



Cati Sicpa

Ce document est mise à disposition selon les termes de la
[Licence Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/)

Solutions de Stockages NoSQL

16 avril 2018

Introduction

Le besoin d'organiser les données, potentiellement de grandes quantités de données, afin d'en optimiser la conservation et la restitution, a toujours été au cœur de l'informatique. Stocker et retrouver, voilà les défis de la base de données. Les bases de données relationnelles sont les premières solutions proposées pour faire face à ces défis. Actuellement, les moteurs de bases de données relationnelles traditionnels semblent totalement dépassés pour la manipulation des gros volumes de données qui se compte en pétaoctets (100 000 téraoctets). C'est ce que les Anglo-Saxons ont appelé le Big Data [8]. Aujourd'hui, la recherche de nouvelles façon – ou en tout cas des façons complémentaires – de gérer et de traiter ces gros volumes de données sont considérés comme un nouveau défi de l'informatique. Un nombre croissant de solutions à ce nouveau défi a vu le jour ces 10 derniers années. Ces solutions sont tournées vers les bases de données non-relationnelles communément appelées bases de données NoSQL (Not Only SQL). Il existe actuellement plus de 225 solutions de stockages NoSQL [10]. Elles sont divisées en quatre catégories principales selon le schéma ou le modèle de données qu'elles utilisent [2, 9] : les bases orientées clé-valeur, les bases orientées documents, les bases orientées colonnes et les bases orientées graphes. Dans la suite, nous décrivons chacune de ces catégories tout en indiquant les cas d'usages appropriés.

1 Les bases orientées clé-valeur

Dans le modèle clé-valeur, les valeurs enregistrées correspondent aux clés spécifiques. La clé représente une information précise et atomique, alors que la valeur peut être complexe et représenter un tableau ou une liste, ou tout autre objet quelconque. Chaque clé est unique et la valeur associée est considérée comme un scalaire. Il n'y pas de notion de schéma de données dans un tel système. Chaque clé peut représenter un objet donc la structure est complètement différente de celle des objets représentés par les autres clés. Ainsi, les clés sont les seules manières d'accéder aux valeurs. Le regroupement des paires clé-valeur en collections est la seule manière d'introduire une structure au modèle de données. Une telle collection peut être comparée à une table dans une base de données relationnelle. Les clés dans cette collection peuvent ainsi être considérées comme les identifiants des lignes dont la seule colonne contient les objets qu'elles représentent. Ce modèle n'offre pas la possibilité d'exploiter ni de contrôler la structure des valeurs à l'aide des requêtes. Les bases orientée

clé-valeur ont en général comme objectif d'offrir les meilleures performances possibles en écriture et en lecture. Nous listons ici quelques exemples de bases de données orientées clé-valeur : Redis, MemCached, Riack KV, Amazon DynamoDB, OrientDB, BerkeleyDB, Aerospike. La base de données orientée clé-valeur la plus utilisée est Redis [6, 3].

2 Les bases orientées documents

Dans le modèle orienté document, les données sont enregistrées sous forme de paires clé-valeurs englobées dans ce qui est appelé « document ». Toutes les clés dans un document doivent être uniques, et chaque document contient une clé spéciale « ID », qui est aussi unique à l'intérieur d'un ensemble de documents appelé « collection ». Ainsi, une clé ID identifie explicitement un document dans une collection. Le format privilégié pour le document est en général du JSON (JavaScript Object Notation), un format de définition de données qui contient une structure hiérarchique de propriétés et de valeurs. Ces dernières peuvent être scalaires ou contenir des tableaux ou des sous-documents. Une collection peut être comparée à une table dans une base de données relationnelle. Les documents sont comme les lignes de cette table. Ce modèle offre la possibilité d'exploiter et de contrôler la structure des valeurs à l'aide des requêtes. Les bases orientées documents sont très souvent utilisées pour structurer les données qui contiennent un niveau profond d'imbrications. Nous listons ici quelques exemples de bases de données orientées documents : MongoDB, Amazon DynamoDB, Couchbase, CouchDB, Microsoft Azure Cosmos DB. La base de données orientée documents la plus utilisée est MongoDB [4].

