



Introduction aux bases de données NOSQL

SOMMAIRE DU CHAPITRE

- Big Data et motivations
- Limites des SGBD relationnels
- Modèles de distribution
- Le Big Data analytique
- Les bases de données NoSQL

SOMMAIRE DU CHAPITRE

Big Data et motivations

Limites des SGBD relationnels

Modèles de distribution

Le Big Data analytique

Les bases de données NoSQL



Motivation

Comment passer à l'échelle ?

- Des millions d'utilisateurs interagissant avec un système donné
- Explosion du volume de données à stocker et à traiter
- 7 To par seconde prévus pour le radiotélescope « Square Kilometre Array »

Idée:

- Passer d'un système large échelle vertical (« dopage » du serveur) vers un système large échelle horizontal (ajout de machines)

Motivation

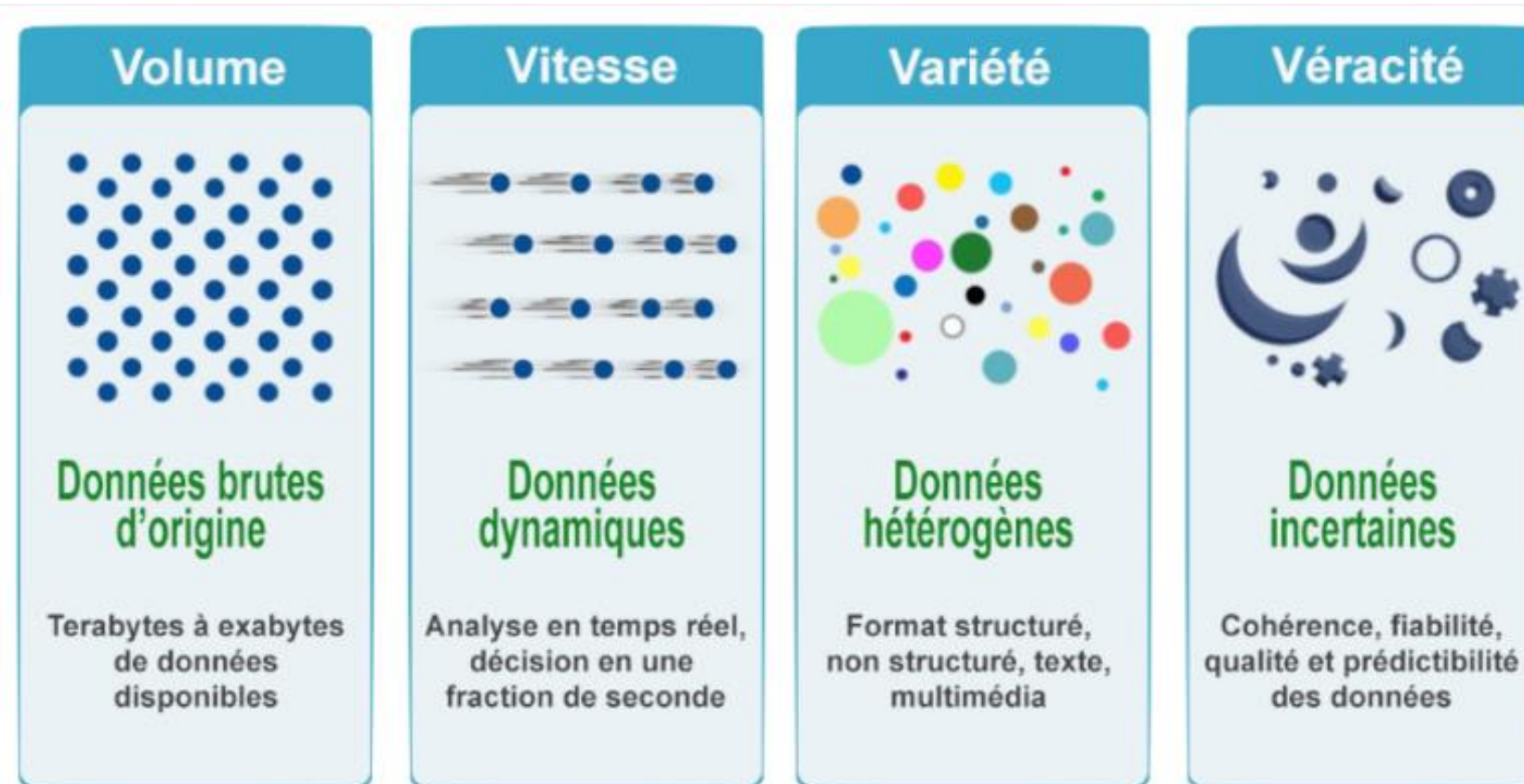
Big data : modélisation, stockage et traitement d'un ensemble de données massives et hétérogènes, dont l'exploitation permet la prise de décision et la découverte de nouvelles connaissances.

Facilité par :

- le faible coût du stockage
- le faible coût des processeurs
- la mise à disposition des données

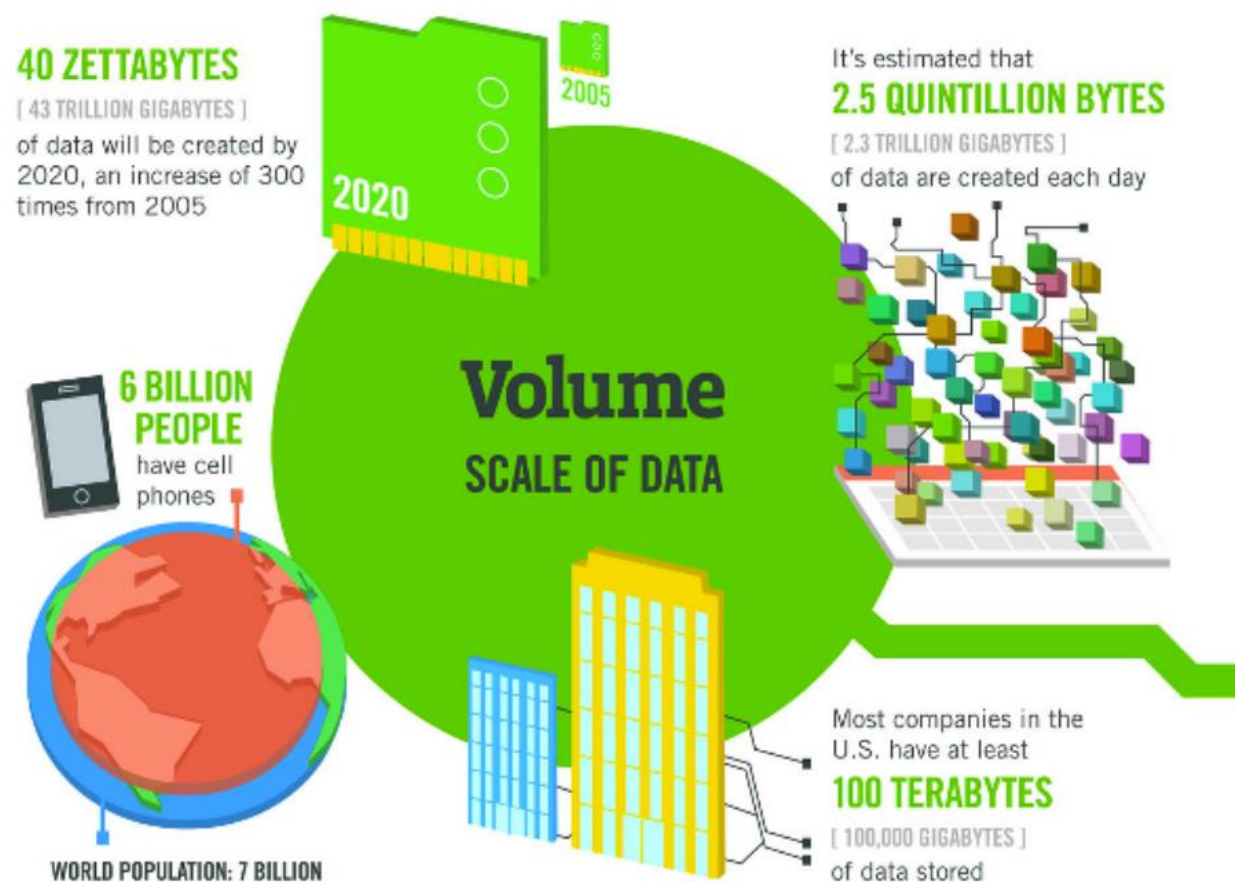
Caractéristiques

Les 4 caractéristiques des Big Data (les 4V)



Caractéristiques :

