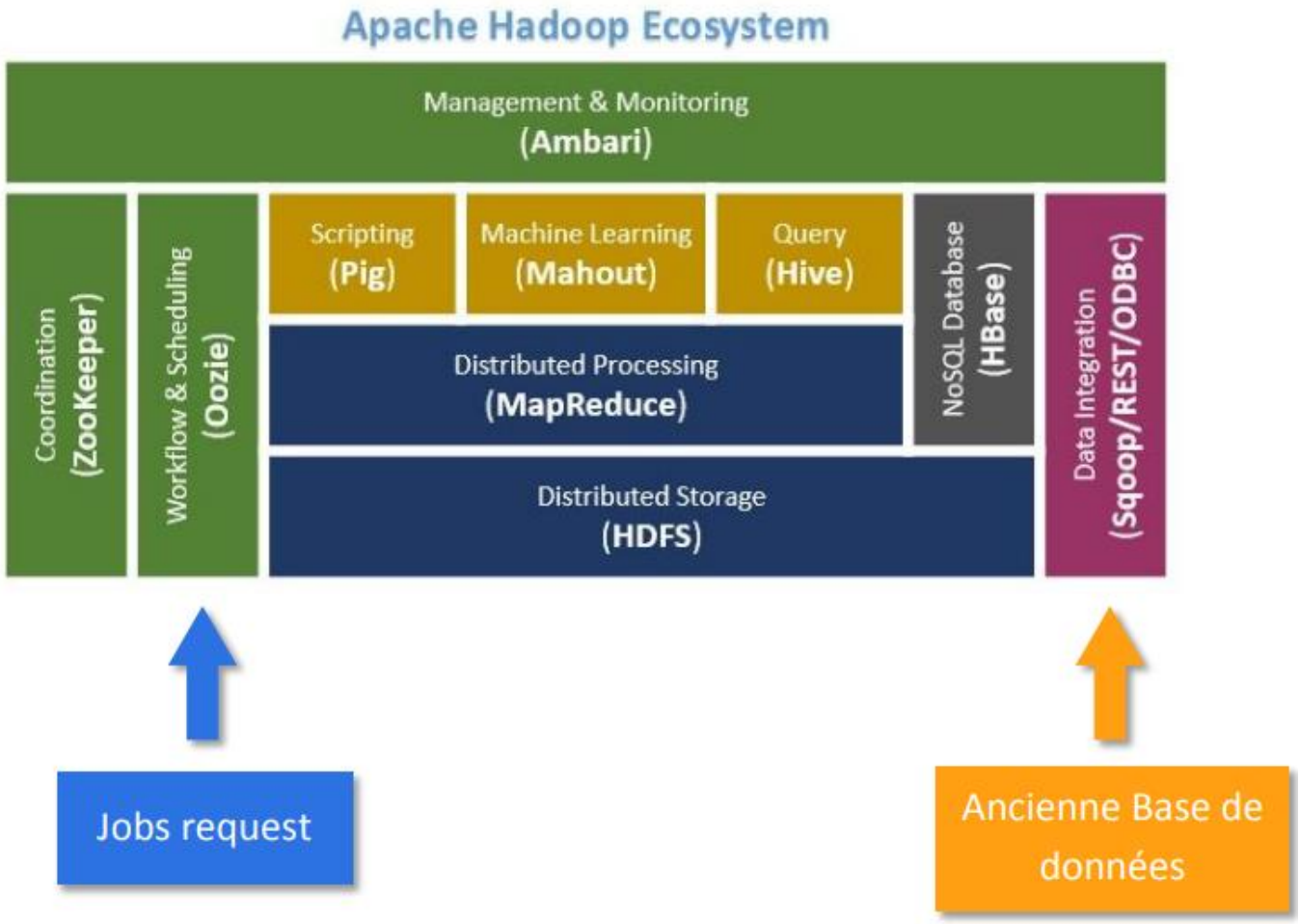
Hadoop - Outils de gestion et configuration de l'application



Hadoop - Outils d’accès aux données



Hadoop - Ecosystème



Hadoop - HBase

SGBD non relationnel (ou NoSQL)

Particularités :

- Open source | Java
- HDFS -> Tolérant aux pannes
- MapReduce
- Requêtage rapide

