

Fiche de révisions - répartition et sauvegarde de données

Rédigé par : Jimmy Paquereau

1. Préliminaire

Au sein d'une organisation, il importe que chacun puisse accéder aux informations qui lui sont utiles aux fins d'exercer son activité. Il convient également de pouvoir partager l'information, d'y avoir accès facilement (cf. fiche sur les architectures client-serveur), d'y avoir tout le temps accès et d'y avoir accès vite.

Or, il vient que le système d'information d'une entreprise est communément réparti sur plusieurs sites (exemple : un siège social + des agences), ce qui complique encore un peu plus les choses. Plus encore, les informations sont plus ou moins sensibles, plus ou moins nombreuses, plus ou moins consultées... Et il faut pouvoir répondre aux exigences, à la qualité de service, attendues.

2. Notion de système réparti : distribution et réplication

Un système réparti est un système qui fait intervenir plusieurs machines à la fois, potentiellement réparties à plusieurs endroits, et qui se comporte comme un système unique. Entre autre, on parle de base de données répartie. Une base de données répartie, c'est un ensemble de bases de données qui se comportent comme s'il n'y en avait qu'une.

On se cantonnera au cas des bases de données.

2.1. Première option : la distribution

Qu'est-ce que c'est ?

Une base de données distribuée est une base de données pleinement répartie. Il s'agit d'un système temps réel particulièrement complexe. Dans cette configuration, la base de données répartie est découpée en plusieurs bases de données qui communiquent librement entre elles afin d'assurer l'intégrité des données. Les mises à jour se font en « temps réel* ». Ainsi, le trafic, la communication entre bases, est potentiellement quasi permanente.

* Temps réel : toute la complexité des systèmes temps réel réside dans l'impossibilité physique (mécanique quantique oblige) de faire circuler et traiter l'information de manière instantanée. Et la synchronisation des machines peut s'avérer particulièrement complexe (on ne peut pas se fier à des horloges classiques). Ainsi, les machines ne peuvent ni être réellement synchronisées, ni avoir accès aux mêmes informations au même moment.

Les bases de données constitutives d'une base de données distribuée se partagent les données. Elles ne possèdent qu'une partie des données. On parle de **niveau de redondance**.

Exemple : une base de données distribuée divisée en 3, avec un niveau de redondance égal à 2.

- base de données 1 : donnée clients + données fournisseurs ;
- base de données 2 : données clients + données facturation ;
- base de données 3 : données facturation + données fournisseurs.

Typiquement, une base de données parfaitement distribuée permet d'interroger n'importe laquelle de ses bases pour obtenir ou modifier n'importe quelle donnée. Une autre possibilité est d'avoir une **base de données maîtresse** et des **bases de données filles**. La base maîtresse répartit le trafic. Ce procédé s'appelle le **load balancing**. Dans ces deux cas, l'avantage est qu'on évite le goulot d'étranglement en répartissant le trafic entre plusieurs machines.

A quoi ça sert ? Quels avantages ?

Les bases de données distribuées sont l'apanage des systèmes et logiciels à trafic extrêmement élevé tels que les réseaux sociaux. On dit qu'elles supportent **la montée en charge (la scalabilité)** car elles résistent très bien à la multiplicité des connexions simultanées et aux trafics élevés, là où toutes les autres bases de données échouent.

Un autre avantage des bases de données distribuées est que les données sont continuellement à jour.

MongoDB est un exemple de SGBD (*Système de Gestion de Base de Données*) gérant des bases de données réparties. Ce n'est pas un SGBDR (*Système de Gestion de Base de Données Relationnelles*) et il n'utilise par conséquent pas de SQL. C'est l'un des plus utilisés actuellement.

Quel inconvénient ?

- o la mise en œuvre d'une base de données répartie est souvent complexe ;
- o les bases de données réparties ne sont pas des bases de données relationnelles et ne peuvent donc pas être interrogées en SQL ;
- o la discussion continue des bases de données entre elles occasionne un important trafic ;
- o les systèmes distribués sont des solutions modernes. Ils font encore l'objet des recherches. Et, un inconvénient est que leur utilisation massive est récente.

2.1. Seconde option : la réplication

Qu'est-ce que c'est ?

La réplication consiste à **dupliquer à l'identique** tout ou partie du contenu d'une base de données dans une autre base de données. On parle de **mirroring**.