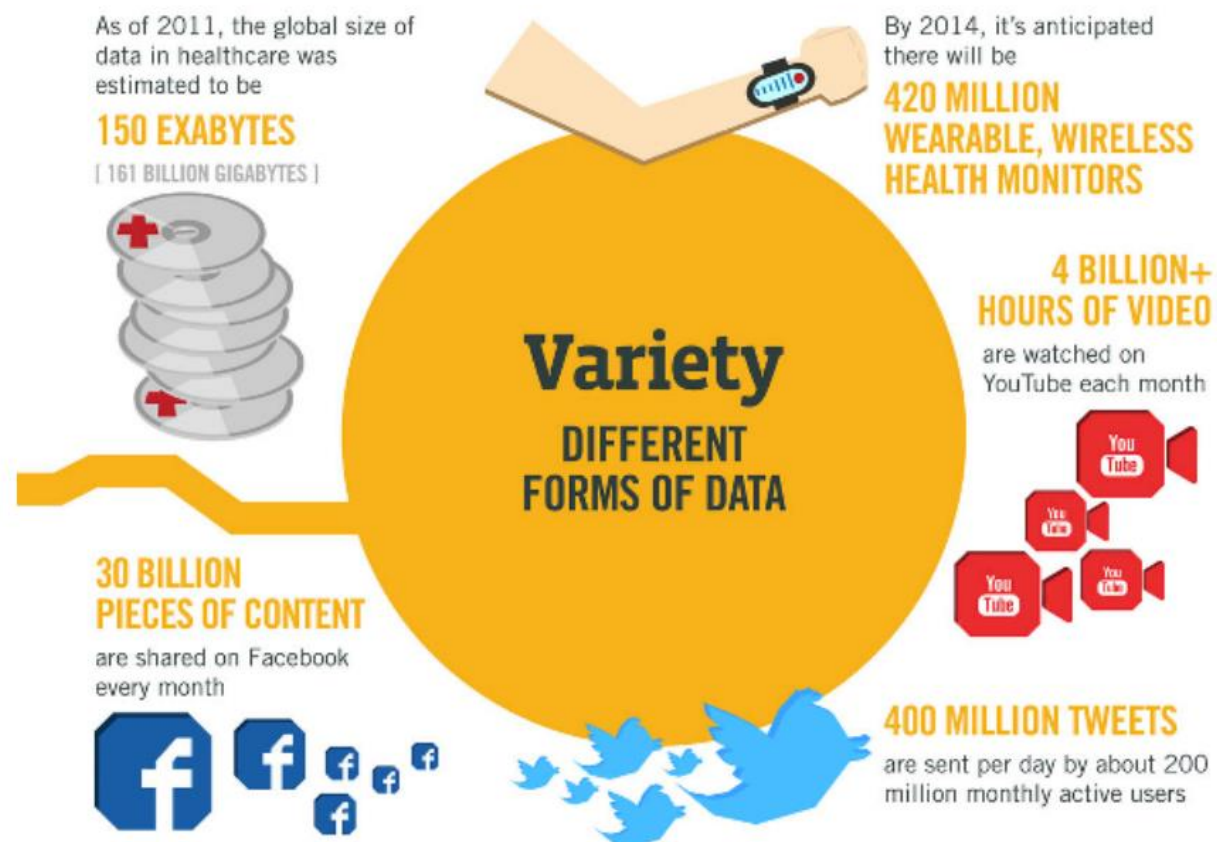
Les 4 V : Volume



Source : IBM

Caractéristiques :

Les 4 V: Variété



Source : IBM

Caractéristiques :

Les 4 V : Vélocité

The New York Stock Exchange captures

1 TB OF TRADE INFORMATION

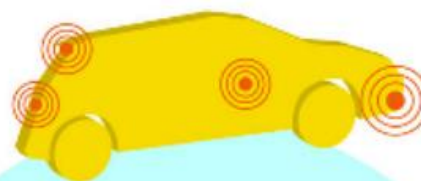
during each trading session



Modern cars have close to

100 SENSORS

that monitor items such as fuel level and tire pressure



Velocity
ANALYSIS OF
STREAMING DATA

By 2016, it is projected there will be

18.9 BILLION NETWORK CONNECTIONS

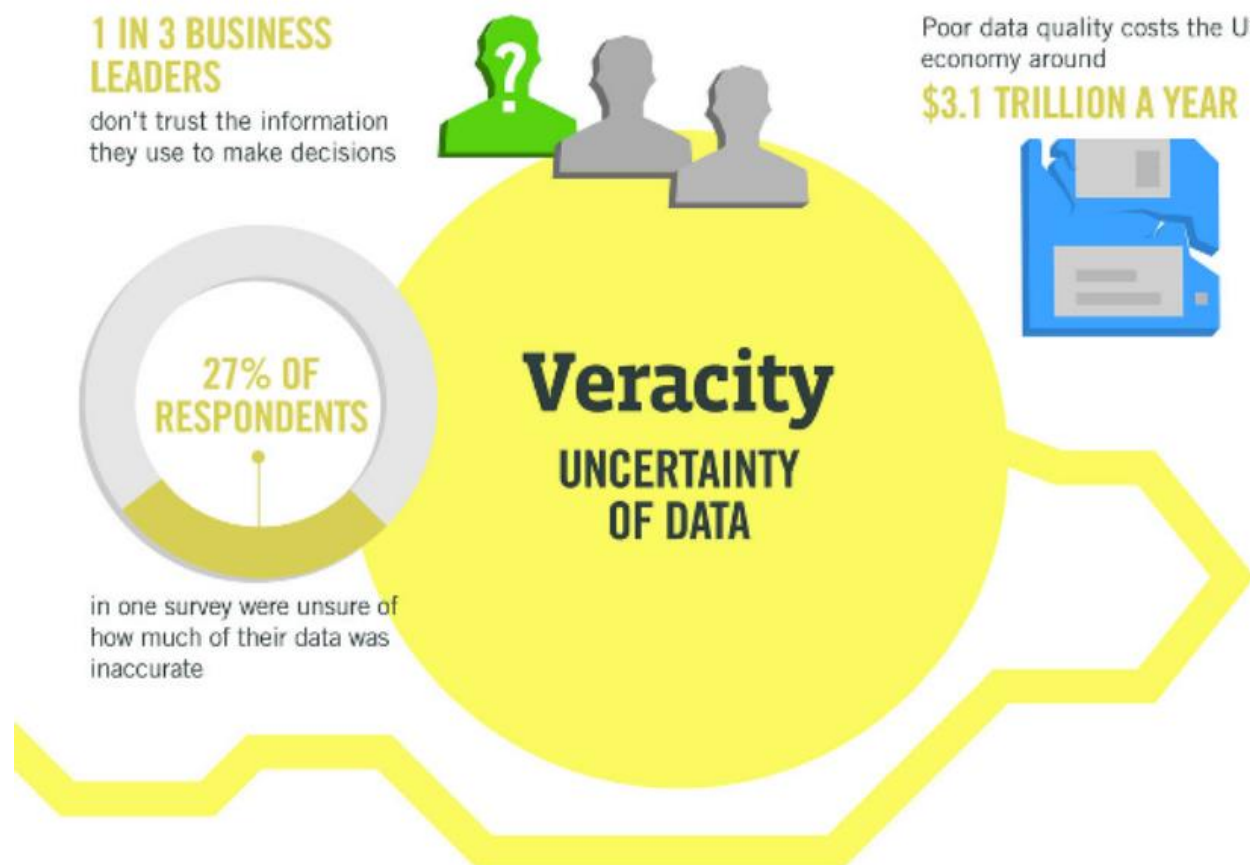
– almost 2.5 connections per person on earth



Source : IBM

Caractéristiques :

Les 4 V : Véracité



Source : IBM

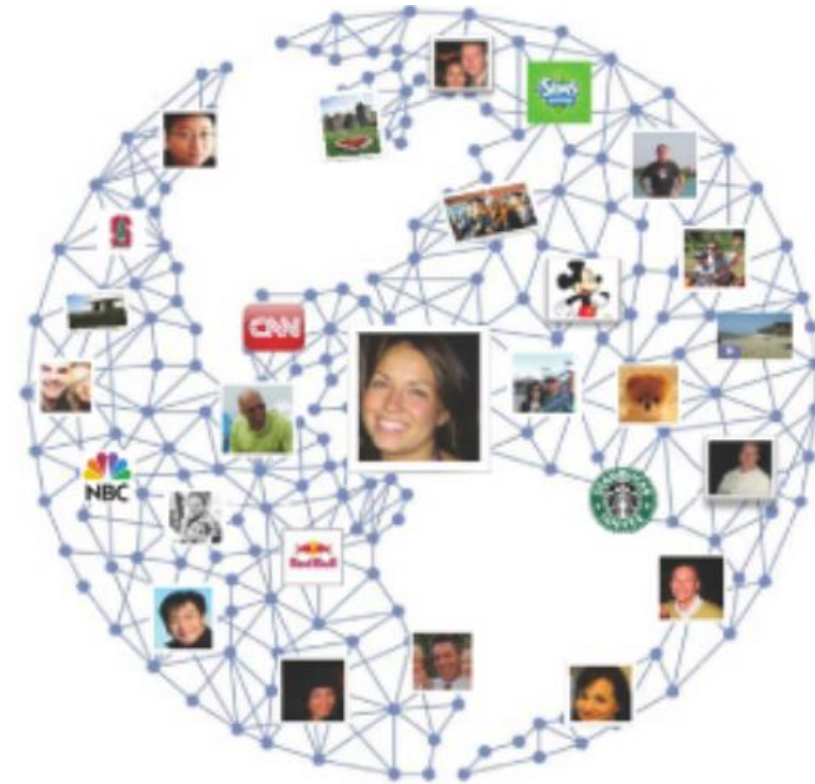
Géants du Web et Big data :



- Ce sont les géants du Web qui ont impulsé le mouvement **BigData** et **NoSQL**
- Souvent avec des solutions propriétaires qui ont vu ensuite leur équivalent OpenSource proposé

Chaque jour, Facebook gère :

- 2,5 milliards d'objets
- 500 To de nouvelles données
- 2,7 milliards de « Like »
- 300 millions de photos intégrées



Source : IBM

Géants du Web et Big data :

Facebook

Data Center de Facebook (2010)

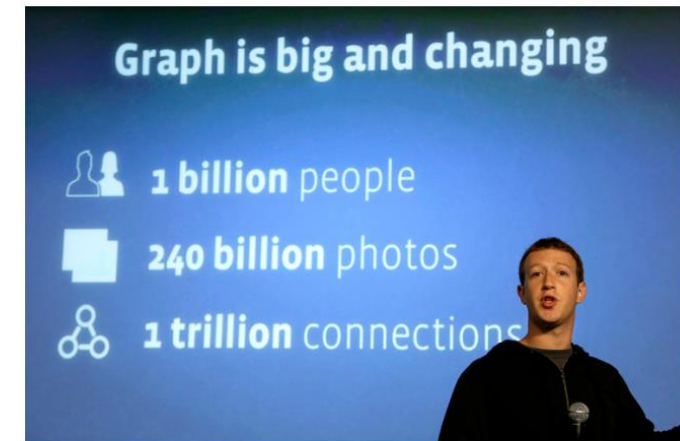
- 2500 CPU (serveurs)
- 1 PetaOctet d'espace disque (=1000 Tera = 1 million de Go)

