

1- Introduction

NoSQL ou « Not Only SQL » est un mouvement très récent (2009), qui concerne les bases de données. L'idée du mouvement est simple : proposer des alternatives aux bases de données relationnelles pour coller aux nouvelles tendances et architectures du moment, notamment le Cloud Computing. Les axes principaux du NoSQL sont une haute disponibilité et un partitionnement horizontal des données, au détriment de la consistance. Alors que les bases de données relationnelles actuelles sont basées sur les propriétés ACID (Atomicité, Consistance, Isolation et Durabilité). NoSQL signifie “Not Only SQL”, littéralement “pas seulement SQL”. Ce terme désigne l'ensemble des bases de données qui s'opposent à la notion relationnelle des SGBDR. La définition, “pas seulement SQL”, apporte un début de réponse à la question “Est ce que le NoSQL va tuer les bases relationnelles?”. En effet, NoSQL ne vient pas remplacer les BD relationnelles mais proposer une alternative ou compléter les fonctionnalités des SGBDR pour donner des solutions plus intéressantes dans certains contextes. Le NoSQL regroupe de nombreuses bases de données, récentes pour la plupart, qui se différencient du modèle SQL par une logique de représentation de données non relationnelle. Leurs principaux avantages sont leurs performances et leur capacité à traiter de très grands volumes de données. En revanche, dans les projets, il ne faut pas opposer ces deux approches mais bien souvent les faire cohabiter ! Cette technologie (le NoSQL) ne vise finalement pas à remplacer les SGBD traditionnels mais plutôt à les compléter en déportant une partie de la charge des traitements et de stockage de données vers des serveurs-tiers (dans les architectures web classiques).[15]

2- Les concepts forts du NoSQL

Les bases de données NoSQL reposent essentiellement sur plusieurs aspects qui font leurs forces et justifient leur usage aujourd'hui des géants du web. Il s'agit principalement du partitionnement horizontal des données, du fait que ces bases sont sans schéma et donc une grande flexibilité du schéma de données.[15]

3- Les types de bases de données NoSQL

Dans la mouvance NoSQL, il existe une diversité d'approches classées en quatre catégories. Ces différents systèmes NoSQL utilisent des technologies fortes distinctes. Les différents modèles de structure sont décrits comme suit :

3.1 Base de données orientée clé- valeur Ce modèle peut être assimilé à une hashmap¹ distribuée. Les données sont, donc, simplement représentées par un couple clé/valeur. La valeur peut être une simple chaîne de caractères, un objet sérialisé... Cette absence de structure ou de typage ont un impact important sur le requêtage. En effet, toute l'intelligence portée auparavant par les requêtes SQL devra être portée par l'applicatif qui interroge la BD. Néanmoins, la communication avec la BD se résumera aux opérations PUT, GET et DELETE. Les solutions les plus connues sont Redis, Riak et Voldemort créé par LinkedIn. [14]

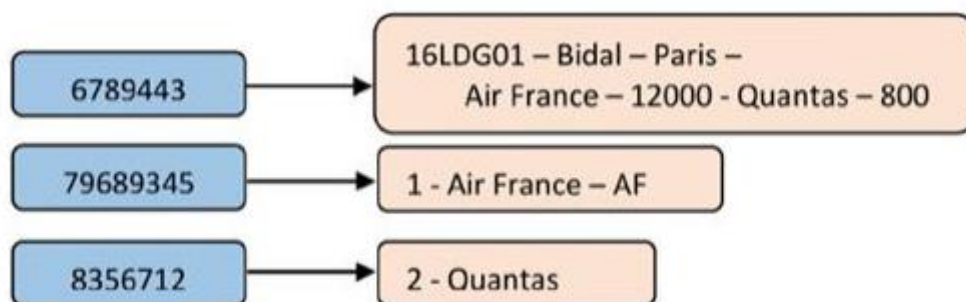


Figure 2.1 :modèle de données clé-valeur [14]

Panorama des bases orientées clé- valeur [17]

La plupart des solutions se basent sur le papier d'**Amazon Dynamo** :

- **Voldemort** : initié en 2008 au sein de la société LinkedIn, peut être stocké en mémoire et sur disque, extensible à la fois par réplication horizontale et verticale, favorise la disponibilité et la tolérance au morcellement ;

¹ Il existe dans le langage Java de nombreuses classes pour stocker ses données et les parcourir facilement. La classe Hashmap permet de stocker des données sous la forme clé - valeur dans une variable.

- **Riak** : initié en 2008, par la société Basho, pas de point unique de défiance, favorise la cohérence et la tolérance au morcellement ;
- **Amazon DynamoDB** : base de données en tant que service BDaaS fournie par Amazon.

D'autres solutions se basent sur **memcached** conçue par Brad Fitzpatrick :

- **Redis** initié en 2009 par Salavador Sanfilippo, peut être stocké en mémoire et sur disque ;
- **OrientDB** initié en 2010 par Luca Garulli, est une base multi-modèle qui peut être configurée pour supporter la cohérence et la tolérance au morcellement, ainsi que la disponibilité selon la configuration du cluster.

