NoSQL

Définition NoSQL :

Pourquoi le NoSQL ?

Le NoSQL est apparu afin de contrer la dominance des bases de données relationnelles dans le domaine de l'internet. En effet, un des problèmes récurrents des bases de données relationnelles est la perte de performance lorsque l'on doit traiter un très gros volume de données. De plus, la multiplication des architectures distribués a apporté le besoin de disposer de solution s'adaptant nativement aux mécanismes de réplication des données et de gestion de la charge.

Caractéristiques générales des BD NoSQL

- Adoptent une représentation non relationnelle de données.
- Ne remplacent pas les BDR mais sont une alternative, un complémentapportant des solutions plus intéressantes dans certains contextes.
- Apportent une plus grande performancedans le contexte des applications Web avec des volumétries de données exponentielle.
- Utilisent une très forte distributionde ces données et des traitements associés sur de nombreux serveurs.
- Pas de schémapour les données ou schéma dynamique.
- Données de structures complexesou imbriquées.
- Données distribuées : partitionnement horizontal des données sur plusieurs nœuds (serveurs) généralement par utilisation d'algorithmes MapReduce.
- Réplication des donnéessur plusieurs nœuds.
- Privilégient la Disponibilité à la Cohérence (théorème de CAP) : AP (Disponible + Résistant au partitionnement) plutôt que CP (Cohérent + Résistant au partitionnement).
  ⇒ N'ont en général pas de gestion de transactions.
- Mode d'utilisation : Peu d'écritures, beaucoup de lectures.

Types de base de données NoSQL :

BDD
Clé / valeur

- Ces SGBD stockent les éléments sous forme d'identificateurs alphanumériques faisant référence aux clés. Chaque clé a des valeurs associées.,
- La valeur peut être simples chaînes de texte ou plus complexes listes et ensembles et(blob, json, image..etc.),
- Leger et compact.

BDD Orienté Document

- Conçu pour gérer et stocker les documents,
- Pas de schéma,
- Ces documents sont encodés dans une format standard d'échange de données tel que XML, JSON (notation d'option JavaScript) ou BSON (JSON binaire),
- Généralement, un modèle d'échange de type JSON (BSON), qui prend en charge les listes, cartes, dates, objets avec imbrication,
- Modèle de requête: JavaScript ou personnalisé,
- Agrégations: map / reduce.

BDD Orienté Colonne

- Elle consiste en une paire clé/valeur dans laquelle la valeur consiste en un ensemble des colonnes,
- Les bases de données de familles de colonnes sont représentées dans des tables, chaque paire clé/valeur étant une ligne,
- Toutes les données associées peuvent être regroupées dans une même famille,
- Conçu pour stocker une quantité énorme des données,
- Facile pour le scaling horizontale,

BDD Orienté Graph

- Elles remplacent les tables relationnelles par des graph relationnels structurés d'appariements interconnectés clé/valeur.
- La base de données relationnelle est une collection de tables faiblement connectées alors que la base de données orienté graph sont un graphe multi-relationnel ,
- Elles sont utiles lorsque vous êtes plus intéressé par les relations entre données que par les données elles-mêmes ,
- Elles son optimisé pour les relations traversantes et non pas pour les requêtes.

Théorème ACID :

ACID

- **A**tomicity (Les tâches d'une transaction est exécutée ou aucune d'entre elles. C'est le principe du tout ou rien. Si un élément d'une transaction échoue, toute la transaction échoue)
- **C**onsistency (La BDD doit rester dans un état cohérent au début et à la fin d'une transaction. il n'y a jamais de transactions à moitié terminées).
- **I**solation (Aucune transaction n'a accès à une autre transaction dont l'état n'est pas terminé. Ainsi, chaque transaction est indépendante en soi. Cela est nécessaire pour la cohérence des transactions dans une base de données).
- **D**urability (Une fois la transaction terminée, elle persistera et ne pourra plus être annulée, Elle survivra aux pannes du système, aux coupures de courant et à d'autres types de pannes du système).

Théorème CAP :

Théorème de CAP

Un système distribué ne peut prendre en charge que deux des caractéristiques suivantes:

- **C**onsistency **(**toutes les nœuds dans un system distribué renvoie la même valeur**)**
- **A**vailability (Chaque nœud non défaillant renvoie une réponse pour toutes les demandes de lecture et d'écriture dans un délai raisonnable)
- **P**artition Tolerance **(**Le système continue de fonctionner et maintient sa cohérence malgré les partitions de réseau **)**

NoSQL avantages :

Avantages

- Conçu pour les systèmes distribués,
- Elle n'a pas de schéma (flexible),
- Base transactions,
- Les requête sont simple (pas de jointure),
- Mise à l'échelle horizontale (scale out) ,
- Développement rapide .

Inconvénients :

Inconvénient

- Elle n'a pas de schéma,
- Pas un langage de requête standard,
- Pas des transaction ACID.

Conclusion :

Conclusion

- Les exigences en matière de calcul et de stockage des applications telles que les analyses Big Data, la Business Intelligence et les réseaux sociaux sur des jeux de données péta-octets nous ont amenés à passer de la base de données SQL à la base de données NoSQL,
- Cela a conduit au développement de bases de données NOSQL non relationnelles distribuées, Scalable horizontalement,
- MongoDB est le plus demandé,
- Chaque Type de BDD a ses avantage et ses incovients il faut choisir la BDD qui vous convient selon vos besoins (Le bon outil pour le bon travail),
- Généralement vous allez vous trouver utiliser plusieurs BDD en même temps.
- Il n' y a pas des relations dans les BDD clé/valeur ,BDD orienté document et BDD orienté colonne.