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?

La haute disponibilité (HA, *High Availability*) désigne un ensemble de techniques et de conceptions informatiques visant à garantir qu'un système, une application ou un service reste accessible et opérationnel presque en permanence, minimisant les temps d'arrêt, même en cas de panne matérielle, logicielle ou réseau. Elle repose sur des mécanismes comme la redondance (serveurs multiples, bases de données répliquées), la tolérance aux pannes (basculement automatique via des *clusters*), et la surveillance continue pour détecter et résoudre rapidement les problèmes. Typiquement, la haute disponibilité vise un taux de disponibilité proche de 99,999 % (les "cinq neuf"), ce qui équivaut à moins de 5 minutes d'arrêt par an. Elle est cruciale pour les systèmes critiques, comme les services bancaires en ligne, les plateformes de commerce électronique ou les infrastructures *cloud*, où toute interruption peut avoir des conséquences financières ou opérationnelles importantes.

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-il 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 et la main-d'œuvre.

Le *cloud computing* désigne la fourniture de ressources informatiques à la demande, accessibles via Internet, sans gestion directe de l'infrastructure physique par l'utilisateur. Ces ressources incluent des serveurs, du stockage, des bases de données, des logiciels et des services d'intelligence artificielle, hébergés dans des centres de données gérés par des fournisseurs comme Amazon Web Services (AWS), Microsoft Azure ou Google Cloud Platform.

- **Modèles de service** :
  - *IaaS* (*Infrastructure as a Service*) : Fournit des infrastructures comme des machines virtuelles (ex. : AWS EC2).
  - *PaaS* (*Platform as a Service*) : Offre des environnements de développement (ex. : Google App Engine).
  - *SaaS* (*Software as a Service*) : Propose des logiciels accessibles en ligne (ex. : Google Workspace).
- **Modèles de déploiement** : *Public* (partagé), *Privé* (dédié à une organisation), *Hybride* (combinaison des deux).
- **Avantages** : Évolutivité, réduction des coûts d'investissement initial, flexibilité, et accès à des technologies avancées.
- **Défis** : Sécurité des données, dépendance au fournisseur, et coûts récurrents à long terme.

Exemple: Une startup utilise AWS pour héberger son application web (*IaaS*), stocker ses données clients sur Amazon S3, et analyser les performances avec un service *SaaS* comme Google Analytics. Cela lui permet de scaler rapidement sans investir dans des serveurs physiques.

L'externalisation consiste à confier des fonctions, des processus ou des services à un prestataire externe spécialisé, souvent pour réduire les coûts, accéder à une expertise spécifique, ou améliorer l'efficacité. En informatique, cela peut inclure la gestion des centres de données, le support technique, le développement logiciel, ou même l'utilisation de services *cloud*.

- **Types d'externalisation** :
  - *IT Outsourcing* : Délégation de services informatiques, comme la maintenance des systèmes.
  - *Business Process Outsourcing (BPO)* : Externalisation de processus métiers, comme la gestion de la paie.
  - *Offshoring* : Externalisation vers un autre pays pour réduire les coûts.
- **Avantages** : Réduction des coûts opérationnels, accès à des compétences spécialisées, et concentration sur les activités stratégiques.
- **Défis** : Perte de contrôle, risques liés à la qualité, et problèmes de communication avec les prestataires.

Exemple: Une entreprise externalise son support client à un centre d'appels en Inde (*BPO*) et confie la gestion de son infrastructure informatique à un fournisseur local qui utilise des services *cloud* pour héberger ses données. Cela réduit les coûts tout en maintenant un service de qualité.

Le *cloud computing* est souvent une forme d'externalisation, car les entreprises délèguent la gestion de leurs ressources informatiques à des fournisseurs de services *cloud*. Par exemple, utiliser Microsoft Azure pour héberger une application est à la fois du *cloud computing* (accès à des serveurs virtuels) et de l'externalisation (le fournisseur gère l'infrastructure). Cependant, l'externalisation est plus large, englobant des services non liés au *cloud*, comme l'externalisation du développement logiciel à une entreprise tierce. Les deux approches partagent l'objectif de réduire les coûts et d'améliorer l'efficacité, mais le *cloud* se distingue par son modèle de livraison à la demande et sa scalabilité.

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 : `http://www.sybase.com/manage/shared-disk-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.

Dans une architecture sans partage, chaque nœud fonctionne de manière autonome, avec son propre processeur, mémoire et disque. Les données sont partitionnées (par hachage, plage, ou autre) entre les nœuds, et les requêtes sont exécutées en parallèle, souvent via un traitement massivement parallèle (MPP, *Massively Parallel Processing*). Cette approche réduit les contentions et permet d'ajouter des nœuds pour augmenter la capacité, rendant ces systèmes idéaux pour les applications à forte charge analytique, comme l'analyse de données ou les systèmes de *business intelligence*. Cependant, la coordination entre nœuds peut introduire une complexité, notamment pour les transactions distribuées."

