



INRAE



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/)

MongoDB et Cassandra

17 avril 2018

1 Quand aller vers MongoDB ?

MongoDB est un système de gestion de base de données (SGBD) orientée documents, répartitionnable sur un nombre quelconque d'ordinateurs et ne nécessitant pas de schéma prédéfini des données.

À choisir pour : tout type de besoin de stockage de documents, c'est-à-dire de données structurées : sérialisation d'objets, pages web, formulaires de saisie, informations d'applications. La représentation en documents permet d'utiliser MongoDB pour des structures proches de ce qui est stocké dans une base relationnelle, ou des structures hiérarchiques, avec une plus grande souplesse dans le schéma de données. Un système d'indexation secondaire permet d'effectuer des recherches dans les données et de récupérer des documents par d'autres points d'entrée que la clé.

Types de données manipulées : BSON, un format JSON « binarisé » pour être plus compact.

Maintien de la cohérence : cohérence finale à travers les réplicas. Les écritures ont une option (write concern) pour indiquer qu'elles sont validées si elles ont écrit sur un certain nombre de réplicas ou sur la majorité des replicats. Une mise à jour est atomique par document, il n'y a pas de transaction multidocument possible.

Mode de distribution : centralisé. sharding (partitionnement des données) et réplication automatique même à travers les data centers géographiquement séparés. Les métadonnées des shards sont centralisées sur un serveur de configuration, qui peut lui-même être répliqué pour fournir de la redondance. La connexion client se fait aussi sur un serveur maître, qui redirige la requête vers le bon serveur de shard. Ce serveur maître peut lui aussi être redondant.

Points forts : moteur solide, bonnes performances. Simplicité d'utilisation du point de vue du développement client et bonne intégration des pilotes dans les langages. Support commercial par la société qui développe le produit, MongoDB Inc. Ses capacités de sharding et de réplication automatiques permettent de monter en charge horizontalement au fur et à mesure de l'augmentation des besoins.

2 Quand aller vers Cassandra ?

Apache Cassandra est un système de gestion de base de données (SGBD) de type NoSQL conçu pour gérer des quantités massives de données sur un grand nombre de serveurs, assurant une haute disponibilité en éliminant les points individuels de défaillance (SPOF). Il permet une répartition robuste sur plusieurs centres de données, avec une réplique asynchrone sans master et une faible latence pour les opérations de tous les clients.

À choisir pour : choix intéressant pour de grands volumes de données et les besoins de bases distribuées, hautement disponibles et décentralisées, sur de multiples data centers même géographiquement séparés.

Types de données manipulées : base de données orientée colonnes. Données organisées dans un keystore (l'équivalent d'une base de données) par famille de colonnes (l'équivalent d'une table) selon une clé et des colonnes. La cellule est composée d'un nom de colonne, de la valeur et d'un timestamp. Les familles de colonnes doivent être prédéfinies, mais les colonnes dans une famille ne font pas l'objet d'un schéma défini. Cassandra a une clé par famille de colonnes, ce qui le rapproche plus d'un modèle de base de données relationnelle.

Maintien de la cohérence : la cohérence en écriture et en lecture sont paramétrables. Par exemple, on peut forcer une écriture, ou une lecture, à être validée sur plusieurs répliques avant d'être considérée comme effectuée. Mode de distribution : décentralisé. Chaque nœud est indépendant et il n'y a pas besoin de serveur maître. Les connexions utilisateurs peuvent se faire sur n'importe quel nœud, qui se chargera de rediriger le client vers le nœud qui contient les données souhaitées. La répartition est réalisée par vnodes ou, historiquement, par hachage consistant.

Développé en : Java. Licence et support : Apache 2.0. Cassandra est un projet de la fondation Apache. La société américaine DataStax (<http://www.datastax.com/>) maintient une édition de Cassandra, nommée DataStax Enterprise, gratuite au téléchargement avec une option d'abonnement avec support.

Points forts : implémentation solide, système décentralisé, souplesse de configuration, cohérence paramétrable. Cassandra est un produit mature, largement utilisé et très populaire, c'est une excellente solution pour bâtir un système de gestion de données volumineux et décentralisé. Simplicité d'utilisation du point de vue du développement client et bonne intégration des pilotes dans les langages.

3 Limitations de Cassandra et de MongoDB

Cassandra et MongoDB ne permettent pas les jointures. Ainsi, les données doivent être dénormalisées de sorte que chaque format de stockage soit orienté par une requête. On peut éventuellement utiliser Apache Spark pour la réalisation des jointures. Par rapport au théorème CAP, Cassandra et MongoDB sont tous les deux tolérants aux partitionnement

dû à la perte du réseau ou d'un nœud du cluster. Cassandra est hautement disponible et éventuellement cohérent (eventually consistent) tandis que MongoDB est fortement cohérent et fondamentalement disponible (basically available). La cohérence éventuelle signifie que le système deviendra automatiquement cohérent après un certain temps une fois qu'il cesse de recevoir des nouvelles données en entrées. La disponibilité fondamentale signifie que le système deviendra automatiquement disponible après un certain temps. La disponibilité est la capacité du système à répondre aux requêtes tandis que la cohérence est la capacité du système à retourner les données les plus récentes.

Conclusion

Les moteurs NoSQL présentent certains avantages : souplesse du schéma, typage dynamique, proximité avec les langages clients, montée en charge facilitée par une approche horizontale avec sharding automatique, etc., ce qui permet une gestion plus aisée du système d'information. Mais cela impose aussi des contraintes. En effet, la relative souplesse du schéma et la proximité avec le langage client orientent les moteurs NoSQL vers une organisation des données plutôt par application que vers une centralisation et une utilisation partagée. Le choix d'un moteur de base de données NoSQL se fait donc à partir des besoins. Autant que possible, ce choix doit aussi se baser sur des tests réalisés en interne pour vous assurer que le moteur réponde à vos besoins, et qu'il est capable de tenir la charge solidement, comme vous le souhaitez. Dans le monde NoSQL, un moteur correspond souvent à une application cliente, vous pouvez donc tout à fait mettre en place plusieurs solutions de gestion de données selon vos besoins. Sources et approfondissements (voir [4, 7, 3, 6, 2, 1, 5])

Références

- [1] Bruchez, R. : Les bases de données NoSQL et le Big Data : Comprendre et mettre en oeuvre. Eyrolles (2015)
- [2] Cassandra : Cassandra limitations. Tech. rep. (2018). URL <https://wiki.apache.org/cassandra/CassandraLimitations>
- [3] Cassandra : Cassandra website. Tech. rep. (2018). URL <http://cassandra.apache.org/>
- [4] DataStax : Why migration from mysql to cassandra? Tech. rep. (2012). URL <https://www.datastax.com/wp-content/uploads/2012/08/WP-DataStax-MySQLtoCassandra.pdf?1>
- [5] Harrison, G. : Next Generation Databases : NoSQL, NewSQL, and Big Data : What every professional needs to know about the future of databases in a world of NoSQL and Big Data. Apress (2016)
- [6] MongoDB : MongoDB limitations. Tech. rep. (2018). URL <https://docs.mongodb.com/manual/reference/limits/>
- [7] MongoDB : MongoDB website. Tech. rep. (2018). URL <https://www.mongodb.com/>