- A développé l'outil Cassandra, devenu projet Apache en 2009, non utilisé par Facebook aujourd'hui, qui lui a préféré **Hbase** en 2010

- Données stockées sur architecture **Hadoop** dans une base de données

HBase

- **Hive** : outil d'analyse de données initialement développé par Facebook



Mark Zuckerberg (Credit: AP/Jeff Chiu), 2013

Géants du Web et Big data :



Google

Google a développé son propre système de fichiers distribué : GoogleFS (GFS)

- Environnement de stockage redondant sur un cluster composé de machines « jetables » de puissance moyenne
- Déploiement d'une solution basée sur les primitives Map et Reduce
- Système de gestion de données basé sur GFS : BigTable

Géants du Web et Big data :



- BigQuery est un service web RESTful intégré à Google Cloud qui permet l'analyse interactive de données massives dans l'espace de stockage Google
- « Append-only tables » : on ne peut pas modifier (UPDATE) ou supprimer (DELETE) des entrées dans une table, seulement y ajouter des entrées

Géants du Web et Big data :

Data Center de Google (2010)

- Un Data Center Google contient entre 100 et 200 racks, chacun contenant 40 serveurs
- environ 5000 serveurs par data-center, pour un total de 1 million de serveurs (estimation d'après la consommation électrique)



Source: Google

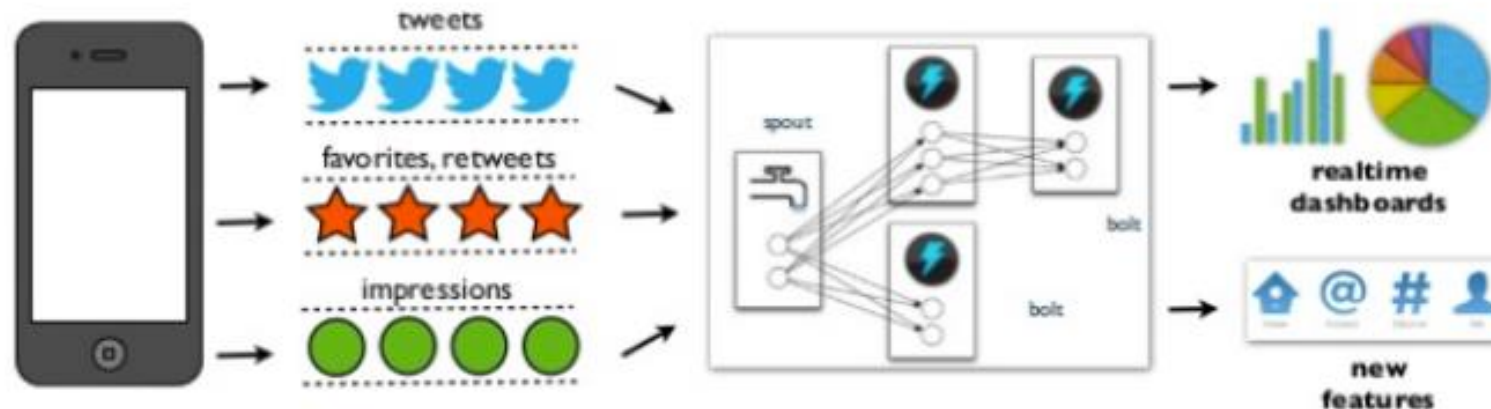
<https://www.google.fr/about/datacenters/>

Géants du Web et Big data :



Twitter

- **Outil Storm** : Traitement temps réel des données réparties sur un cluster de serveurs
- Idée : proposer une alternative à Hadoop qui réalise ses traitements distribués en mode batch
- Rendu OpenSource en 2011 (Apache)



Géants du Web et Big data :



Amazon

Dynamo (2007)

- Entrepôt de paires clés-valeurs entièrement distribué dans une architecture sans maître
- Dynamo a eu une grande influence dans le monde NoSQL
- Dynamo n'est pas distribué librement, mais de nombreux projets se sont basés sur ce principe pour proposer une solution libre (Cassandra, Riak, Voldemort (LinkedIn))

DynamoDB

- Offre cloud basée sur une technologie semblable

SOMMAIRE DU CHAPITRE

Big Data et motivations

Limites des SGBD relationnels

Modèles de distribution

Le Big Data analytique

Les bases de données NoSQL



Les SGBD relationnels

- Système de **jointure** entre les tables permettant de construire des requêtes complexes impliquant plusieurs entités
- Un système **d'intégrité référentielle** permettant de s'assurer que les liens entre les entités sont valides
- Gestion des transactions respectant les contraintes ACID :
 - Atomicité
 - Consistence
 - Isolation
 - Durabilité

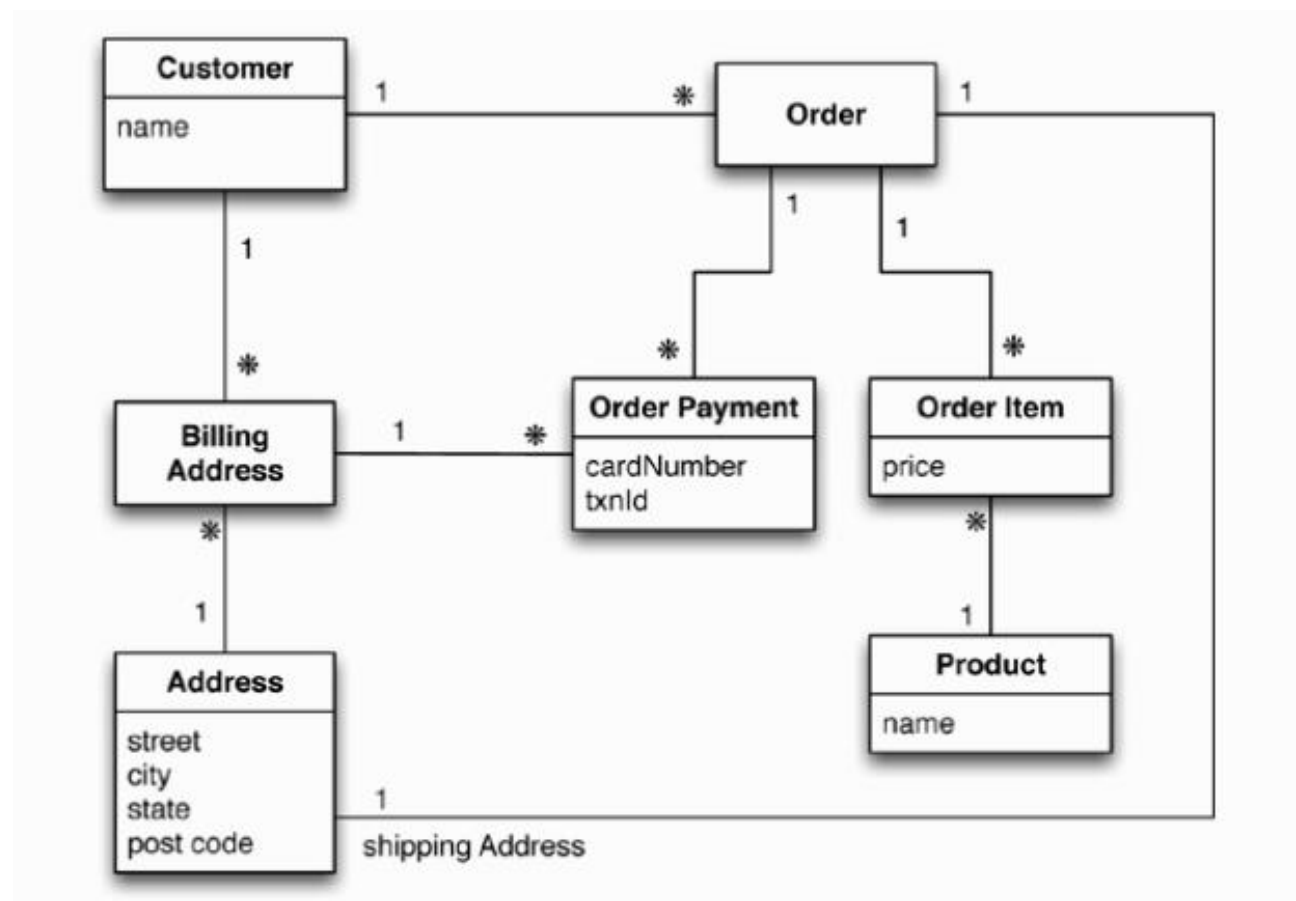
Les SGBD relationnels

Les contraintes ACID :

- **Atomicité** : une transaction se fait au complet... ou pas du tout
- **Cohérence** : les modifications apportées sur la base doivent être valides, en accord avec l'ensemble de la base et ses contraintes d'intégrité
- **Isolation** : les transactions lancées au même moment ne doivent jamais interférer
- **Durabilité** : toutes les transactions sont lancées de manière définitive