3.2 Base de données orientée document :

Ce modèle se base sur le paradigme clé valeur. La valeur, dans ce cas, est un document de type JSON ou XML. L'avantage est de pouvoir récupérer, via une seule clé, un ensemble d'informations structurées de manière hiérarchique. La même opération dans le monde relationnel impliquerait plusieurs jointures. Pour ce modèle, les implémentations les plus populaires sont CouchDB d'Apache, RavenDB (destiné aux plateformes .NET/Windows avec la possibilité d'interrogation via LINQ) et MongoDB.[14]

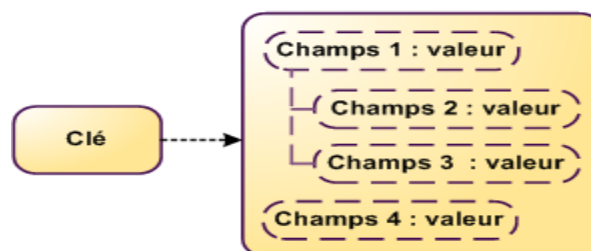


Figure 2.2 :modèle de données orienté documents [17]

Panorama des bases orientées documents [17]

- **MongoDB** : initié en 2007 par la société 10gen pour créer une Plateforme (PaaS), favorise la cohérence et la tolérance au morcellement.
- **CouchDB** : initié en 2005 par Damien Katz pour la mise en place d'un système de stockage dédié au site web, favorise la disponibilité et la tolérance au morcellement.
- **OrientDB** : initié en 2010 par Luca Garulli, est une base multi-model.
- **DocumentDB** .

3.3 Base de données orientée colonne : Ce modèle ressemble à première vue à une table dans un SGBDR à la différence qu'avec une BD NoSQL orientée colonne, le nombre de colonnes est dynamique. En effet, dans une table relationnelle, le nombre de colonnes est fixé dès la création du schéma de la table et ce nombre reste le même pour tous les enregistrements dans cette table. Par contre, avec ce modèle, le nombre de colonnes peut varier d'un enregistrement à un autre ce qui évite de retrouver des colonnes ayant des valeurs NULL. Comme solutions, on retrouve principalement HBase (implémentation Open Source du modèle BigTable publié par Google) ainsi que Cassandra (projet Apache qui respecte l'architecture distribuée de Dynamo d'Amazon et le modèle BigTable de Google).[14]

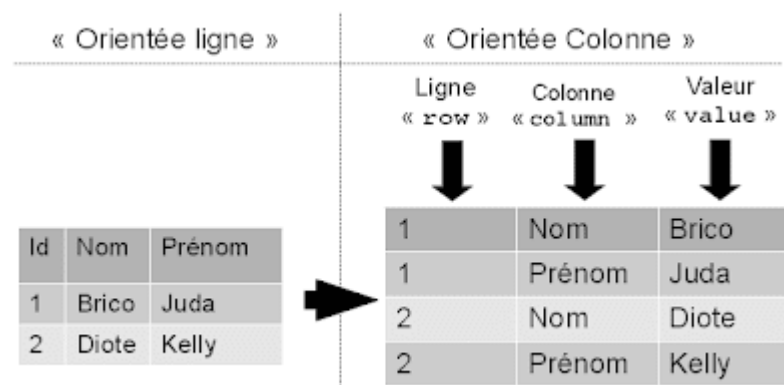


Figure 2.3 : modèle de données orienté colonnes [16]

Panorama des bases orientées colonnes [17]

- **Cassandra** : initiée et développée par Facebook en 2008, pour son système de mail puis devenu un projet de la fondation Apache, Cassandra est très utile pour le traitement temps réel et favorise la disponibilité et la tolérance au morcellement.
- **HBase** : initiée en 2007 par la société Powerset, qui est un clone open source de BigTable, qui utilise le système de fichier **HDFS** de **Hadoop**, **ZooKeeper** pour la coordination et a un accès direct (aléatoire) à la donnée favorisant la cohérence et la tolérance au morcellement, avec une latence inférieure à la seconde ce qui le rend utile pour les traitements temps réel.

3.4 Base de données orientée graphe

Modèles orienté graphes Le modèle orienté graphes se distingue des modèles précédents car il ne peut prendre en compte nativement la notion d'agrégats d'information proposée par [Evans 2004]. Le modèle orientés graphes est fondé sur la théorie des graphes. Le modèle orienté graphes repose sur trois notions ; nœud, relation et propriété. Chaque nœud possède des propriétés. Les relations relient les nœuds et possèdent éventuellement des propriétés. Ce type d'approche facilite les requêtes navigationnelles entre les nœuds en suivant les relations qui les relient : chaque nœud a un pointeur physique vers les nœuds voisins permettant une recherche locale rapide.[8]

