NoSQL - Not Only SQL

A. Lacayrelle

Contexte

- Apparition de très grands plateformes et applications autour du Web
  - Google, Facebook, Twitter, LinkedIn, Amazon, ...
- Volume considérable de données à gérer par ces applications
  - données complexes et hétérogènes
- Croissance exponentielle des données
  - la quantité produite double environ tous les 2 ans

Big Data

Modélisation, stockage et traitement (analyse) d'un ensemble de données très volumineuses, croissantes et hétérogènes.

Les 3 V du BigData

- Volume - Volume
  - plusieurs zettaoctets générés par an sur le Web
- Variety - Hétérogénéité
  - données brutes, structurées ou pas
  - images, textes, données capteurs, ...
- Velocity - Vitesse
  - les données sont créées de plus en plus vite et nécessitent parfois d'être traitées en temps réel
  - notion de flux de données
  - Exemple en 2012
    - ⋆ 7 Go par jour pour Twitter
    - ⋆ 70000 To par seconde pour le radiotélescope "Square Kilometre Array"

Conséquences

Les volumes à gérer impliquent

- des données hétérogènes, complexes et souvent liées
  - produites par des applications parfois différentes
  - par des utilisateurs différents
  - avec des liens implicites ou explicites
- pas possible d'avoir un serveur unique
- besoin de distribuer les calculs et les données

Besoin de "scalabilité" (Passage à l'échelle)

Scalabilité

Désigne la capacité d'un produit à s'adapter à un changement d'ordre de grandeur (montée en charge).

- "scalabilité" verticale
  - augmenter la puissance des serveurs
- "scalabilité" horizontale
  - augmenter le nombre de serveurs

Nouvelles approches de stockage des données et de gestion des données

Systèmes NoSQL (Not Only SQL)

- ne remplacent pas les SGBDR (SGBD relationnels), mais les complètent
  - quantité de données énormes
  - temps de réponse
  - faible cohérence
- permettent une meilleure scalabilité dans des contextes fortement distribués
- gestion des objets complexes et hétérogènes
  - pas de schémas

Systèmes distribués

Système distribué

Système logiciel qui permet de coordonner plusieurs ordinateurs. Généralement, cette coordination se fait par envoi de messages via un réseau auquel sont reliés ces ordinateurs.

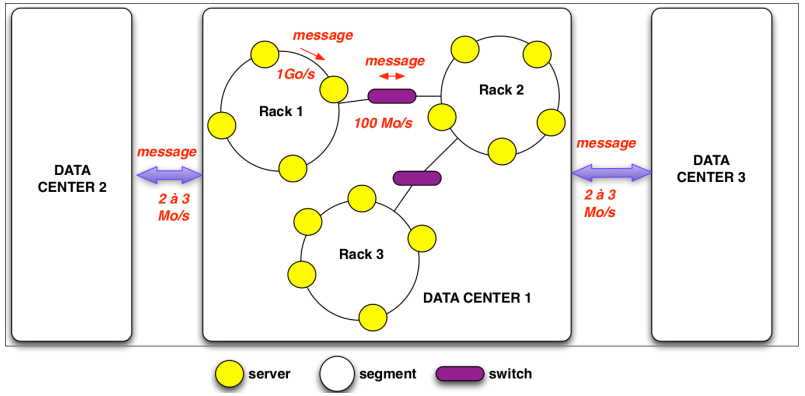
Cas particulier : Gestion des données distribuées

- accès efficace à des volumes de données très importants
- assurer l'accès même en cas d'indisponibilité de serveurs

$\Rightarrow$ utilisation des data center

Data centers

- distribution des données sur plusieurs serveurs
- utilisent des LAN avec 3 niveaux de communication
  - les serveurs sont regroupés en "rack" : *liaison réseau rapide, environ 1 Go/sec*
  - un "data center" consiste en un grand nombre de racks interconnectés par des routeurs (switches) : *liaison à 100 Mo/sec*
  - entre différents "data centers" : *communication internet à 2-3 Mo/sec*
- la communication entre serveurs s'effectue par envoi de messages
  - pas de partage de disques, ni de ressources de traitement
  - architecture "shared nothing"



Ex : Data center de Google en 2010

- 1 data center : entre 100 et 200 racks
- 1 rack : 40 serveurs
- **total :** environ 1 millions de serveurs

source : cours B. Espinasse

Pourquoi les SGBD relationnels ne répondent pas aux nouveaux besoins de gestion de données ?