Les SGBD relationnels

Exemple d'application : Produits-Clients-Commande



Les SGBD relationnels

Exemple d'application : Produits-Clients-Commande

Customer	
Id	Name
1	Martin

Orders		
Id	CustomerId	ShippingAddressId
99	1	77

Product	
Id	Name
27	NoSQL Distilled

BillingAddress		
Id	CustomerId	AddressId
55	1	77

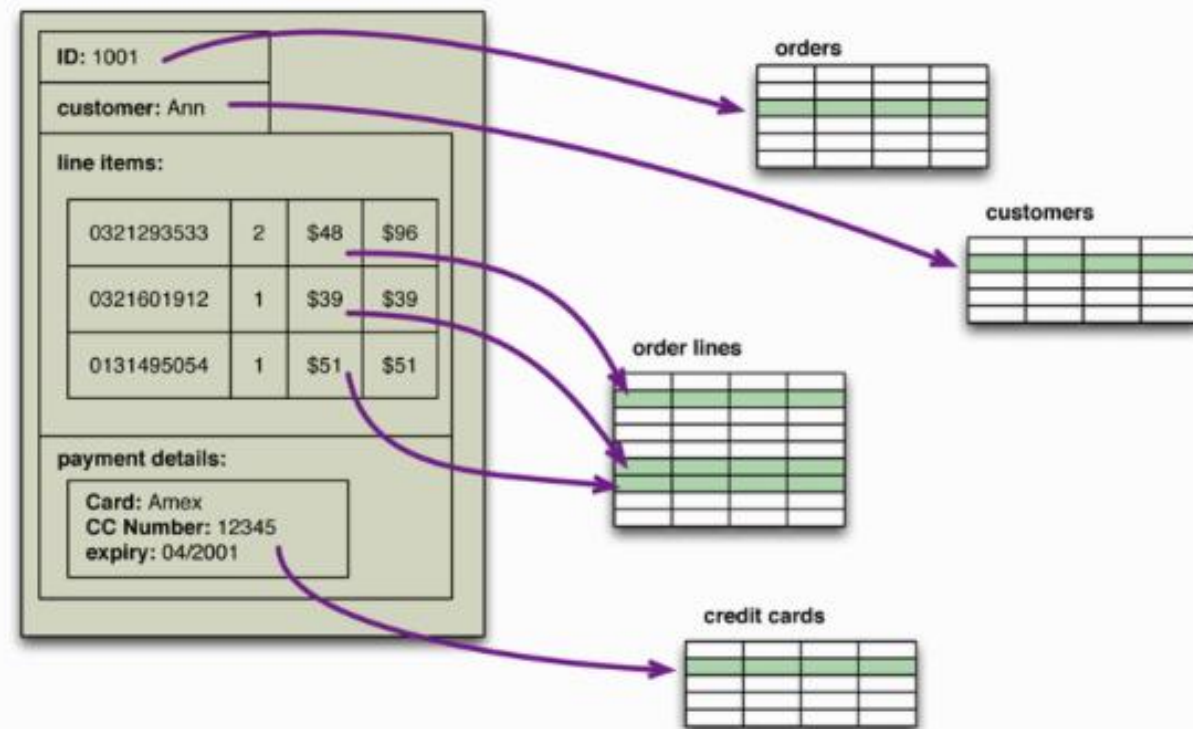
OrderItem			
Id	OrderId	ProductId	Price
100	99	27	32.45

Address	
Id	City
77	Chicago

OrderPayment				
Id	OrderId	CardNumber	BillingAddressId	txnId
33	99	1000-1000	55	abelif879rft

Limites des SGBD relationnels

- Les valeurs dans les N-uplets doivent être atomiques
- Ce n'est pas le cas pour les structures de données en mémoire qui peuvent être plus riches ...
- Il faut transformer les données



Limites des SGBD relationnels

SGBDR : schéma rigide

- Un moteur relationnel doit indiquer d'une façon ou d'une autre l'absence de valeurs dans une cellule
- marqueur NULL
- Coût en stockage

```
select *  
from Contact  
where titre='mme' or titre <> 'mme' or titre is null;
```

Limites des SGBD relationnels

- Les BD relationnelles sont des BD d'intégration
- un seul schéma pour toutes les applications, qui stockent leurs données dans une même base.
- le schéma est rendu plus complexe à cause de nombreuses applications qui se servent des données
- si un changement se fait dans le schéma, toutes les applications sont touchées...

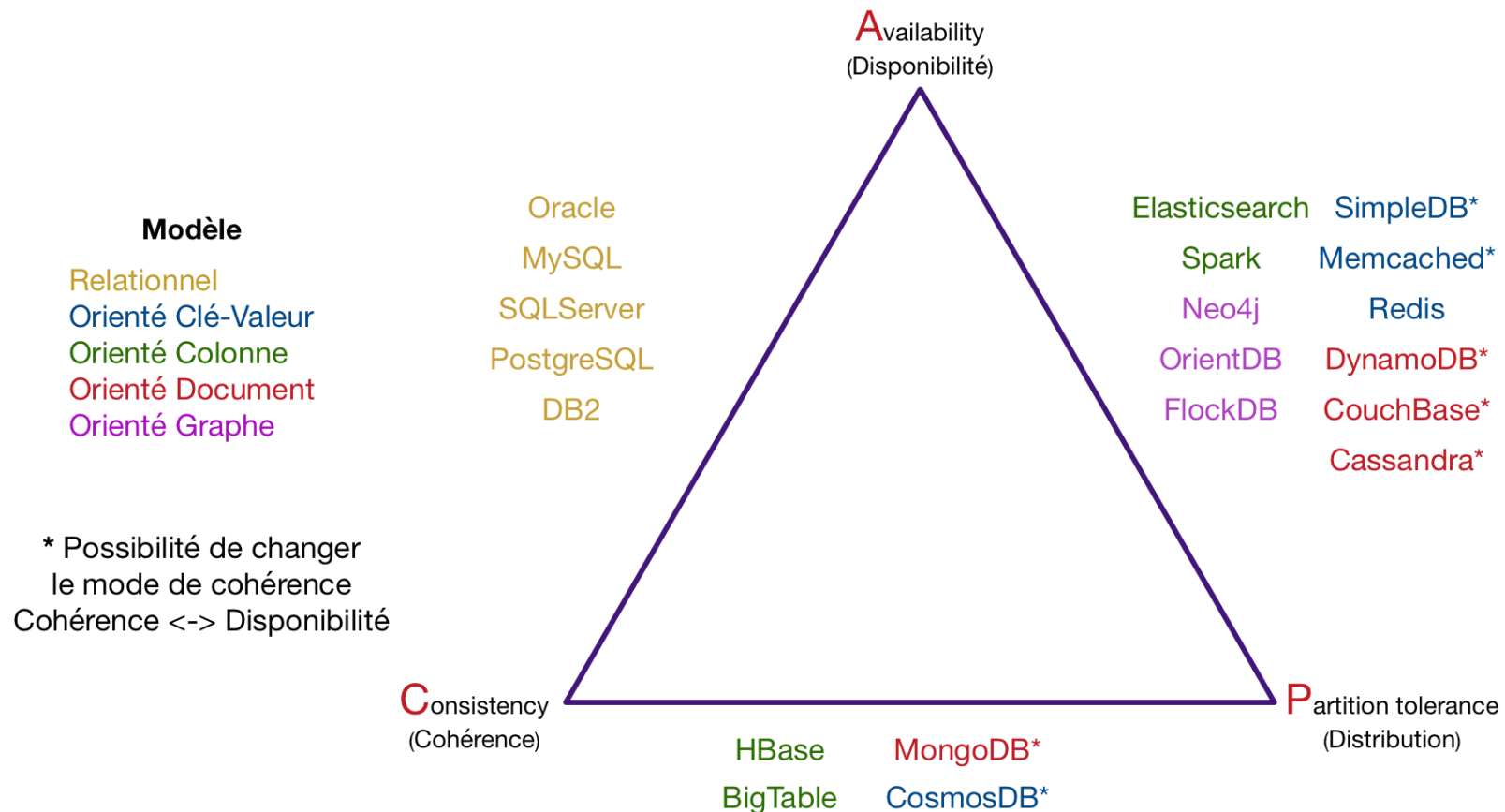
Limites des SGBD relationnels

Coût considérable pour les SGBD relationnels

- **Niveau données**
 - Avec la plupart des SGBD relationnels, les données d'une BD liées entre elles sont placées sur le même noeud du serveur
 - Si le nombre de liens est important, il est de plus en plus difficile de placer les données sur des noeuds identiques
- **Niveau traitements**
 - Difficile de maintenir avec des performances correctes les contraintes ACID à l'échelle du système distribué entier

Théorème de CAP (Brewer, 2000):

Dans un système distribué, il est impossible d'obtenir ces 3 propriétés en même temps, il faut en choisir 2 parmi 3



Théorème de CAP

Coherence (Consistance) : tous les nœuds du système voient exactement les mêmes données au même moment

Partition tolerance (Résistance au morcellement) : le système étant partitionné, aucune panne moins importante qu'une coupure totale du réseau ne doit l'empêcher de répondre correctement en cas de morcellement en sous-réseaux, chacun doit pouvoir fonctionner de manière autonome

Availability (Disponibilité) : la perte des nœuds n'empêche pas les survivants de continuer à fonctionner correctement, les données restent accessibles

Protocoles d'accès de données

Les moteurs NoSQL implémentent différentes méthodes d'accès aux données

- Des bibliothèques existent pour tous les grands langages clients afin de les rendre plus faciles d'utilisation

Protocole de type RPC (Remote Procedure Call) liant les clients et un serveur via des interfaces bien définies maintenant généralement une connexion TCP

SOMMAIRE DU CHAPITRE

Big Data et motivations

Limites des SGBD relationnels

Modèles de distribution

Le Big Data analytique

Les bases de données NoSQL



Modèles de distribution

Un modèle de distribution peut être basé sur du sharding, ou de la réplication, ou les 2 à la fois

Complexité croissante de distribution :

- serveur seul
- sharding peer-to-peer
- réplication maître-esclave
- réplication peer-to-peer
- combinaison du sharding et de la réplication

Modèles de distribution

SHARDING (Shard : éclat)