Dans une telle configuration, on trouve en général :

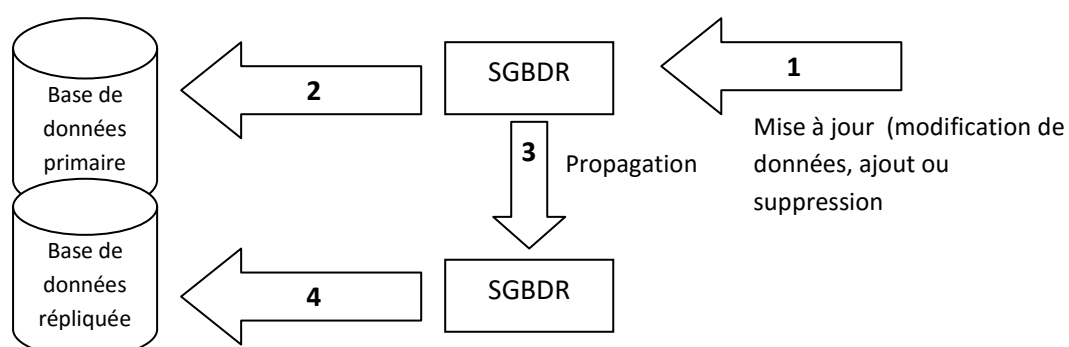
- o une **base de données primaire** centralisant toutes les données. On parle encore de base de données maîtresse ou de base de données de référence ;
- o des **bases de données secondaires** (=base de données filles) contenant une copie complète ou partielle des données de la base de données primaire.

Dans une telle configuration, on parle encore de :

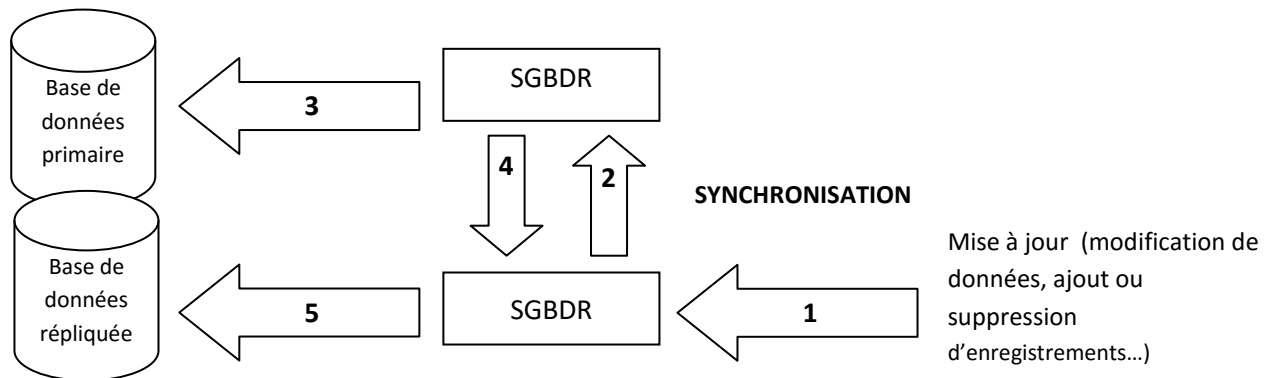
- o **table de référence** : table centralisant les données ;
- o **table répliquée** : table contenant une copie complète ou partielle des données.

Quels modes de réplication ? Selon la manière dont on fait la copie, on parlera de :

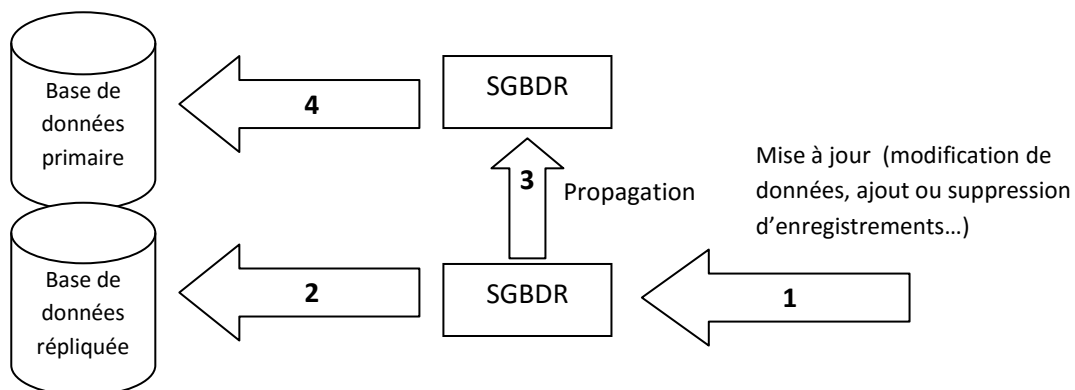
- o **réplication par clicé** : la copie typiquement d'une table (voire d'une base de données entière) est effectuée **de façon périodique**, de la table de référence vers la table répliquée. La table répliquée n'est accessible **qu'en consultation** (=lecture, SELECT), ce qui est l'inconvénient de cette méthode.



