1 Contexte et généralités

2 Principaux modèles de bases de données NoSQL

3 Fondements des systèmes NoSQL
- Partitionnement des données
- Réplication des données
- MapReduce
- Gestion des pannes

4 Travaux pratiques

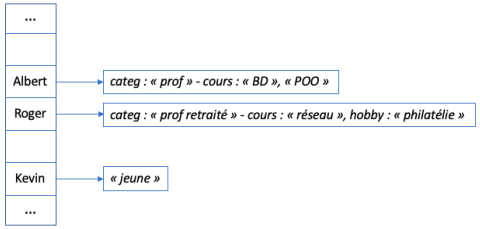
Modèles de BD NoSQL

Les principaux modèles sont :

- Modèle "Clé-Valeur"
- Modèle orienté "Colonne"
- Modèle orienté "Document"
- Modèle orienté "Graphe"

Modèle Clé-Valeur

- Base assimilée à une table de hachage distribuée
- Données représentées par des couples *(clé, valeur)*
  - *clé* unique
  - *valeur*
    - structuré ou pas (⇒ pas de schéma)
    - chaine, entier, objet, ...
- On ne peut accéder à une valeur que via sa *clé*
  - pas de requêtes sur les valeurs



- Manipulation avec 4 opérations (CRUD)
  - Create : crée une nouvelle valeur avec sa clé *create(key, value)*
  - Read : lit une valeur à partir de sa clé *read(key)*
  - Update : met à jour une valeur à partir de sa clé *update(key, value)*
  - Delete : supprime une valeur à partir de sa clé *delete(key)*
- Accès à la base via une interface `HTTP REST` depuis n'importe quel langage de programmation

Exemples de bases "clé-valeur"

- Riak (implémentation open source d'Amazon Dynamo)
- Redis (projet sponsorisé par VMWare)
- Voldemort (développé par LinkedIn en interne puis passage en open source)

Points forts

- modèle de données simple
- passage à l'échelle
- disponibilité
- évolutivité des valeurs (sous réserve d'adapter les programmes de traitement)

Points faibles

- modèle de données trop simple
- interrogation uniquement sur la clé
- couche applicative complexe vue les fonctionnalités offertes

Utilisation principale : masse de données avec des besoins de requêtage simples

- informations de sessions, logs de données
- profils, préférences utilisateurs
- données des paniers d'achat
- données capteurs
- ...

Modèle orienté colonne

- Les données sont stockées par colonne et non par ligne
  - ajout de colonne simple
  - ajout de ligne plus couteux
  - pas de stockage des valeurs NULL
- Modèle proche d'une table dans un SGBDR mais le nombre de colonnes
  - est dynamique
  - peut varier d'un n-uplet à un autre

- Colonne
  - définie par un couple *(clé, valeur)*
  - plus une estampille (pour gérer les versions et les conflits)
- Super-colonne
  - colonne contenant plusieurs colonnes ($\approx$ ligne)
- Famille de colonne
  - regroupe des colonnes ou des super-colonnes
  - les colonnes d'une famille sont stockées ensemble

Exemples de bases orientées "colonne"

- HBase (version open source basé sur BigTable de Google et Hadoop DFS)
- $Cassandra_{V1}$ (projet né chez Facebook qui est réalisé par Apache)
- SimpleDB (développé par Amazon)

Points forts

- bon passage à l'échelle
- indexation des colonnes
- résultat des requêtes en temps réel

Points faibles

- pas de données structurées complexes (données interconnectées)
- à éviter pour les lectures de données complexes
- modification de la structure en colonne $\Rightarrow$ maintenance
- requêtes doivent être pré-écrites

Utilisation principale : traitements massifs d'analyse de données

- analyse de clientèle et recommandation (Netflix, sociétés de TV, ...)
- optimisation de la recherche (Ebay)
- traitement des données et "Business Intelligence" (Adobe)
- journalisation d'événements, de compteur
- ...

Modèle orienté document

- BD NoSQL orientée document = collection de documents
  - manipulation de structures complexes
  - basée sur le modèle "clé-valeur"
    - valeur = document
  - "schemaless"
    - pas nécessaire de définir la structure des documents au préalable
    - ⇒ grande flexibilité
- Document
  - structure arborescente
  - contient des champs, chaque champ ayant une valeur
  - souvent **JSON** ou XML
- Manipulation
  - interface d'accès HTTP REST permettant
    - opérations CRUD des BD clé-valeur
    - requêtes sur le contenu des documents

Exemples de bases orientées "document"

- MongoDB
- CouchDB
- Terrastore

Points forts

- modèle de données simple mais puissant
- bon passage à l'échelle
- pas de maintenance de la BD pour ajouter/supprimer des colonnes
- expressivité des requêtes

Points faibles

- inadapté pour des données interconnectées
- modèle de requête limité (comparé au SQL)
- baisse des performances pour de grosses requêtes

Utilisation

- outil des gestion de contenu (CMS)
- catalogues de produit
- web analytique
- gestion de données semi-structurées
  - enregistrement d'événements
  - stockage de profil utilisateur
  - ...
- ...

Modèle orienté graphe

- Modélisation, stockage et manipulation de données complexes liées par des relations
- Basé sur la théorie des graphes
  - notions de noeuds, de relations et de propriétés
- Noeud
  - document
- Arc
  - orienté
  - possède des propriétés *(nom, date, ...)*

Exemples de bases orientées "graphe"

- Neo4J
- OrientDB

Points forts

- modèle de données riche et évolutif
  - bien adapté pour la modélisation de nombreuses relations
- bonne performance pour exploiter les données liées
- langage d'interrogation

Points faibles

- fragmentation des données problématique quand il y a beaucoup de relations

Utilisation

- moteur de recommandation
- informatique décisionnelle
- web sémantique
- internet des objets
- données liées
- réseaux sociaux
- réseaux de transport
- ...