Il y a plusieurs examples dans l'industrie. IBM DB2, dans sa version avec la fonctionnalité *Database Partitioning Feature* (DPF), adopte une architecture sans partage pour les entrepôts de données. Les données sont partitionnées sur plusieurs nœuds, et chaque nœud traite ses partitions indépendamment, avec un coordinateur gérant les

requêtes globales. DB2 BLU, une extension colonnaire introduite en 2013, améliore les performances analytiques. DB2 est apprécié pour sa robustesse dans les environnements financiers et industriels, mais sa complexité et le besoin d'expertise pour l'optimisation peuvent limiter son adoption face à des systèmes plus spécialisés comme Teradata." Vertica, acquis par HP (puis Micro Focus), est une base de données analytique colonnaire sans partage, conçue pour les charges de travail à faible latence, comme chez Facebook ou Uber. Elle utilise une architecture MPP où les données sont segmentées sur les nœuds, avec une compression avancée pour réduire les coûts de stockage. Vertica excelle dans les requêtes analytiques complexes et offre une intégration avec des outils comme Hadoop ou Spark, mais son coût total de possession (TCO) peut être élevé en raison de cycles d'implémentation complexes." MySQL Cluster (ou NDB Cluster) est une version de MySQL conçue pour la haute disponibilité et la scalabilité, utilisant une architecture sans partage. Les données sont partitionnées sur des nœuds de données, avec une réplication synchrone pour assurer la cohérence. Contrairement aux autres systèmes ici, MySQL Cluster est optimisé pour les charges transactionnelles (OLTP) plutôt que purement analytiques, ce qui le rend adapté aux applications web nécessitant des lectures/écritures rapides, comme les télécommunications. Cependant, sa gestion des requêtes complexes est moins performante que celle des bases analytiques comme Vertica." Paraccel (aujourd'hui intégré à Actian Matrix) est une base de données analytique colonnaire sans partage, qui a servi de base à Amazon Redshift. Utilisant une architecture MPP basée sur PostgreSQL, elle répartit les données et les traitements sur des nœuds pour des performances élevées dans les entrepôts de données. Paraccel était compétitif dans les années 2000, mais son impact a été amplifié par Redshift, qui a démocratisé son architecture. Paraccel reste pertinent pour les déploiements sur site, mais il est éclipsé par des solutions cloud modernes." Greenplum, acquis par EMC (puis Pivotal, VMware, et Broadcom), est une base de données MPP sans partage basée sur PostgreSQL, optimisée pour les charges analytiques et les entrepôts de données. Elle répartit les données sur des nœuds avec une exécution parallèle des requêtes, et supporte des fonctionnalités OLTP depuis sa version 6 (2019). Greenplum est apprécié pour son intégration avec Hadoop et son open-source partiel, mais son passage à une licence fermée en 2024 par Broadcom a suscité des critiques. Elle reste compétitive pour les analyses à grande échelle." Netezza, acquis par IBM en 2010, est un appareil d'entrepôt de données (*data warehouse appliance*) sans partage, utilisant une architecture MPP avec des FPGA pour accélérer les traitements. Conçu pour les charges analytiques, il

excelle dans les environnements nécessitant des requêtes simples sur de grands volumes, mais sa technologie non colonnaire (basée sur des *zone maps*) limite sa compétitivité face à Vertica ou Greenplum. Netezza reste populaire dans les environnements IBM, bien que son adoption ralentisse face aux solutions cloud." Teradata est un pionnier des entrepôts de données MPP sans partage, incorporé en 1979. Il répartit les données sur des nœuds avec une exécution parallèle, offrant une gestion flexible des charges de travail et une intégration avancée des analyses. Teradata domine le marché des grandes entreprises grâce à sa maturité et ses performances, mais son coût élevé et la concurrence des solutions cloud comme Redshift ou Snowflake challengent sa position. En 2016, Teradata a introduit des modèles à bas prix pour rester compétitif." Oracle Real Application Clusters (RAC) adopte une architecture à disque partagé (*shared-disk*), où plusieurs nœuds accèdent à un stockage commun, contrairement aux modèles sans partage. Bien que scalable, RAC peut souffrir de goulots d'étranglement dus à la coordination du disque partagé, surtout pour les charges analytiques lourdes. Oracle Exadata, une solution matérielle optimisée, améliore les performances, mais RAC reste moins adapté aux entrepôts de données que les systèmes sans partage comme Teradata ou Vertica, qui évitent les contentions."

- **IBM DB2** : Robuste pour les environnements mixtes (OLTP/-analytique), mais complexe à optimiser.
- **Vertica** : Excellent pour les analyses à faible latence, avec une forte compression, mais coûteux.
- **MySQL Cluster** : Idéal pour les transactions à haute disponibilité, moins performant pour l'analytique.
- **Paraccel** : Base de Redshift, performant mais dépassé par les solutions cloud.
- **Greenplum** : Flexible avec intégration Hadoop, impacté par la fermeture de la licence.
- **Netezza** : Efficace pour les requêtes simples, limité par l'absence de stockage colonnaire.
- **Teradata** : Leader pour les grandes entreprises, mais coûteux face aux alternatives cloud.
- **Oracle RAC** : Architecture à disque partagé, moins adaptée aux charges analytiques lourdes.

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.