o **réplication synchrone** : la copie des données d'une table ait effectuée à chaque opération d'insertion, de modification ou de suppression (INSERT, UPDATE, DELETE). Autrement dit, l'opération est reproduite sur chaque base de données impactée. Et cette opération est **bloquante** pour le client. L'inconvénient réside dans le fait que c'est lent ! Il faut attendre d'avoir effectué l'opération sur chaque base avant de répondre au client. A l'inverse, l'avantage est que les données demeurent à jour. Un tel système est par ailleurs assez facile à mettre en œuvre ;



o **réplication asynchrone** : la copie des données est effectuée **sans bloquer** l'utilisateur. Typiquement, cette copie est effectuée **de façon périodique**. Les données sont périodiquement centralisées, à savoir propagées des tables répliquées vers la table de référence puis réparties (de sorte que chaque table répliquée soit une copie à jour). Il importe alors de définir la **périodicité**. L'avantage de cette méthode réside dans le fait que la copie ne bloque pas le client. L'inconvénient est en revanche le fait que les données ne soient pas tous le temps à jour ;



Quel est l'intérêt de la réplication ?

Dans le cadre d'une organisation possédant plusieurs sites (exemple : siège + agences), l'intérêt réside dans le fait que :

- o le site principal puisse centraliser les données ;
- o les sites secondaires puissent n'avoir accès qu'aux données qui les concernent ;
- o le trafic (sur le réseau) entre agences et siège soit restreint ;
- o l'information soit accessible plus vite puisque chaque site peut accéder aux informations disponibles sur un serveur local au lieu d'accéder à un serveur du siège (distant), ce qui limite la distance et donc le temps de circulation de l'information (les temps d'accès). Qui plus est, cela évite d'interroger continuellement un unique serveur. Cela évite donc un goulot d'étranglement.

2. Notions de continuité de service et de tolérance aux pannes

2.1. Continuité de service

Dans le cas d'un système d'information, le fait d'assurer la **continuité de service** signifie qu'on doit être en mesure d'assurer le minimum de pertes de données en cas de sinistre (dégât des eaux, incendie, chute d'un serveur...) et d'assurer la **reprise d'activité**. A cet égard, on rédige généralement un plan de continuité d'activité (synonyme de plan de continuité de service). Concernant les bases données, les méthodes de **réplication** et de **distribution** évoquées ci-avant peuvent s'inscrire dans ce cadre, sous-réserve pour la réplication d'être totale et de permettre la **restauration des données**, qui plus est de manière rapide (ce qui n'est pas gagné). Ainsi apparaît la notion de **sauvegarde**.

2.2. Tolérance aux pannes

La **tolérance aux pannes** répond à un autre impératif. En informatique, la tolérance aux pannes est la capacité d'un système à resté disponible, éventuellement de manière réduite (on parle de « **mode dégradé** »), en cas de panne, sinistre, erreur... A cet impératif, la réplication de base de données n'est, à elle-seule, pas une solution. En revanche, les systèmes distribués sont biens une solution puisqu'un serveur peut tomber et le système continuer à fonctionner.

2.3. Sauvegarde

La **sauvegarde de données** consiste à pouvoir restaurer les données d'un système à un état antérieur. Cette première définition répond à une problématique de continuité de service. Une sauvegarde de données doit également répondre à des impératifs de tolérance aux pannes (exemple : un disque dur est détruit mais on est encore capable de restaurer les données ; un incendie a eu lieu, mais on a préservé les données).

En ce sens, un CD/DVD et une clef USB ne sont pas des dispositifs adaptés aux sauvegardes de données.

En particulier, il importe de définir une **stratégie de sauvegarde** définissant entre autre la **périodicité des sauvegardes**, le nombre de **versions antérieures conservées**. En effet, il est possible de conserver des versions d'un système ou de données. On parle d'**historisation** ou de **versioning**.

Quelques pistes :

o **systèmes RAID** (RAID 0, RAID 1, RAID 10, RAID 5) : il s'agit des dispositifs de sauvegardes communément employées. On parle de grappes de disques durs (i.e. un ensemble de disque durs) :

- **RAID 0** (*striping*) : les données sont réparties entre les disques durs de la grappe ce qui permet d'écrire sur plusieurs disques en même temps et ainsi diminuer le temps de sauvegarde ;

- **RAID 1** (*mirroring*) : les disques durs de la grappe sont des copies identiques. Si un disque dur est détruit, les données sont encore disponibles sur les autres ;

- **RAID 10** : c'est la combinaison du RAID 0 et du RAID 1 ;

- **RAID 5** : les données sont réparties entre les disques durs, mais, si un disque est détruit, on est capable de reconstituer ses données grâce à ce qu'on appelle des bits de parités ;

o changement de disque à chaud (*hot sphere*) : ce sont des grappes de disques dur où il est possible de changer un disque à chaud, c'est-à-dire, sans redémarrage et donc sans perte de temps ;

o **sauvegardes incrémentales et différentielles** : afin d'optimiser les temps de sauvegardes, les données ou les systèmes sont bien rarement dupliqués, on se contente plutôt de sauvegardées les différences vis-à-vis de la version précédente.