Pour résumé sur Hadoop

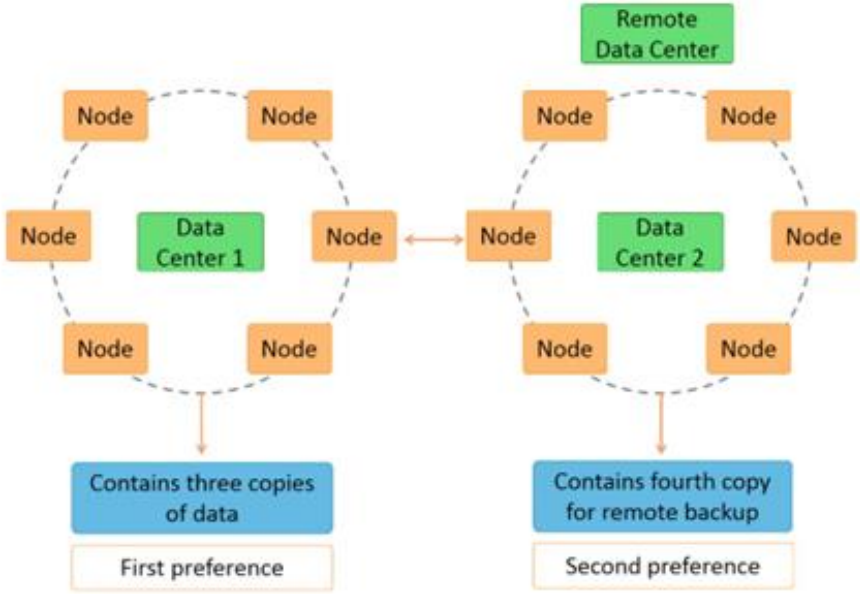
- Framework open-source
- Connu pour sa tolérance aux pannes et sa fonctionnalité de haute disponibilité
- Les clusters Hadoop sont évolutifs
- Facile à utiliser
- Il assure un traitement rapide des données grâce au traitement distribué
- Hadoop est rentable
- La fonctionnalité de localisation des données Hadoop réduit l'utilisation de la bande passante du système

NoSQL

Cassandra

Architecture

Keys	Values
2398239	{ "name": "Michał", "Age": "31"}
2398240	"Lorem ipsum dolor sit amet"
2398241	{ "name": "Marlon Brando", "Profession": "Actor"}
2398242	42



Pourquoi choisir Cassandra ?

Adaptabilité

Hybride

Réplication des données

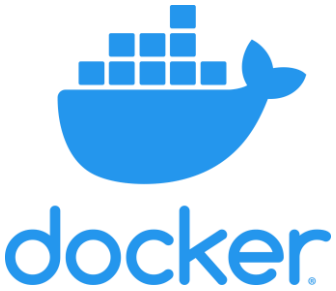
Performant

Distribué

Extensibilité

Comment l'utiliser ?

Installation



Application

Qui utilise Cassandra ?

Pour résumer sur Cassandra

- Tolérant aux pannes grâce à la réplication des données
- Architecture peer to peer
- Facteur de réplication proportionnelle au nombre de nœuds
- Echange simplifié par API

Conclusion

Hadoop	Cassandra
Framework évolutif pouvant être utilisé sur du matériel à faible cout et déployé sur un seul centre de données	Déploiement de manière distribué sous forme de cluster
Utilisé principalement pour le traitement de données volumique (niveau pétaoctets)	Utilisé principalement pour le traitement de données en temps réel et le traitement de mégadonnées
Architecture maitre esclave	Architecture peer to peer
Latence élevé	Faible latence
Facteur de réplication de 3	Facteur de réplication qui dépend du nombre de nœuds
Protocol UDP et TCP	Protocol gossip
Langage similaire au SQL (HQL)	Langage de requêtage simplifié

Sources

Documentation, **Cassandra apache officiel**

Talend

Book, **Hadoop Illuminated**

Awesome-Hadoop, **GitHub**

Documentation, **Apache Hadoop**

Apache Hadoop In Theory And Practice, **Spotify**

Ressources internes, **SFR**

Merci de votre attention

SGBDR / NoSQL

(transaction une question de chimie)

ACID (Base relationnelle)	BASE (NoSQL)
Atomicity : L'ensemble des opérations dans une action est insécable (tout ou rien)	**Basic Availability** : Chaque requête est garantie d'avoir une réponse
Consistency : Une transaction ne peut pas laisser la base de données dans un état incohérent	**Soft state** : L'état du système peut changer au cours du temps
Isolation : Une transaction ne peut interférer avec une autre transaction	**Eventual consistency** : La base peut momentanément être inconsistante mais sera consistante au final
Durability : Système dans un état stable et durable même après redémarrage	

SGBDR / NoSQL
(la philosophie)

ACID (Base relationnelle)	BASE (NoSQL)
Vision pessimiste : tout peut aller de travers	Vision optimiste : tout va s'arranger au final
Beaucoup de verrouillages et déverrouillages	Garder les choses simples et éviter d'utiliser les verrouillages
Pas d'affichage de la donnée pendant une transaction	Ne jamais bloquer en écriture
	Se concentrer sur le débit même si la donnée momentanément n'est pas juste

⇒ ACID est bien adapté lorsqu'il y a besoin d'une donnée juste et fiable

⇒ BASE donne la priorité à ne jamais bloquer une écriture