

# NoSQL - Les données dans les nuages

©EISTI



8 janvier 2018

- 1 Sommaire
- 2 Introduction
- 3 Définitions
- 4 Paradigmes de stockage
  - Bdd orientée colonne
  - Bdd orientée clé-valeur
  - Bdd orientée document
  - Bdd orientée graphe

## SQL

- Système de gestion de bases de données relationnelles (SGBDR) : système le plus utilisé
- Langage de requête SQL
  - Langage riche, outillé
  - Intuitif
  - Majoritairement intégré
  - Standardisé (ou presque !)

## SQL

- Un SGBDR vérifie les propriétés ACID :
  - **A**tomicité : un ensemble de traitements d'une transaction est effectué entièrement ou globalement annulé
  - **C**onsistance : garantit l'état d'une BDD avant et après une transaction
  - **I**solation : gère les transactions concurrentes afin qu'elles n'interfèrent pas entre elles
  - **D**urabilité : les données sont sauvegardées dans la BDD de manière permanente à la validation d'une transaction
- Les données respectent les formes normales

## SGBDR

- Propriétés :
  - centralisation des données et des traitements
  - efficacité par normalisation des données et des traitements
  - optimisations coûteuses : réplication, parallélisation
- Un SGBDR répond à la majorité des problématiques industrielles

## Limitations

Problème de gestion de larges quantité de données ("One size fits all"). Solutions envisagées puis assimilées :

- Bases de données objet
- Stockage XML
- Outsourcing

## Problèmes

- Solutions peu choisies
- Coûts de gestion et matériels élevés

## Web 2.0

- Nouveaux marchés :
  - applications web
  - indexations
  - statistiques
- Nouvelles problématiques :
  - gros volumes de données
  - large demande d'accès aux données
  - temps de réponse immédiat ou, au moins, constant

## NoSQL : Pourquoi ?

- Stocker le contenu du web prend de la place...
- Offrir un service efficace de vente en ligne 365j/an est éprouvant
- Stocker, traiter et retourner des millions de nouvelles, d'opinions d'amis, d'amis d'amis, {d'amis}\* est un sacré challenge

## NoSQL

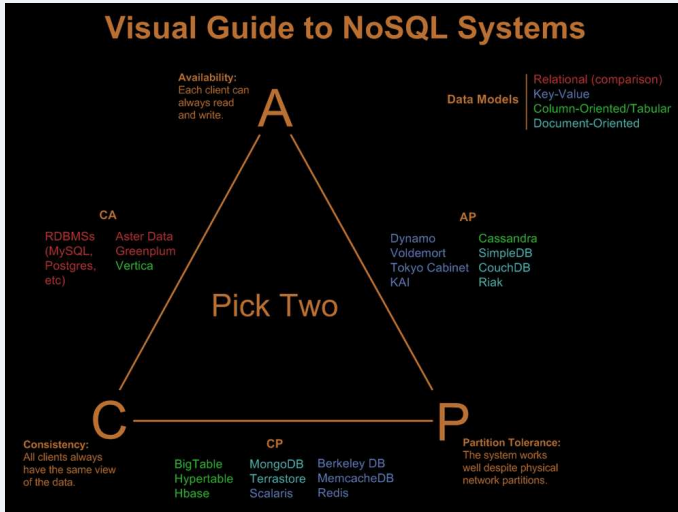
- Début 2000 grands acteurs : Google, Amazon
- Stockage serveur et temps machine disponibles
- NoSQL (Not only SQL)
- Technologie web, liée au cloud
- Objectifs :
  - adaptabilité de taille
  - répartition des données
  - réplication des données
  - souplesse de structure
  - montée en charge
- Comment ? : en relachant une des propriétés ACID

## Théorème CAP

Il est impossible, pour un système distribué de respecter les 3 contraintes suivantes :

- **C**onsistency : tous les noeuds voient les mêmes données au même moment
- **A**vailability : Les données sont disponibles pour l'ensemble des utilisateurs, en temps constant
- **P**artition tolerance : les pannes réseaux ne doivent pas altérer la réponse

# Théorème CAP (source : Visual Guide to NoSQL Systems, [blog.nahurst.com](http://blog.nahurst.com))



## Propriétés BASE

La partition des données rend difficile l'application des propriétés ACID. L'approche BASE est alors préférée.

- **B**asically **A**vailable : l'application fonctionne la plupart du temps
- **S**oft state : un état stable de la donnée est toujours renvoyé mais pas de garantie sur la dernière version
- **E**ventual consistency : les modifications par d'autres utilisateurs sont vues au cours du temps

## Scalabilité (montée en charge)

- Capacité à gérer plus de données, de calculs sans altérer les performances d'un système
- Moyens :
  - mutualisation de processeurs
  - ajout de mémoire
  - duplication de services, répartition de charge

## 4 grands paradigmes

- Orienté colonnes
- Orienté clé-valeur
- Orienté document
- Orienté graphe

# Bdd relationnelle

## Bdd relationnelle

Une relation contient un ensemble d'attributs. Une donnée est un élément de cet ensemble, elle est définie par l'ensemble des attributs constituant la relation.