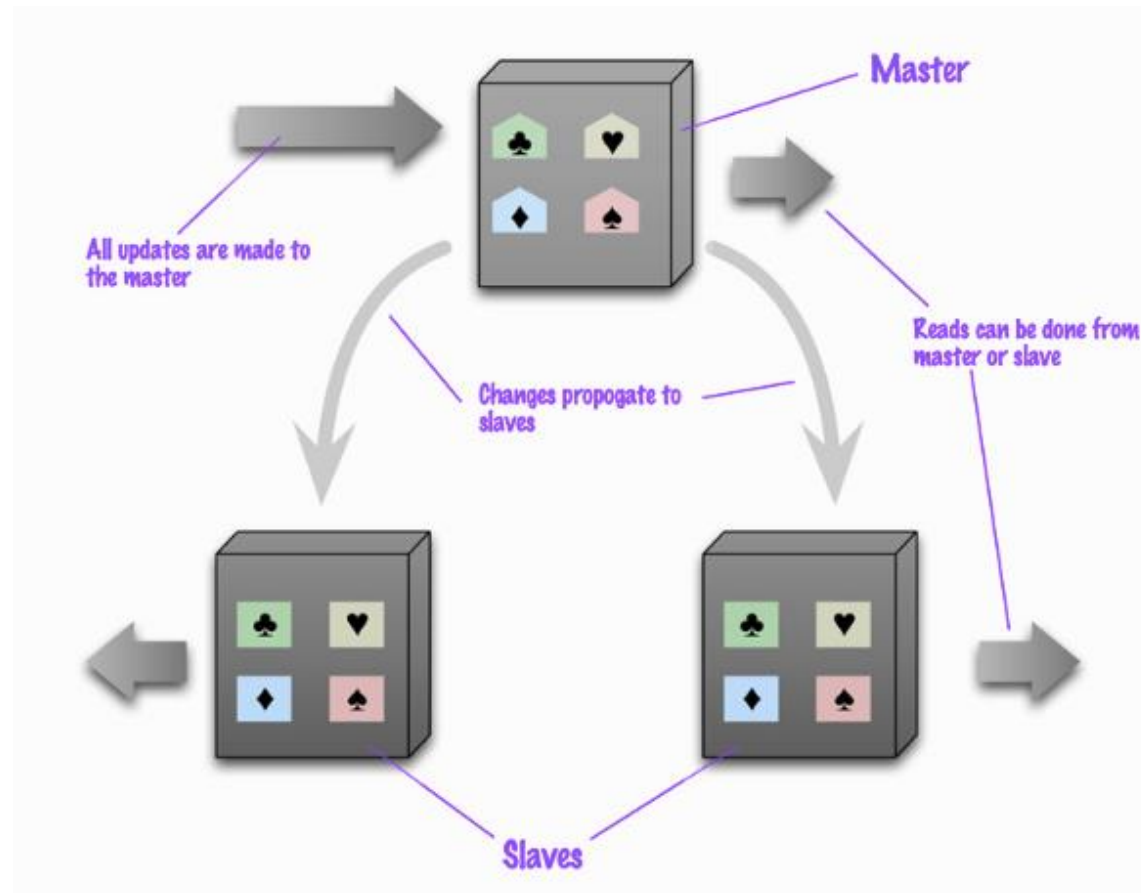
Sharding : éclatement des données en partitions distribuées sur plusieurs machines, de façon aussi automatique que possible par le moteur NoSQL

Complexité croissante de distribution :

- serveur seul
- sharding peer-to-peer
- réplication maître-esclave
- réplication peer-to-peer
- combinaison du sharding et de la réplication

Modèles de distribution

Réplication maître-esclave



Modèles de distribution

Réplication maître-esclave

La machine maître conserve la configuration du système, reçoit les demandes de mise à jour et les propage vers les machines esclaves

- les machines esclaves sont accessibles en lecture seule
- + Intéressant en cas de système demandant beaucoup d'accès en lecture sur les données
- + Résilience en lecture : si le master tombe, les esclaves peuvent toujours fournir les données
- Présence d'un SPOF (Single Point of Failure) en cas de mise à jour
- Incohérences possibles si accès en lecture sur différents esclaves pour lesquels le niveau de mise à jour n'est pas le même

Modèles de distribution

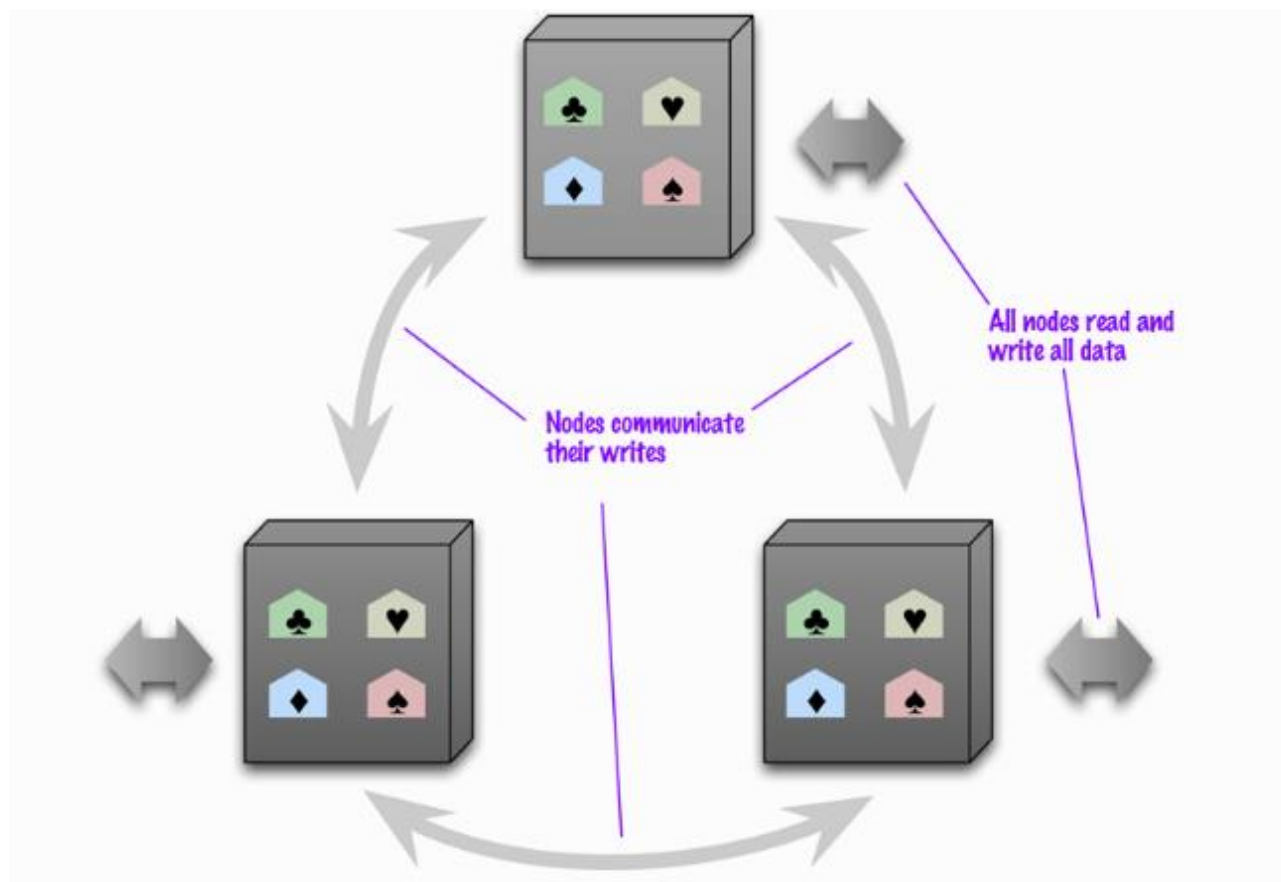
Réplication peer-to-peer

Tous les nœuds ont le même poids, acceptent les écritures, et la perte de l'un d'entre eux ne remet pas en cause l'accès aux données

- Il faut mettre en place un protocole de bavardage (gossip protocol) entre les nœuds
- Inconvénient : Consistance
- 2 personnes peuvent mettre à jour le même N-uplet en même temps sur 2 nœuds différents...

Modèles de distribution

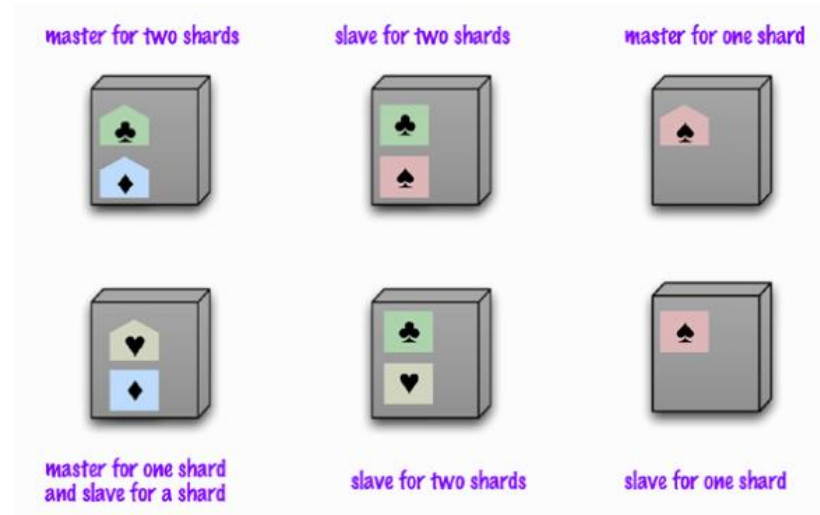
Réplication peer-to-peer



Modèles de distribution

Combinaison du sharding et de la réplication

- Réplication maître esclave et sharding
- Plusieurs maîtres, mais chaque objet-donnée n'est que sur un seul maître
- Un nœud peut être maître pour certaines données et esclave pour d'autres
- On peut aussi choisir de faire des maîtres et esclaves « exclusifs »



Modèles de distribution

Combinaison du sharding et de la réplication

Deux façons de distribuer les données

- le sharding distribue les données sur des nœuds différents, et chaque nœud est la seule source pour un ensemble d'objets-données
- la réplication copie les données sur plusieurs nœuds, chaque objet-donnée pouvant se trouver à plusieurs endroits

