L'informatique des entrepôts de données

Daniel Lemire

SEMAINE 14
NoSQL

14.1. Présentation de la semaine

On construit souvent les entrepôts de données en utilisant des systèmes de bases de données relationnels conventionnelles. Il existe cependant de nombreuses alternatives. On désigne souvent l'ensemble de ces alternatives par le terme NoSQL [**3**]. Cette semaine, nous ferons un survol de ces alternatives.

14.2. Une grande diversité

Dans une certaine mesure, les systèmes tels que MySQL, VoltDB, SQL Server ou Oracle sont interchangeables. Certains systèmes sont plus puissants et permettent plus aisément de traiter des bases de données volumineuses. D'autres, comme MySQL, sont peu coûteux. Mais ils reposent tous sur les mêmes principes mis de l'avant par Codd [**2**] dans les années 1970. Tout d'abord, ils permettent de construire des bases de données relationnelles. Les données prennent donc nécessairement la forme de rangées dans des colonnes. De plus, elles tentent le plus possible de satisfaire aux contraintes ACID qui permettent d'assurer l'intégrité des données :

A lire : `http://fr.wikipedia.org/wiki/Codd`

A lire : `http://fr.wikipedia.org/wiki/Proprietes_ACID`

En pratique, il n'est pas toujours nécessaire ou souhaitable de stocker les données en utilisant un modèle relationnel. Dans certains cas, par

exemple, un modèle en graphe peut être plus approprié. Et les contraintes ACID ne sont pas toujours nécessaires. Par exemple, il n'est pas toujours essentiel, que lorsque de nouvelles informations ont été saisies dans le système, toutes les requêtes subséquentes en tiennent compte. Dans la vie courante, les systèmes informatiques sont plutôt éventuellement cohérents plutôt que strictement cohérents. Lors d'un achat avec une carte de crédit, par exemple, il peut arriver que vous soyez, temporairement, facturé deux fois, ou pas du tout. L'important est que le système puisse corriger l'anomalie suffisamment rapidement. En somme, on peut donc utiliser d'autres types de systèmes. Il en existe une grande variété, et le bon choix dépend de vos données et de l'utilisation que vous voulez en faire.

A lire : `http://fr.wikipedia.org/wiki/NoSQL`

14.3. Bases de données orientées document

Dans une base de données orientées document, on remplace les rangées de la base de données relationnelle par un document. Un document n'est qu'un objet ayant un nombre variable d'attributs. On peut penser par exemple à un système stockant des courriels. Chaque courriel peut avoir plusieurs attributs, dont le destinataire ou l'objet. Certains attributs peuvent n'apparaître que dans certains documents.

A lire : `http://fr.wikipedia.org/wiki/CouchDB`

A consulter : `http://en.wikipedia.org/wiki/MongoDB`

Les utilisateurs d'AWS (Amazon Web Services) voudront sans doute prendre en considération DocumentDB qui offre une interface compatible avec MongoDB. Elle permet de stocker, de gérer et de requêter des données semi-structurées, telles que les documents JSON, avec une faible latence.

14.4. Bases de données en graphe

Dans les bases de données en graphe, on remplace l'unité de base des bases de données relationnelles (le tuple ou rangée) par le nœud dans un graphe. Si vos données correspondent, par exemple, à un réseau social ou à un ensemble de pages web, une base de données en graphe peut être mieux adaptée à vos besoins.

A lire : `http://fr.wikipedia.org/wiki/Theorie_des_graphes`

A consulter : `http://en.wikipedia.org/wiki/Neo4j`

14.5. Bases de données orientées objet

Les bases de données orientées objet ne sont pas une invention récente, elles sont nées en même temps que l'adoption des langages de programmation orienté-objet. En somme, l'idée est relativement simple : il s'agit de stocker non pas des relations, mais des objets (au sein de la programmation orientée-objet).

A lire : `http://fr.wikipedia.org/wiki/Base_de_donnees_orientee_objet`

14.6. Bases de données de type clé-valeur

Il arrive parfois que l'on n'ait besoin que de retrouver des valeurs étant donné une clé. Par exemple, si votre application a besoin de retrouver le dossier d'un étudiant en n'utilisant que son code étudiant, alors une base de données de type clé-valeur sera idéale. Elles tendent à être plus rapides et plus simples que les bases de données relationnelles. On les utilise, notamment, pour stocker des pages web précalculées afin d'accélérer certains sites web. Dans un tel cas, la clé peut être, tout simplement, l'hyperlien de la page.

A consulter : `http://www.membase.org/`

A consulter : `http://fallabs.com/tokyocabinet/`

A consulter : `http://code.google.com/p/redis/`

14.7. Bases de données orientées colonne

Pour les très grandes tables devant être distribuées sur un réseau de machine, le modèle proposé par Google par sa base de données BigTable [**1**] a été adopté par plusieurs autres sociétés. Au lieu d'être orienté par rangées, comme la plupart des bases de données relationnelles, BigTable est orienté colonne, ce qui signifie que les données d'une même colonne sont stockées de manière consécutive. Par ailleurs, dans ce modèle, la compression des données joue un rôle important afin d'accélérer le traitement.

A lire : `http://fr.wikipedia.org/wiki/HBase`

A lire : `http://fr.wikipedia.org/wiki/BigTable`

Le système le plus populaire de ce type est sans doute DuckDB [**4**].

DuckDB est un système de gestion de base de données SQL analytique, orienté colonne, qui fonctionne en mode intégré (embedded). Cela signifie qu'il peut être exécuté directement dans l'application qui l'utilise, sans nécessiter un serveur séparé. DuckDB est conçu pour offrir des performances élevées sur des requêtes analytiques complexes

avec de grandes quantités de données, tout en conservant une empreinte mémoire faible. Il est particulièrement adapté pour les environnements où l'analyse de données doit être effectuée rapidement et efficacement, comme dans les outils de science des données, les notebooks Jupyter, ou encore dans des scripts Python ou R. Il supporte une grande partie du standard SQL, ce qui le rend accessible et facile à utiliser pour ceux qui ont déjà une expérience avec d'autres bases de données SQL.

A consulter : `https://duckdb.org`

14.8. Conclusion

Il est impossible de rendre justice à l'ensemble des alternatives aux systèmes de bases de données traditionnelles. C'est un domaine en constante mutation. Par contre, il faut retenir que les bases de données traditionnelles ne sont pas la seule possibilité au sein d'un entrepôt de données.

BIBLIOGRAPHIE

1. F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber. Bigtable: A distributed storage system for structured data. *ACM Trans. Comput. Syst.*, 26(2):1–26, 2008.
2. E. F. Codd. A relational model of data for large shared data banks. *Commun. ACM*, 13(6):377–387, 1970.
3. W. Khan, T. Kumar, C. Zhang, K. Raj, A. M. Roy, and B. Luo. Sql and nosql database software architecture performance analysis and assessments—a systematic literature review. *Big Data and Cognitive Computing*, 7(2):97, 2023.
4. M. Raasveldt and H. Mühleisen. Duckdb: an embeddable analytical database. In *Proceedings of the 2019 International Conference on Management of Data*, pages 1981–1984, 2019.