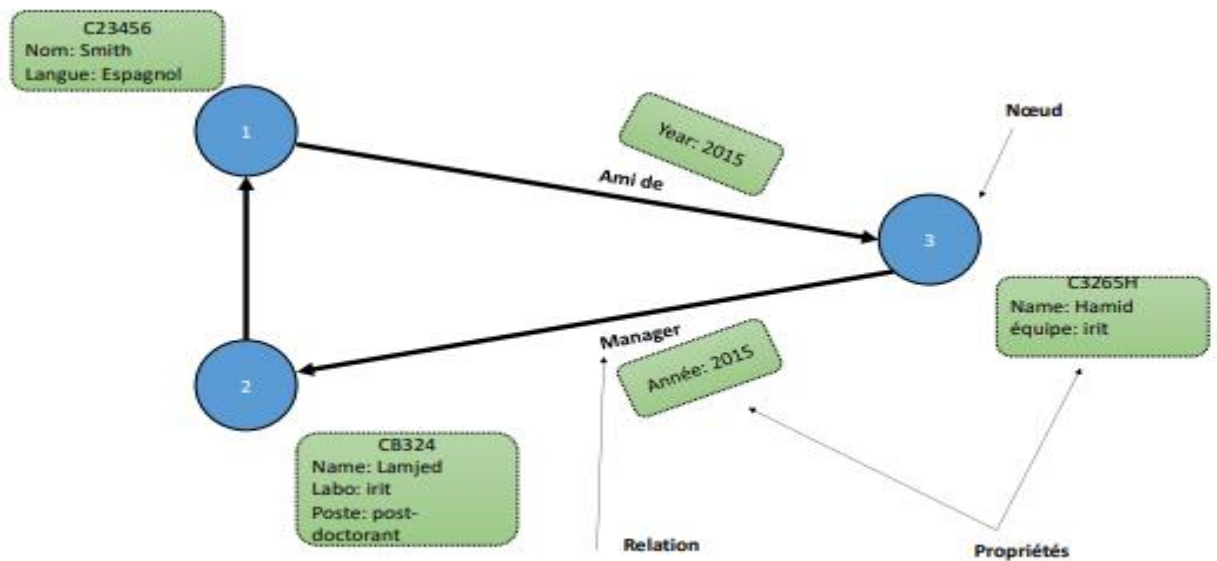


Figure 2.4 Principe du modèle orienté graphes [8]

4 -Comparaison des modèles nosql

Dans le Tab. 2.2, nous synthétisons les caractéristiques de ces quatre modèles NoSQL ainsi que le modèle relationnel. Nous considérons 3 caractéristiques principales : – La structure de la valeur peut être soit atomique (plus petite valeur manipulable) soit composée (valeur composée de plusieurs éléments manipulables). – Le schéma peut être statique (schéma unique pour un ensemble d'enregistrements) ou dynamique (schéma différent pour chaque enregistrement).

	Relationnel	Clé-valeur	Colonnes	Documents	Graphes
Structure	atomique	atomique	atomique & composée	atomique & composée	atomique & composée
Schéma	statique	dynamique	dynamique (colonnes)	dynamique	dynamique
Requêtage	SQL	CRUD	CRUD + langage	CRUD + langage	CRUD + langage
Exemple de systèmes	Oracle, PostgreSQL	Redis	HBase, Cassandra	MongoDB, CouchDB	Neo4j, OrientDB

Tab. 2.1Comparatif des modèles NoSQL [8]

5- Comparaison des moteurs Nosql [17]

Le tableau comparatif suivant (Tab. 2.1) montre d'une manière synthétique les différents moteurs NoSQL et leurs caractéristiques sans être exhaustif.

	Format de stockage	Transactions	Relation	Haute disponibilité	Fonctions avancées
HBase	Colonnes	Non	Oui	Cluster	
Cassandra	Colonnes	Non	Oui	Cluster	Réplication avancée
Redis	Clé/Valeur	Non	Non	Cluster	Communication asynchrone
Riak	Clé/Valeur, Série temporelle			Cluster	Tolérance aux pannes
MongoDB	Document JSON	Non	Non	Cluster	Agrégation
OrientDB	Multi-Format	Oui	Oui	Cluster	Transactions et sécurité
Neo4j	Graphe	Oui	Oui	Cluster	Intégration avec d'autres solutions NoSQL

Tab. 2.2 – tableau comparatif des moteurs NoSQL [17].

6- Conclusion

Le modèle clé-valeur est le modèle NoSQL fondamental. Il bénéficie d'une structure simple qui lui permet un gain de performances importantes. Cependant il ne permet pas une manipulation fine de la valeur. Cette limite a motivé le développement de nouveaux modèles orientés colonnes et orienté documents, qui peuvent être considérés comme une forme évoluée du modèle orienté clé-valeur. Ces deux modèles introduisent une structuration de la valeur, mais selon des principes orthogonaux. La valeur peut ainsi être soit atomique, soit composée. Ces modèles se distinguent par le stockage des données qui est effectué soit par document (horizontal), soit par ligne décomposée en familles de colonnes (vertical). Un quatrième modèle NoSQL repose sur le modèle orienté graphes. Il se caractérise par une structure basée sur la théorie des graphes étiquetés. [8]