L'informatique des entrepôts de données

Daniel Lemire

SEMAINE 7
Bases de données réparties

7.1. Présentation de la semaine

Nous avons vu comment les ingénieurs peuvent accélérer les requêtes dans les entrepôts de données en utilisant les index, la matérialisation, et la compression. Ils peuvent mettre à profit le grand volume de mémoire vive de nos ordinateurs, ou les microprocesseurs toujours plus puissants. Cependant, quand l'entrepôts de données est suffisamment volumineux, ou distribué, il est essentiel de faire appel à l'informatique répartie. Par exemple, une partie de nos données peuvent résider sur un serveur en Californie alors que le reste sera à Paris. Ou bien alors on fera appel à une salle de serveurs complète pour stocker les données et répondre aux multiples requêtes.

7.2. Quelques notions de base

Les ingénieurs font souvent référence à la *mise à l'échelle* (scalability) comme étant la capacité d'un système à pouvoir évoluer en fonction des besoins et des ressources disponibles. Ainsi, certaines solutions simples peuvent paraître intéressantes lorsque les besoins sont modestes. Par contre, que se passera-t-il si le nombre d'utilisateurs passe de 10 à 5000 utilisateusr ou si le volume de données pass de 4 Go à 12 To? Cette capacité de mise à l'échelle porte non seulement sur l'infrastructure matérielle, mais aussi les infrastructures logicielles et humaines. Elle se mesure à la performance, mais aussi à la qualité. Par exemple, s'il est nécessaire d'exécuter à la main certaines opérations critiques et difficiles, l'infrastructure peut être incapable de répondre à la demande

parce qu'il est impossible de trouver de la main-d'œuvre qualifiée. Sur le plan matériel, on peut faire une mise à l'échelle verticale (*scaling in*) en augmentant la puissance de nos serveurs dédiés (notamment en ajoutant de la mémoire et de l'espace disque). Cette approche est simple, mais elle a ses limites. On peut aussi faire une mise à l'échelle horizontale (*scaling out*) en ajoutant des serveurs.

La *disponibilité* d'une infrastructure est un autre concept important. Il s'agit de la capacité d'un système à continuer d'offrir un service même en cas de panne. Par exemple, si votre architecture dépend de douze serveurs, que se passe-t-il si l'un d'entre eux tombe en panne? Est-ce que vous avez une copie des données que contenait ce serveur? Est-ce que votre logiciel est assez bien conçu pour pouvoir automatiquement compenser les pannes et les bogues? Qu'est-ce qui se passe si un pare-feu rend l'âme? Est-ce que tout le système cesse de fonctionner? À la rigueur, que se passerait-il en cas d'attaque terroriste?

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

7.3. Cloud computing

La convention jusqu'à récemment était que les entrepôts résidaient sur des serveurs dédiés au sein d'une entreprise. Cette approche permet de contrôler finement la puissance de calcul requise. Par contre, la fiabilité des systèmes est souvent déficiente. Par exemple, est-ce qu'il est facile d'avoir du personnel compétent à toute heure du jour pour administrer l'entrepôt? L'informatique dans les nuages est une solution de remplacement peu coûteuse et simple afin de s'assurer de la disponibilité et de la mise à l'échelle matérielle d'une infrastructure. Elle est une forme d'impartition portant à la fois sur le matériel est la main-d'œuvre.

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

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

7.4. Bases de données parallèles et réparties

La plupart des systèmes de bases de données bénéficient directement de l'ajout de microprocesseurs ou de cœurs. Cela se fait plus ou moins automatiquement. Ainsi, une même machine qui a plusieurs processeurs répartira automatiquement la charge (les requêtes) sur les microprocesseurs ou les cœurs disponibles. Comme les données sont souvent statiques dans les entrepôts de données, il est facile de paralléliser le traitement des requêtes. Cela est d'autant plus vrai si on utilise le partitionnement des données. Par contre, il n'est pas toujours facile

d'ajouter des serveurs. Il existe plusieurs stratégies dont le partage complet, le partage des disques, et l'absence de partage. La différence entre les différents modèles est parfois une affaire de marketing : il n'est pas toujours facile de catégoriser les produits commerciaux.

7.4.1. Modèles avec partage complet (*share everything*)

L'approche utilisée notamment par Oracle est un modèle avec partage complet. En gros, le système fonctionne comme si on avait une seule machine (virtuelle). À chaque fois qu'on ajoute un nouveau serveur, cette machine virtuelle devient plus puissante. Les serveurs ne sont pas autonomes, mais plutôt fédérés. Dans des conditions idéales, ces systèmes sont très performants. Par contre, la mise à l'échelle peut être limitée, car les différents serveurs doivent communiquer finement. De plus, la disponibilité peut être un problème en cas de panne, parce que les serveurs dépendent parfois beaucoup les uns des autres.

7.4.2. Modèles avec partage de disque (*shared disk*)

Dans un modèle avec partage de disque, les données se trouvent sur un réseau de stockage partagé. Lorsqu'une nouvelle requête arrive dans le système, un serveur est désigné pour y répondre. Ce serveur peut faire appel aux unités de stockage partagées, mais il ne peut faire appel aux autres serveurs. Évidemment, dans ce modèle, les unités de stockage peuvent créer un goulot d'étranglement. Par contre, il y a une excellent fiabilité: un serveur peut tomber en panne sans causer une panne visible. Ce type de système a aussi une bonne mise à l'échelle. En effet, les données qui sont souvent utilisées peuvent être répliquées sur plusieurs disques. Par contre, il y une certaine limite à la mise à l'échelle à cause de la saturation de la bande passante.

A consulter : `https://www.couchbase.com/resources/concepts/database-clustering`

7.4.3. Modèles sans partage (*share nothing*)

Dans un modèle sans partage, chaque sereur dispose de sa propre unité de stockage. Chaque serveur est autonome. Les requêtes sont acheminées sur les différents serveurs qui doivent parfois collaborer. Presque toujours, les données sont partitionnées, soit horizontalement, soit verticalement. Comme les modèles avec partage de disque, la fiabilité est excellente: une panne de serveur est sans conséquence visible pour l'ensemble du système. La mise à l'échelle est excellente. Par contre,

ces systèmes sont plus complexes et moins efficaces. Il y a une évolution vers les modèles avec absence de partage dans les gros entrepôts à cause de leur capacité exceptionnelle de mise à l'échelle.

A consulter : `http://en.wikipedia.org/wiki/IBM_DB2`

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

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

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

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

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

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

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

7.5. Questions d'approfondissement

(a) Selon vous, est-ce que l'architecture des grandes sociétés web comme Amazon ou Google est avec partage complet, partage de disque ou sans partage?
(b) Si vous disposez de PC générique avec lesquels vous voulez construire un entrepôt de données, quel type de modèle est le plus approprié?
(c) Vous souhaitez minimiser la consommation électrique de votre installation, quel type de modèle est plus approprié?
(d) Vous ne souhaitez pas partitionner vos données, quels modèles devez-vous considérer?

7.6. Réponses suggérées

(a) Sans partage.
(b) Sans partage.
(c) Avec partage complet.
(d) Avec partage complet ou partage de disque.