⇒ un système peut utiliser l'une des techniques ou les 2

- la **réplication** se fait de 2 façons :
 - réplication **maître-esclave**
 - réplication **peer-to-peer**

Modèles de distribution

Combinaison du sharding et de la réplication

- la **réplication** se fait de 2 façons
 - réplication **maître-esclave** : le maître gère les écritures et la synchronisation avec les esclaves, et les esclaves ne sont accessibles qu'en lecture
 - réplication **peer-to-peer** : l'écriture et la lecture se font sur tous les nœuds, qui se synchronisent entre eux
- => la réplication maître-esclave réduit les conflits de mise à jour mais la réplication peer-to-peer évite d'avoir un SPOF en écriture

SOMMAIRE DU CHAPITRE

Big Data et motivations

Limites des SGBD relationnels

Modèles de distribution

Le Big Data analytique

Les bases de données NoSQL



Le Big Data analytique

On peut classer les orientations Big Data en 2 catégories :

- **BD opérationnelles**

- correspond à l'OLTP en relationnel
- BD constamment modifiée pour soutenir une activité en perpétuel changement
- Montée en charge sur des PetaOctets
- Exemples : Cassandra, Hbase

- **BD analytiques**

- correspond à l'OLAP en relationnel
- Traitement massif des données et agrégations
- Approche principale : Map-Reduce

Map Reduce : Présentation

Modèle de programmation parallèle pour le traitement de grands ensembles de données

- Développé par Google pour le traitement de gros volumes de données en environnement distribué
 - permet de répartir la charge sur un grand nombre de serveurs (cluster)
 - abstraction quasi-totale de l'infrastructure matérielle
 - gère entièrement, de façon transparente le cluster, la distribution de données, la répartition de la charge, et la tolérance aux pannes
 - ajouter des machines augmente la performance
- La librairie Map-Reduce existe dans plusieurs langages (C++, C#, Java, Python, Ruby...)

Map Reduce : Usage

Utilisé par les grands acteurs du Web notamment pour :

- Construction des index (Google Search)
- Détection de spams (Yahoo)
- Data Mining (FaceBook)
- Classification de données
- Analyse d'images

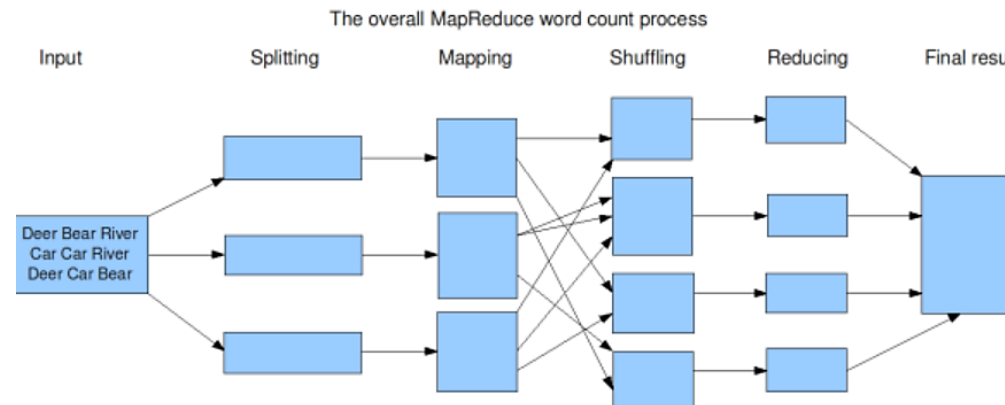
Map Reduce : Gestion de données

Principales opérations à faire sur de grands ensembles de données :

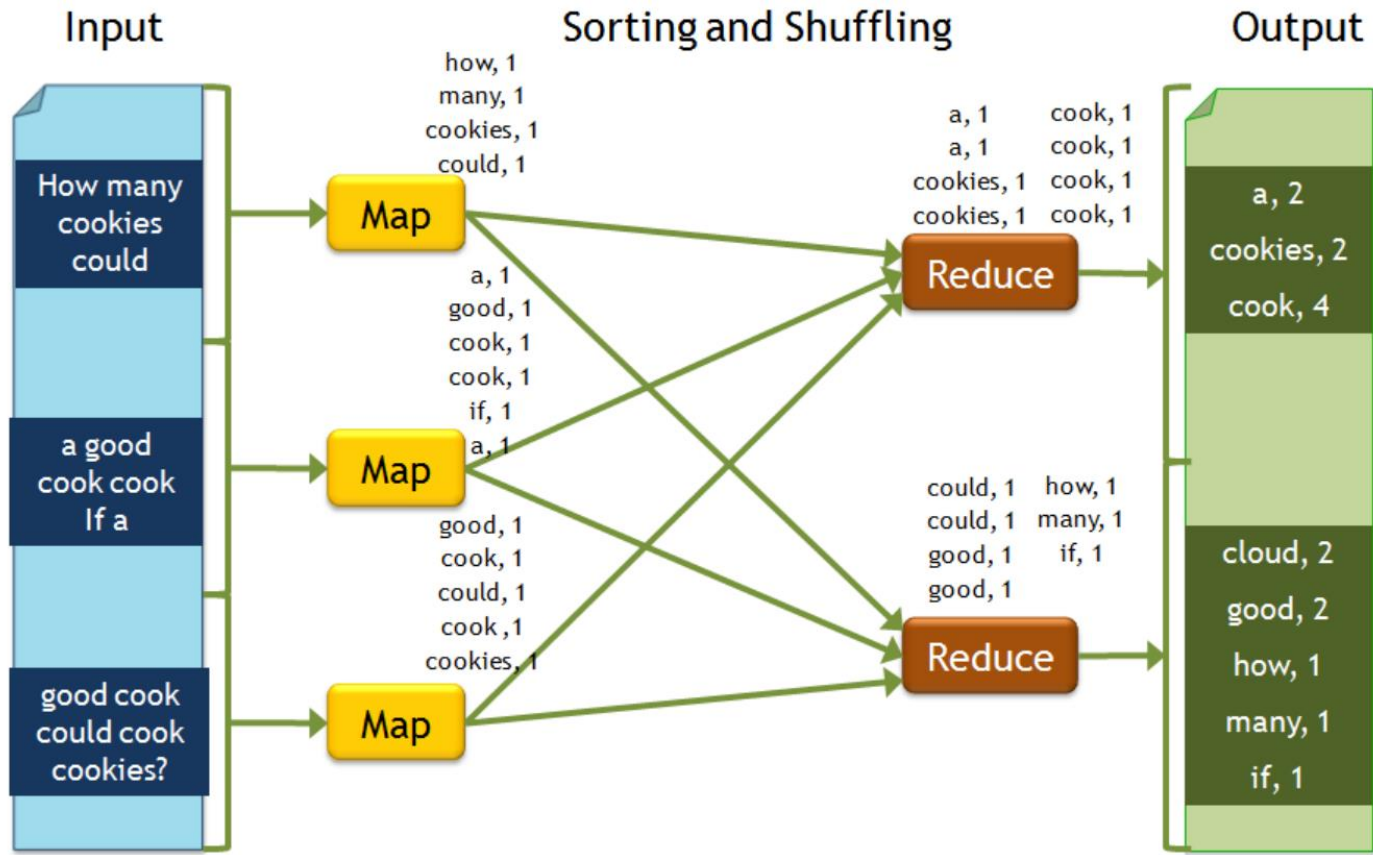
1. itérer sur un grand nombre d'enregistrements
2. extraire quelque chose ayant un intérêt, de chacun de ses enregistrements
 1. regrouper et trier les résultats intermédiaires
 2. agréger tous ces résultats ensemble
 3. générer le résultat final

Map Reduce : Gestion de données

- Dans le modèle de programmation Map-Reduce, le développeur implémente 2 fonctions : la fonction Map et la fonction Reduce
- **opérations 1 et 2 :**
la fonction **Map** traite une paire clé/valeur et génère un ensemble de paires de clés intermédiaires/valeurs
- **opérations 3, 4 et 5:**
la fonction **Reduce** fusionne toutes les valeurs intermédiaires associées avec la même clé intermédiaire