Rappels sur les systèmes relationnels

- repose sur une base mathématiques
- abstraction du niveau physique
- données structurées
  - tables + contraintes d'intégrité
- langage d'interrogation riche : SQL
- efficacité
  - optimisation des requêtes
  - indexation des données
- multi-utilisateurs ⇒ gestion de la concurrence
- transaction ACID

Transaction ACID

- Atomicité
  - une transaction s'effectue entièrement ou pas du tout
- Cohérence
  - le contenu de la base doit être cohérent au début et à la fin de la transaction
- Isolation
  - les modifications d'une transaction A ne sont visibles par les autres transactions que lorsque la transaction A est validée
- Durabilité
  - une fois la transaction validée, ses mises à jour (insertion, modification, suppression) doivent perdurer, même en cas de défaillance du système

⇒ toutes les opérations de mise à jour concurrentes sont prises en compte et sérialisées tout en préservant l'intégrité des données

Pour en savoir plus ▶

Limites dans le contexte distribué

- Comment partitionner / distribuer les données ?
  - données liées sur le même serveur ?
  - et s'il y a beaucoup de liens ?
- Comment gérer les transactions et assurer la concurrence ?
- Intérêt de la réplication des données ?
- Efficacité sur les données non structurées ?

Théorème de CAP

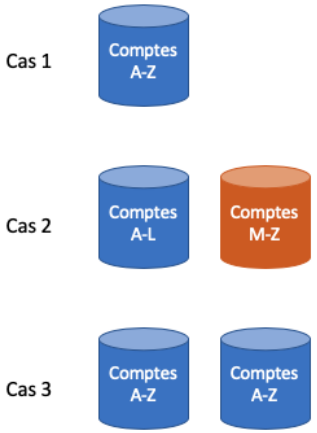
3 propriétés fondamentales dans un système distribué :

- Consistency *(cohérence)*
  - tous les noeuds du système voient les mêmes données au même moment
- Availability *(disponibilité)*
  - si un noeud tombe en panne, les données restent accessibles aux autres noeuds
- Partition tolerance *(résistance au partitionnement)*
  - le système, même partitionné, doit répondre à toute requête (chaque sous-réseau doit pouvoir fonctionner de manière autonome)

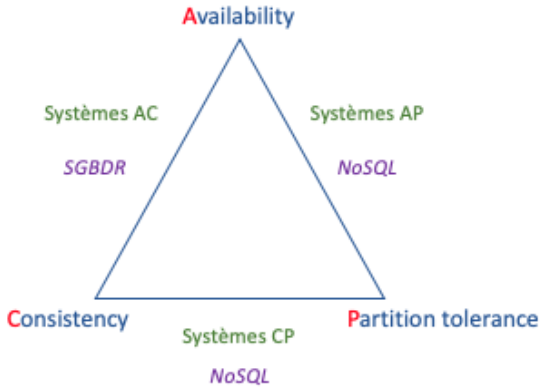
Théorème de CAP *(Brewer, 2000)*

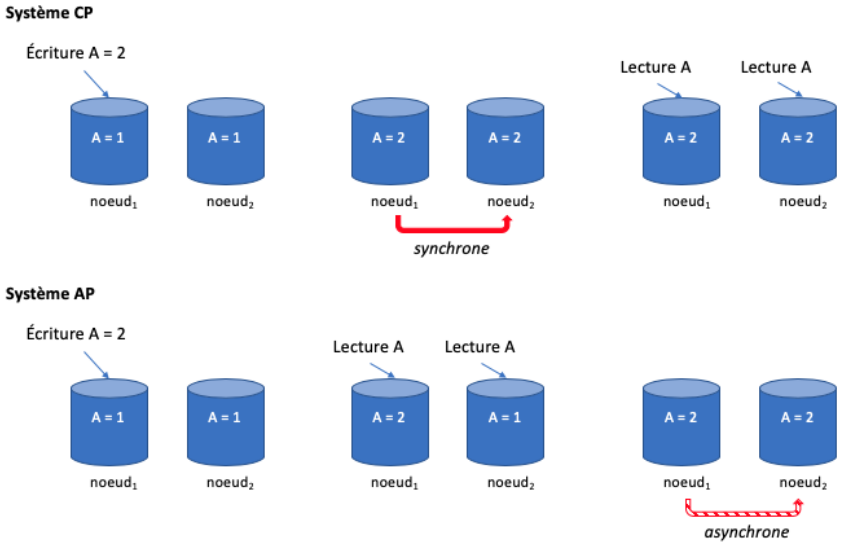
Dans un système distribué, il est impossible d'obtenir ces 3 propriétés en même temps : il faut en choisir 2 parmi les 3.

Pourquoi pas les 3 en même temps ?



- Consistency
- Availability
- Partition tolerance





Caractéristiques des SGBD NoSQL

- Pas de schéma pour les données (ou schéma dynamique)
- Données éventuellement complexes ou imbriquées
- Utilisation : peu d'écriture, beaucoup de lecture
- Distribution des données
  - parallélisation des traitements (Map Reduce)
- Réplication des données
  - privilégie la disponibilité à la cohérence (système AP) ⇒ pas de transaction en général

⇒ propriétés BASE

Propriétés BASE

- Basically Available
  - le système doit toujours être accessible (ou indisponible sur de courtes périodes)
- Soft-state
  - l'état de la BD n'est pas garanti à un instant donné (les mises à jour ne sont pas immédiates)
- Eventually consistent
  - la cohérence des données à un instant n'est pas primordiale (mais assurée à terme : verrouillage optimiste en reportant à plus tard la vérification de l'intégrité).