3 Les bases orientées colonnes

Les bases de données orientées colonnes sont assez proches conceptuellement des tables relationnelles. Ces tables comportent des colonnes avec un type de données. Une table comporte des clés uniques de lignes, souvent appelées rowkeys. À l'intérieur de la table, les colonnes peuvent être regroupées en familles de colonnes selon la répartition que l'on désire pour le stockage des données dans les différentes machines. D'une ligne à l'autre, les colonnes présentes dans une famille peuvent varier selon les données à stocker. Ce modèle offre la possibilité d'exploiter et de contrôler la structure des valeurs à l'aide des requêtes. Les bases orientées colonnes sont appropriées pour le stockage et la manipulation des gros volumes de données dans des clusters très grands, parce que leur modèle de données permet un partitionnement efficace. Nous listons ici quelques exemples de bases de données orientées colonnes : Cassandra, HBase, Microsoft Azure Cosmos DB, Datastax Enterprise, Accumulo, Microsoft Azure Table Storage. La base de données orientée colonnes la plus utilisée est Cassandra [7].

Les bases orientées graphes

Les bases de données orientées graphes sont différentes des autres types de bases de données NoSQL. Elles visent à modéliser et à stocker les données sous formes de graphe

essentiellement composé d'un ensemble de noeuds ou sommets et d'arêtes. Chaque noeud représente une entité labélisée ou typée et chaque arête, une connexion ou une relation entre deux noeuds. Chaque noeud d'une base de données orientée graphes est défini par un identifiant unique, un ensemble d'arêtes sortantes et/ou entrantes, ainsi qu'un ensemble de propriétés ou attributs exprimés sous la forme de paires clé-valeur. Chaque arête se définit à son tour par un identifiant unique, un noeud de départ et un noeud d'arrivée, ainsi qu'un ensemble de propriétés ou attributs. Un label peut être comparée à une table dans une base de données relationnelle. Les noeuds ayant ce label sont comme les lignes de cette table. Les colonnes se composent non seulement les attributs de ces noeuds, mais aussi de ceux des relations qui leurs sont associées. Ce modèle offre la possibilité d'exploiter et de contrôler la structure des valeurs à l'aide des requêtes. Les bases de données orientées graphes conviennent particulièrement à la manipulation et à l'analyse des données impliquant des relations complexes entre les entités. Ce qui explique le grand intérêt qu'elles suscitent pour l'exploitation des données issues des réseaux sociaux. Nous listons ici quelques exemples de bases de données orientées graphes : Neo4j, Microsoft Azure Cosmos DB, Datastax Enterprise, OrientDB Giraph, FlockDB. La base de données orientée graphes la plus utilisée est Neo4j [5, 1].

Références

- [1] Benhenni, A.L., Bois, F.X. : Bases de données orientées graphes avec Neo4j : Manipuler et exploiter vos bases de données orientées graphes. Eyrolles (2016)
- [2] Bruchez, R. : Les bases de données NoSQL et le Big Data : Comprendre et mettre en oeuvre. Eyrolles (2015)
- [3] Das, V. : Learning Redis : Design efficient web and business solutions with Redis. Packt Publishing (2015)
- [4] DB-Engines : Db-engines ranking of document stores. Tech. rep. (2018). URL <https://db-engines.com/en/ranking/document+store>
- [5] DB-Engines : Db-engines ranking of graph dbms. Tech. rep. (2018). URL <https://db-engines.com/en/ranking/graph+dbms>
- [6] DB-Engines : Db-engines ranking of key-value stores. Tech. rep. (2018). URL <https://db-engines.com/en/ranking/key-value+store>
- [7] DB-Engines : Db-engines ranking of wide column stores. Tech. rep. (2018). URL <https://db-engines.com/en/ranking/wide+column+store>
- [8] Frampton, M. : Big Data Made Easy : A Working Guide to the Complete Hadoop Toolset. Apress (2014)
- [9] Makris, A., Tserpes, K., Andronikou, V., Anagnostopoulos, D. : A classification of nosql data stores based on key design characteristics. *Procedia Computer Science* **97**, 94 – 103 (2016). DOI <https://doi.org/10.1016/j.procs.2016.08.284>. URL <http://www.sciencedirect.com/science/article/pii/S1877050916321007>. 2nd International Conference on Cloud Forward : From Distributed to Complete Computing
- [10] NoSQL : List of nosql databases. Tech. rep. (2018). URL <http://nosql-database.org/>