

REPUBLIQUE DEMOCRATIQUE DU CONGO
Université catholique Maria Malkia
Département informatique



Historique de NoSQL

Présenter par : YENGA NTETI JOSUE
Promotion : Bac2 ISI
Travail pratique élaborer dans le cadre du cours
De Base de données

Prof Patrick Kasonga

28 Octobre 2021

Introduction

Qu'est-ce qu'une base de données ? A quoi cela peut-il servir ? Une base de données est un dispositif servant à stocker des données de différents types telles que des lettres, des chiffres, des dates... de manière structurée. Aujourd'hui, pratiquement tous les programmes informatiques utilisent une base de données afin de pouvoir consulter, ajouter, modifier des informations diverses. La gestion de ces données, ainsi que l'accès se fait à travers une suite de programme qu'on appelle SGBD (Système de gestion de base de données).

Le SQL qui est aujourd'hui le langage le plus utilisé par les systèmes informatiques, fut développée par IBM en 1970, sous le nom SEQUEL (Structured English Query Language), par la suite renommé SQL en 1975 pour des raisons de propriété intellectuelle, soit de marque déposée. Il sera finalement normalisé internationalement par ISO en 1987.

Bien que le terme soit entendu pour la première fois en 1988 par Carlo Strozzi qui présentait ça comme un modèle de base de données plus léger et sans interface SQL, c'est en 2009, lors de la rencontre « meetup » NoSql de San Francisco, qu'il prend un essor important. Durant cette conférence, il y sera présenté des solutions de projet telles que Voldemort, Cassandra Project, Dymomite, HBase, Hypertable, CouchDb, MongoDB. Ce « meetup » sera considérée comme l'inauguration de la communauté des développeurs de logiciel NoSql.

L'émergence du NoSQL

Avec l'augmentation de la bande passante sur internet, ainsi que la diminution des coûts des matériels informatiques, de nouvelles possibilités ont vu le jour dans le domaine de l'informatique distribuée. Le passage au 21ème siècle par la révolution du WEB 2.0 a vu le volume de données de certaines entreprises augmenter considérablement. Ces données proviennent, essentiellement, de réseaux sociaux, base de données médicales, indicateurs économiques, etc. L'informatisation croissante de traitement en tout genre a eu pour conséquence une augmentation exponentielle de ce volume de données qui se compte désormais en pétaoctets, les anglo-saxon l'ont nommé le Big Data. La gestion de ces volumes de données est donc devenue un problème que les bases de données relationnelles n'ont plus été en mesure de gérer. Le NoSQL regroupe donc de nombreuses bases de données qui ne se reposent donc plus sur une logique de représentation relationnelle. Il n'est toutefois pas simple de définir explicitement ce qu'est une base de données NoSql étant donné qu'aucune norme n'a encore été instaurée.

Pourquoi l'alternative du NoSQL ?

Le besoin fondamental auquel répond le NoSQL est la performance. Afin de résoudre les problèmes liés au « Big data », les développeurs de sociétés telles que Google et Amazon ont procédé à des compromis sur les propriétés ACID des SGBDR. Ces compromis sur la notion relationnelle ont permis aux SGBDR de se libérer de leurs freins à la scalabilité

horizontale. Un autre aspect important du NoSql est qu'il répond au théorème de CAP qui est plus adapté pour les systèmes distribués.

L'autre avantage du NoSQL est d'ordre financier. Servant à manipuler de gros volumes de données, il est donc destiné aux grandes entreprises. Dès 2010, le NoSql a commencé à s'étendre vers les plus petites entreprises. Majoritairement des start-ups n'ayant pas les moyens d'acquérir des licences Oracle qui ont donc développé leurs propres SGBD en imitant les produits Google et Amazon.

Les différences entre le NoSQL et le SQL

Pour commencer, il est important de savoir que le SQL n'est pas un modèle relationnel en soit, mais un langage de manipulation de données conçu autour du modèle relationnel. Les bases de données NoSql n'ont pas pour but de s'éloigner de ce langage mais du modèle relationnel. Nous allons voir les facteurs différenciateurs.