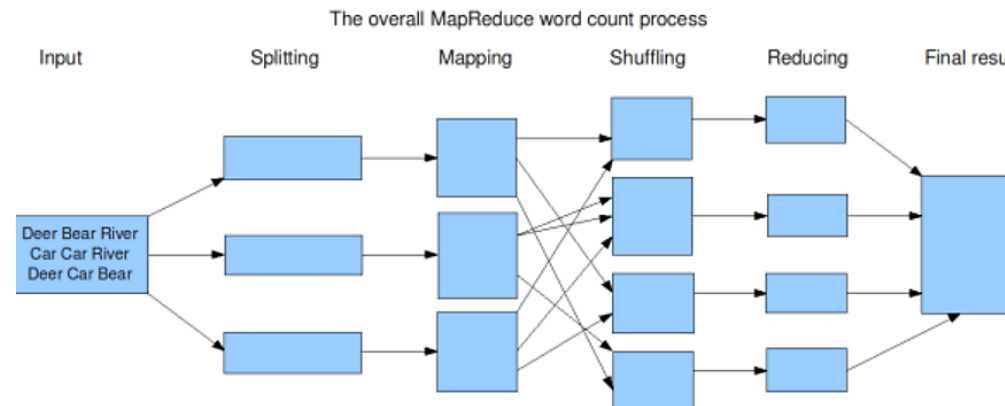
Map Reduce : Gestion de données



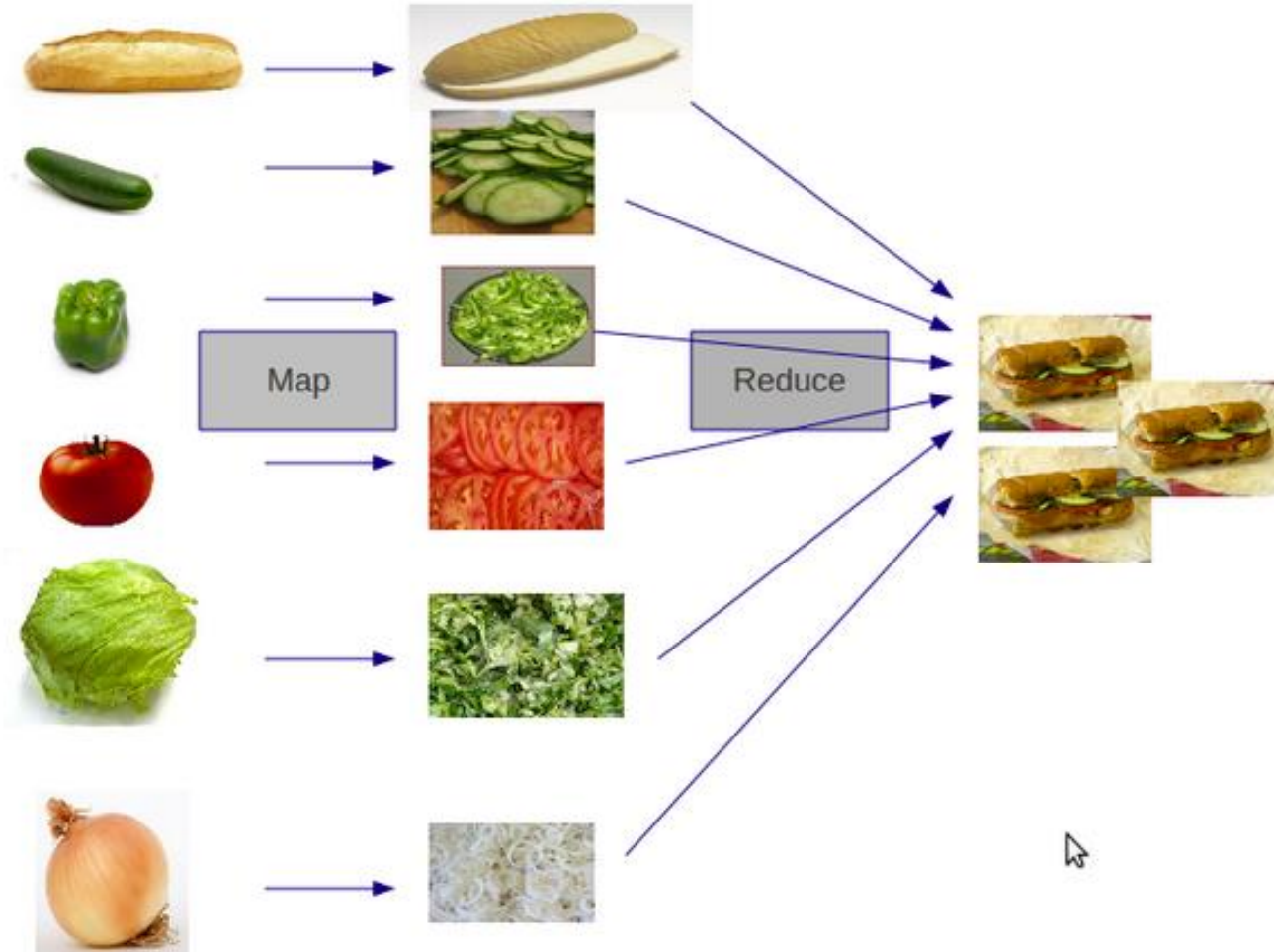
By Manaranjan Pradhan

Map Reduce : Gestion de données

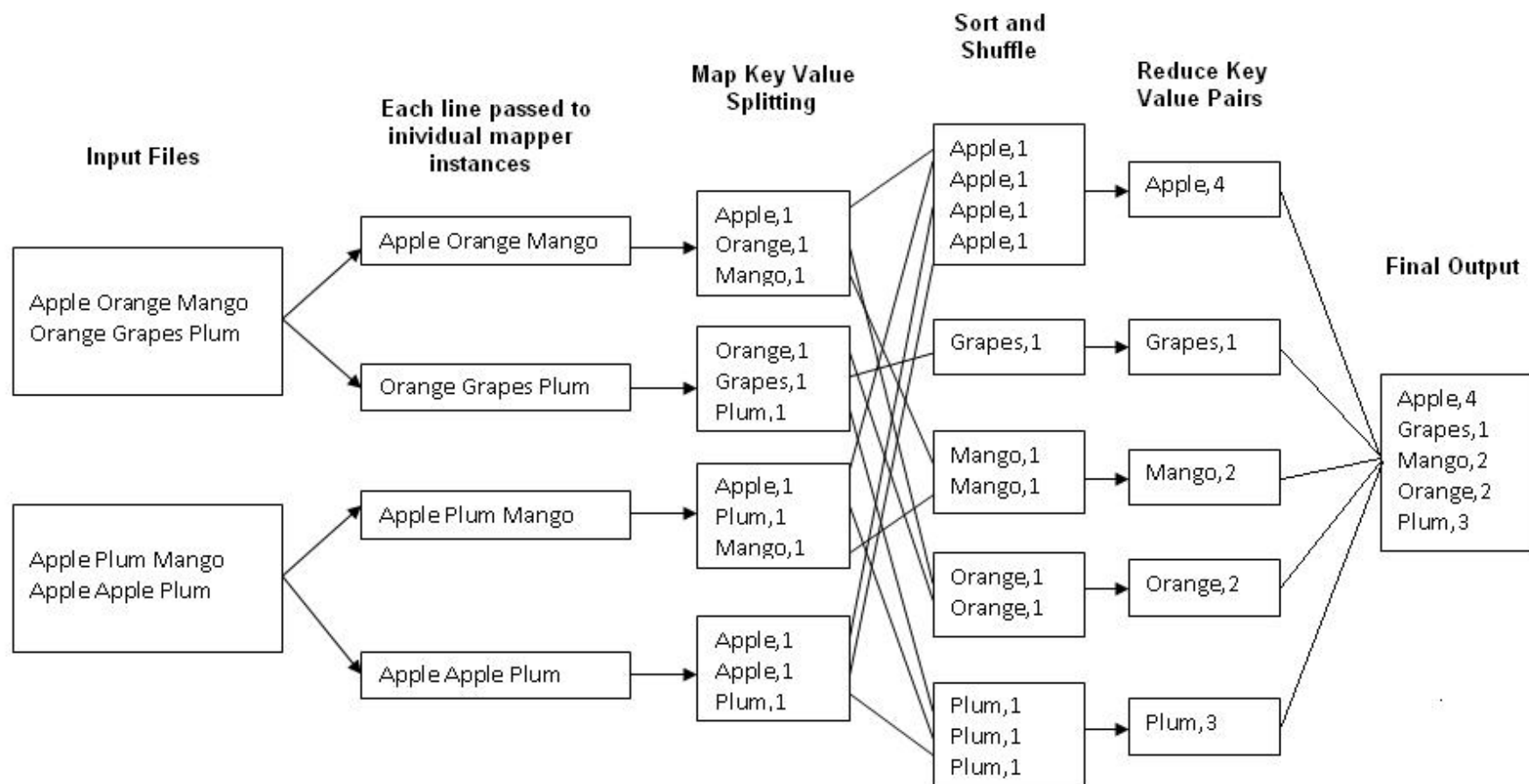
- Dans le modèle de programmation Map-Reduce, le développeur implémente 2 fonctions : la fonction Map et la fonction Reduce
- **opérations 1 et 2 :**
la fonction **Map** traite une paire clé/valeur et génère un ensemble de paires de clés intermédiaires/valeurs
- **opérations 3, 4 et 5:**
la fonction **Reduce** fusionne toutes les valeurs intermédiaires associées avec la même clé intermédiaire