## Exemple pour la suite

id	nom	prenom	age
1	Devin	Florent	22
2	LeNir	Yannick	18
3	Loubiere	Peio	

# Stockage orienté colonnes

## Présentation

- Données organisées en familles de colonnes
- Chaque famille contient un attribut de toutes les données
- L'union des familles de colonnes reconstruit la donnée
- Un identifiant commun permet d'unification de la donnée
- Le traitement d'une partie de la donnée importe plus que la récupération de la donnée complète

## utilisé par...

Google (BigTable), Facebook (Cassandra), Amazon (SimpleDB)

# Stockage orienté colonne

## Objectifs

- Une structure différente pour chaque donnée
- Un stockage réparti
- Extrêmement rapide pour des opérations sur un attribut (stats)
- Jointures pour reconstruire la donnée coûteuses en temps

# Stockage orienté colonne

## Exemple

Famille Nom : ("Devin", "LeNir", "Loubière")

Famille Prenom : ("Florent", "Yannick", "Peio")

Famille Age : (22, 18)

# Stockage orienté clé-valeur

## Présentation

- Stockage classique type map ;
- Pas de contrainte sur la clé ou la valeur
- À une clé est associée une donnée complète

## utilisé par...

Amazon dynamo, Projet Voldemort (Linkedin)

# Stockage clé-valeur

## Objectifs

- Une structure différente pour chaque donnée
- Un stockage réparti
- Rapide pour récupérer une donnée
- Jointures pour faire des statistiques en colonnes

# Stockage clé-valeur

## Exemple

Donnée1 : ("prof1" : ("nom" : "Devin", "prenom" : "Florent", "age" : 22))

Donnée2 : ("prof2" : ("nom" : "LeNir", "prenom" : "Yannick", "age" : 18))

Donnée3 : ("prof3" : ("nom" : "Loubiere", "prenom" : "Peio"))

# Stockage orienté document

## Présentation

- Extension du modèle clé-valeur
- Stocke sous forme de document structuré
- Implémentation en différents langages (XML, JSON)

## utilisé par...

MongoDB, CouchDB (Ubuntu one)

# Stockage orienté document

## Objectifs

- une structure différente pour chaque donnée ;
- un stockage réparti ;
- rapide pour récupérer une donnée ou une colonne ;
- adapté aux applications web.

# Stockage orienté document

## Exemple

```
Donnée1 : ("prof1" :  
<prof><nom>Devin</nom><prenom>Florent</prenom><age>22</age></prof>))  
Donnée2 : ("prof2" :  
<prof><nom>LeNir</nom><prenom>Yannick</prenom><age>8</age></prof>))  
Donnée3 : ("prof3" :  
<prof><nom>Loubiere</nom><prenom>Peio</prenom></prof>))
```

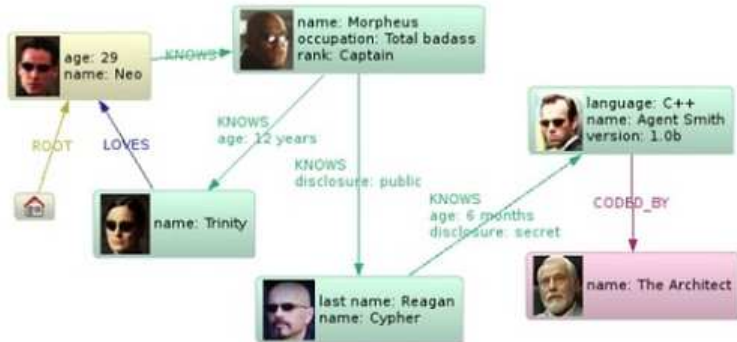
# Stockage orienté graphe

## Présentation

- Graphe dont les sommets sont les données
- Chaque sommet peut contenir un poids pour orienter une recherche
- Les relations entre ces données sont représentées par des arrêtes orientées
- Les arrêtes contiennent des attributs
- Neo4j leader !
- De nouvelles bases : Dgraph, Amazon Neptune

# Stockage orienté graphe

Présentation ([infoq.com/fr/articles/graph-nosql-neo4j](http://infoq.com/fr/articles/graph-nosql-neo4j))



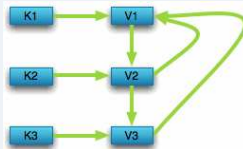
# Stockage orienté graphe

## Objectifs

- Une structure différente pour chaque donnée
- Modélise un réseau social
- Permet l'utilisation d'algorithmes de parcours de graphe

# Stockage orienté graphe

Tout est graphe... (source : neo4j)



# Stockage orienté graphe

Tout est graphe... (source : neo4j)