Les principes du relationnel

1. Le modèle de données

Le modèle relationnel basé est sur un modèle mathématique qui est celui de la notion des ensembles. Chaque ensemble ici est représenté par une table, et ces attributs sont quant à eux représentés par des colonnes. L'un des principes fondamentaux est justement cette notion de relation entre tables à l'aide de cardinalités, clés primaires et clé étrangères, ceci implique au préalable une étude minutieuse sur la modélisation du schéma de la base de données. Comme par exemple les tables dont le système aura besoin, ses attributs, les relations possibles entre différentes tables, etc. Une fois ce modèle mis en place dans un SGBDR, il est difficile de changer la structure de celui-ci et ceci pose par conséquent des problèmes lors de sa réingénierie.

Le mouvement NoSql lui est plus pragmatique, basé sur des besoins de stockage de données ainsi qu'une liaison plus forte avec les différents langages clients. La plupart des moteurs de base de données NoSQL n'utilisent pas de schémas prédéfinis à l'avance, d'où l'appellation « schema-less », ce qui signifie sans schéma. Ainsi le moteur de la base de données n'effectue pas de vérification et n'impose pas de contrainte de schéma, les données étant organisées du côté client du code. Toutefois le principe « schema-less » est théorique et ne se vérifie pas entièrement en pratique. Le maintien d'une structure de données homogènes est important, pour des questions d'indexation, de recherche par critère ou tout simplement pour des raisons de logique.

Différents types de base de données NoSQL

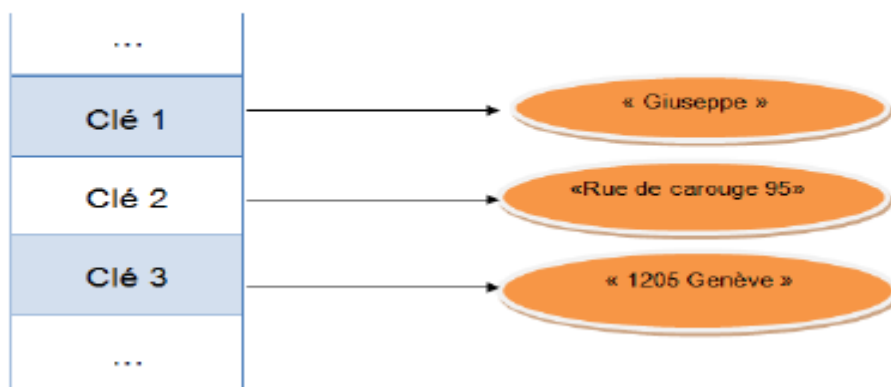
Les bases de données NoSql sont donc une catégorie de base de données qui n'est plus fondée sur l'architecture classique des bases relationnelles, et non pas un type à part entière. Quatre grandes catégories se distinguent parmi celles-ci.

Les bases de données clé-valeur

La base de données de type clé-valeur est considérée comme la plus élémentaire. Son principe est très simple, chaque valeur stockée est associée à une clé unique. C'est uniquement par cette clé qu'il sera possible d'exécuter des requêtes sur la valeur.

La structure de l'objet stocké est libre et donc à la charge du développeur de l'application. Un avantage considérable de ce type de base de données est qu'il est très facilement extensible, on parle donc de scalabilité horizontale. En effet dans le cas où le volume de données augmente, la charge est facilement répartissable entre les différents serveurs en redéfinissant tout simplement les intervalles des clés entre chaque serveur.

Représentation clé-valeur 1



En ce qui concerne le besoin de la scalabilité verticale, celle-ci est très fortement réduite par la simplicité des requêtes qui se résument à PUT, GET, DELETE, UPDATE. Ce type de base de données affiche donc de très bonnes performances en lecture/écriture et tendent à diminuer le nombre de requêtes effectuées, leur choix étant restreint.

Néanmoins, les requêtes ne peuvent être exécutées uniquement sur les clés à cause de la simplicité du principe. Permettant un stockage de données sans schéma, le client est contraint à récupérer tout le contenu du blob et ne peut obtenir un résultat de requêtes plus fin tel que des résultats de requêtes SQL pourraient permettre.

Ce type de base de données orienté clé-valeur implique donc des cas d'utilisation très spécifiques, à cause de leur simplicité de modèle et de la fine palette de choix de requêtes proposées. Ces systèmes sont donc principalement utilisés comme dépôt de données à condition que les types de requêtes nécessitées soient très simples. On les retrouve comme système de stockage de cache ou de sessions distribuées, particulièrement là où l'intégrité des données est non significative