Map Reduce : exemple



Map Reduce : WordCount



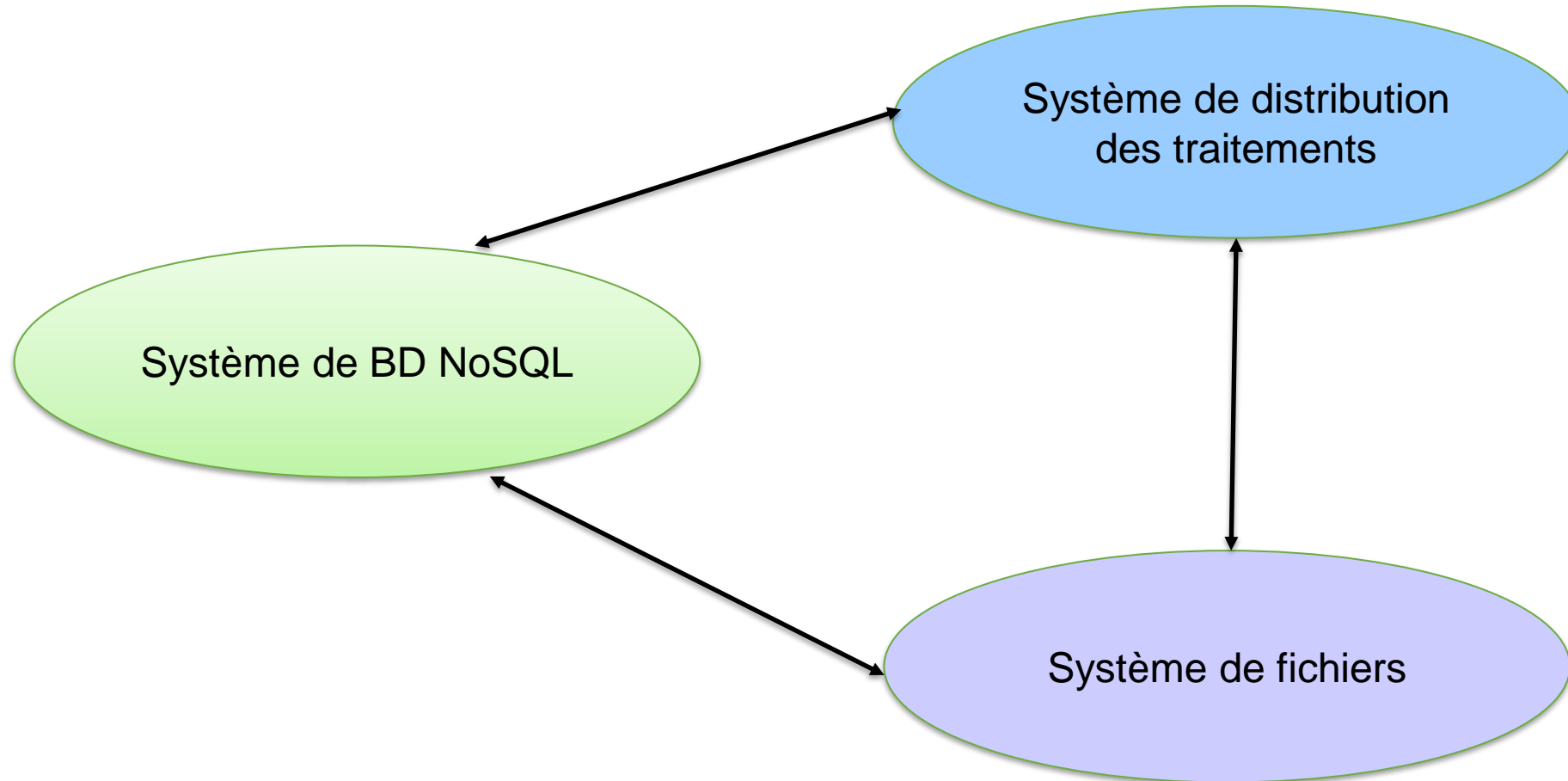
Exemple de framework Big Data : Hadoop

- Dans le modèle de programmation Map-Reduce, le développeur implémente 2 fonctions : la fonction Map et la fonction Reduce
- **opérations 1 et 2 :**
la fonction **Map** traite une paire clé/valeur et génère un ensemble de paires de clés intermédiaires/valeurs
- **opérations 3, 4 et 5:**
la fonction **Reduce** fusionne toutes les valeurs intermédiaires associées avec la même clé intermédiaire

Exemple de framework Big Data : Hadoop

- Hadoop = High-Availability Distributed Object-Oriented Platform
- Framework libre et open source
- Hadoop a été créé en 2002 pour les besoins du projet « Apache Nutch » intégrant MapReduce à la suite de la sortie de l'article de Google en 2004
- Projet indépendant de la fondation Apache depuis 2008
- Utilisé par les géants du Web comme Yahoo, Twitter, LinkedIn, eBay, Amazon...

Exemple de framework Big Data : Hadoop



SOMMAIRE DU CHAPITRE

Big Data et motivations

Limites des SGBD relationnels

Modèles de distribution

Le Big Data analytique

Les bases de données NoSQL

Le mouvement NoSQL

Historique

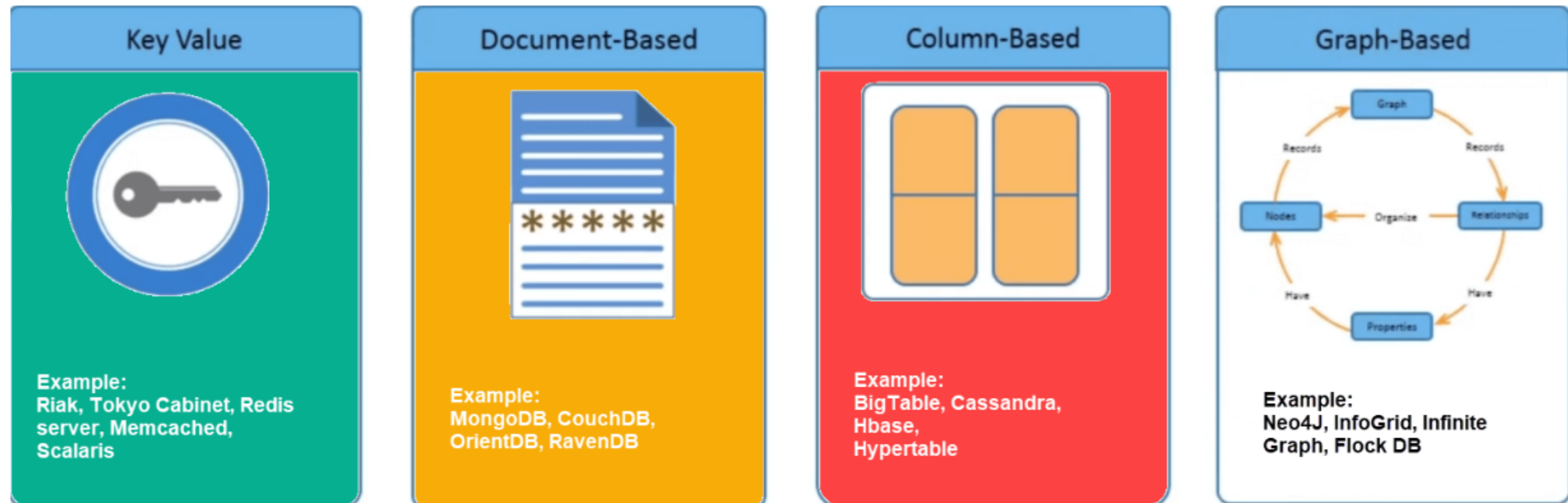
- Le concept du NoSQL à une bonne décennie d'ancienneté
- Utilisé pour la première fois en 1998
- Développé par des start-up comme logiciel de stockage des données
- Inauguration du NoSQL le 11 novembre 2009 à San Francisco
- Début de normalisation d'un langage de manipulation des données UnQL (Unstructured Query Language) en 2011

Le mouvement NoSQL

- NoSQL = **Not Only SQL**
- SGBD **non relationnel** issue du monde Web
- Les données ne sont plus manipulées avec le SQL
- L'unité de stockage n'est plus la table
- La définition d'un schéma de données relationnel n'est plus nécessaire
- **Renonciation aux fonctionnalités des SGBDR (ACID)**
- Les principaux axes sont la haute disponibilité et le partitionnement des données
- Permet de gérer de très grosses volumétries de données

Classification des modèles NoSQL

En fonction du schéma des données



Le mouvement NoSQL

1) Les bases clefs/valeurs

Principes du modèle

- Associe un identifiant unique à chaque valeur dans la base de données
- Une valeur peut être de type simple ou un objet sérialisé
- Fournit quatre opérations basiques-création, lecture, modification, suppression
- Toute l'intelligence dans la récupération des données se situe dans l'applicatif client

Le mouvement NoSQL

2) Les bases orientées colonnes

Principes du modèle

- Stocke les informations sous forme de colonnes
- Permet d'ajouter très facilement des informations
- Limite les valeurs nulles



Le mouvement NoSQL

3) Les bases orientées documents

Principes du modèle

- Destinée à stocker des documents
- Le contenu d'un document ne se limite pas à des attributs simples
- Chaque document a sa propre structure dans une même collection

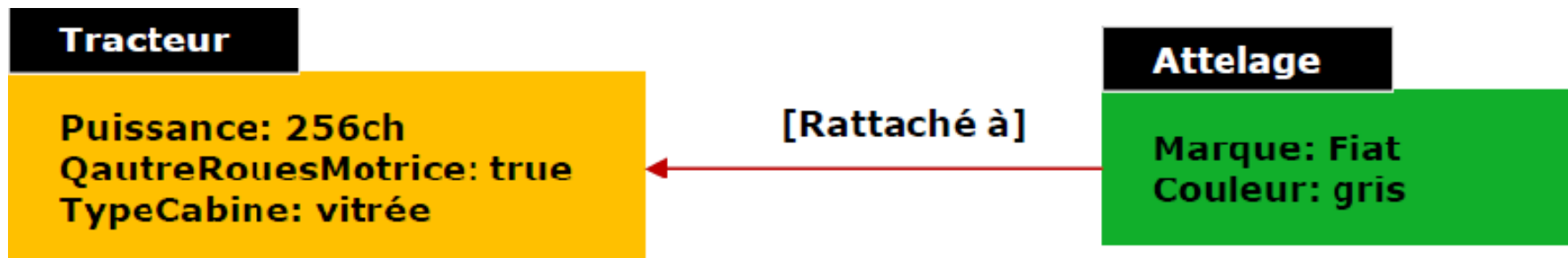
Le mouvement NoSQL

4) Les bases orientées graphes

Le moins connus et assez complexe à mettre en œuvre

Principes du modèle

- Repose sur un modèle mathématique appelé Théorie des Graphes
- Données modélisées sous format de nœud
- Données reliées entre elles par des Arc nommés



Le mouvement NoSQL

Autres types de bases NoSQL

- Base de données hiérarchique
- Base de données objet

Références

- *Introduction aux systèmes NoSQL*, Bernard Espionnasse, mai 2016
- *La révolution Big Data: les données au coeur de la transformation de l'entreprise*. J.C. Cointot. Junod, 2014
- *Bases de données noSQL et Big Data*. P. Lacomme, S. Aridhi, R. Phan, Ellipse, 2014.
- *NOSQL Distilled: a brief guide to the emerging world of polyglot persistence*. Pramod J. Sadalage, Martin Fowler. Addison-Wesley, 2013
- *Les bases de données et le Big Data: Comprendre et mettre en oeuvre*. Rudi Bruchez, Eyrolles, 2015

Des questions ?

Contactez-nous en scannant le QR code
et nous répondons à vos questions.



Numéro de téléphone

[01 89 16 90 08](tel:0189169008)

Adresse e-mail

contact@caplogy.com

Site Web

www.caplogy.com