

# Présentation de la réplication des données

Un environnement de données distribuées comprend plusieurs copies d'informations identiques sur des serveurs différents. Cela permet de copier et de distribuer des données et des objets de base de données d'une base de données vers une autre, puis de synchroniser ces bases de données afin de préserver leur cohérence. Les données sont distribuées à différents emplacements, à des utilisateurs distants ou mobiles sur des réseaux locaux et étendus, reliés par des connexions d'accès à distance, des connexions sans fil ou Internet.

Deux stratégies principales existent et se différencient par l'autonomie des serveurs, le degré de latence dans la synchronisation des réplicas, la cohérence des bases de données et la gestion des conflits de mise à jour.

## Les stratégies possibles :

- les transactions distribuées ou le temps de latence entre les mises à jour est pratiquement nul pour des sites connectés en permanence (pas d'autonomie) et une cohérence transactionnelle complète et immédiate. Cela élimine les conflits de mise à jour. De cette manière, les sites distants disposent des mêmes données au même moment en utilisant un protocole de validation à deux phases pour garantir l'exécution simultanée d'une même transaction sur tous les sites concernés.
- la réplication ou le temps de latence peut être de plusieurs jours, l'autonomie très importante avec une déconnexion des sites, sans forcément de cohérence transactionnelle et avec une gestion des conflits de mise à jour plus difficile. C'est de cela qu'il s'agit quand on parle de réplication de données.

## La gestion des données selon le type de réplication :

- la **réplication transactionnelle** ; seules les données modifiées sont distribuées et les données ne sont modifiées qu'à un endroit. La réplication transactionnelle est généralement utilisée dans des **scénarios serveur à serveur** qui nécessitent un débit élevé, pour une haute disponibilité, les entrepôts de données, l'intégration de données depuis plusieurs sites.
- La **réplication de fusion** ; plusieurs sites apportent des modifications aux données de manière indépendante et les données sont fusionnées régulièrement au niveau du serveur source. Ce type de réplication est conçu essentiellement pour les applications mobiles ou les applications de serveur distribuées. Cela est utilisé dans des **scénarios serveur à client**. Des conflits peuvent survenir et être résolus mais la cohérence transactionnelle n'est pas garantie.
- la **réplication de capture instantanée** ; une image de l'ensemble des données (modifiées ou non) sur un serveur source remplace les données se trouvant sur un serveur cible, régulièrement ou à la demande. Cela est utilisée pour fournir le jeu de données initiales pour la réplication transactionnelle et de fusion ; elle peut s'utiliser également lorsque des actualisations complètes

## Dans le processus de réplication :

- un serveur de base de données éditeur gère la base de données source publiée, pré-pare et envoie au distributeur les modifications apportées aux données publiées. Il est aussi appelé serveur maître.
- un serveur de base de données abonné conserve une copie des données répliquées et reçoit les mises à jour. Il est appelé serveur esclave.

Parfois, ce n'est pas l'ensemble de la base de données qui est répliquée mais seulement une partie des données. C'est le cas quand l'organisation estime que seules les données nécessaires au site distant doivent être répliquées.

On parle de publication pour définir un ensemble d'articles et cela constitue la base d'un abonnement. Il peut y avoir plusieurs publications dans une base mais une publication ne peut concerner plusieurs bases.

L'**article** est l'unité de base de la réplication (une table ou partie de table, etc.).

## La cohérence des données répliquées

Il existe plusieurs types de réplication car les besoins sont différents. La réplication se présente alors comme différentes technologies de réplication à adapter et à combiner pour répondre le plus exactement possible aux besoins des applications. Chaque technologie présente ses avantages et ses inconvénients et voici les trois critères principaux pour sélectionner une technologie de réplication :

- la cohérence des données répliquées,
- l'autonomie des sites,
- le partitionnement des données pour éviter les conflits.

Il n'est pas possible d'optimiser les trois critères en même temps. Ainsi une solution qui favorisera la cohérence des données fournira une faible autonomie aux sites pour connaître à tout instant l'ensemble des modifications qui ont lieu sur les données.

Il existe deux principaux types de cohérence :

- l'homogénéité des transactions en prenant en compte les critères ACID (Atomicité, Cohérence, Isolement et Durabilité)
- la convergence des données afin que tous les sites tendent à posséder le même jeu de données.

## La cohérence transactionnelle

Tous les sites qui participent à la réplication ont la garantie d'avoir les mêmes données à jour au même moment. Cela est possible avec un protocole de validation à deux phases avec tous les sites participants. De cette manière, les modifications sont effectuées sur tous les sites ou sur aucun, ce qui impose des connexions réseaux fiables. En effet une validation de transaction ne peut se faire tant que le serveur n'est pas reconnecté au réseau.

Il peut y avoir cependant un temps de latence, c'est à dire qu'un délai peut s'écouler entre l'instant où la transaction est jouée sur le serveur de publication et l'instant où les modifications sont réfléchies sur les autres sites.

## La convergence des données

Tous les sites finissent par obtenir le même jeu de données. Tout d'abord, les données des différents sites évoluent librement et indépendamment les uns des autres. Puis la convergence des données est mise en place à l'aide de la réplication de fusion qui tend à amener tous les sites qui y participent vers la gestion du même jeu de données commun.

## L'autonomie des sites

Cela se mesure par l'impact d'une opération effectuée sur un site sur les autres sites. Quand un site peut évoluer d'une manière totalement libre, l'autonomie est parfaite. Ce site autonome ne se soucie pas des opérations qui peuvent intervenir sur les autres sites. Lorsque la réplication garantit la convergence des données, l'autonomie des sites est à son maximum, car chaque serveur de base de données peut évoluer librement par rapport aux autres sites. À l'inverse, la cohérence transactionnelle immédiate impose aucune autonomie des sites car une transaction doit être approuvée par tout le monde avant d'être validée. Si un seul serveur ne peut pas le faire alors la transaction n'est validée nulle part.

## Le partitionnement des données

Cela consiste à répartir les données sur plusieurs sites afin que chaque site travaille avec son propre jeu de données, c'est-à-dire les données dont il a besoin et qui sont rigoureusement distinctes des autres. Les transactions d'un site ne mettent en jeu que les données du site et la cohérence globale est conservée.

Par exemple, chaque agence d'une organisation possède un fichier client sur une zone géographique bien déterminée et un client ne peut alors être géré que par une seule agence. Toute source de conflit est alors exclue.

Cela est possible en partitionnant les données. **Exemple :** Quand des billets de spectacles doivent être vendus en plusieurs points de vente, il faut s'assurer que la même place ne soit pas vendue deux fois. La solution la plus simple consiste à attribuer à chaque point de vente un certain nombre de places. La salle est donc partitionnée en fonction des points de vente ce qui permet à chacun de ces points de vente de gérer de façon autonome les places qui lui ont été attribuées. Par contre chaque point de vente peut connaître les places non encore vendues par les autres. On peut mettre en œuvre dans ce cas, la cohérence transactionnelle latente pour que l'information sur les places disponibles puisse être diffusée.

**Un schéma pour résumer :**

**Retour au dossier sur la réplication des données ...**

[Fiche Cours \(C6\) : La réplication des données](#)

From:

[/ - Les cours du BTS SIO](#)

Permanent link:

[/doku.php/si7/a10presentation](#)

Last update: **2016/03/11 12:40**

