**CHAP 2. LA RÉPLICATION SOUS SQL SERVER 2008**

1. Introduction

Une application de base de données repose sur un modèle client – serveur. Dans ce modèle, le client se connecte au SGBD pour passer des ordres. Ces ordres sont de deux natures : lecture (on parle alors de requêtes) ou mise à jour (on parle alors de transactions). Pour les transactions, il y a une modification des données sur le serveur, mais cela reste des ordres de courte durée. À l’inverse, dans le cas d’une lecture, il n’y a pas de modification des données mais les traitements peuvent être longs et porter sur une grande masse de données.

Lors de l’utilisation d’une application de base de données, les problèmes majeurs rencontrés par cette dernière est la disponibilité des données, le temps de réponse et la résistance de l’application à d’éventuel pannes.

Et l’une des solutions à ce problématique est la réplication des bases de données.

La réplication est une puissante fonctionnalité de SQL Server qui permet de distribuer les données et d’exécuter les procédures stockées sur plusieurs sites. Cette technologie a considérablement évolué et permet maintenant de copier, déplacer les données à différents endroits et de synchroniser automatiquement les données.

Dans SQL Server distingue deux grandes catégories de réplication :

* La réplication de serveur à serveur ;
* La réplication de serveur à client ;

Dans le cas de la réplication de serveur à serveur, il y a une meilleur intégration ou rapprochement des données entre plusieurs serveurs de base de données.

La réplication de serveur à client concerne principalement les utilisateurs déconnectés du réseau de l’entreprise et qui souhaitent travailler avec les données de l’entreprise. Ces utilisateurs travaillent avec une application spécifique et utilisent SQL Server comme serveur de base de données local. Et lorsqu’il se connecte sur le réseau, la synchronisation des données entre leur poste et l’instance SQL Server central est prise en compte par le mécanisme de réplication de SQL Server.

1. Principe

La réplication met en jeu au minimum deux SGBD et consiste en un processus de propagation d’éventuelle modification réaliser sur la base de données et se déroule généralement en trois étapes :

* La base maitresse reçoit un ordre de mise à jour (INSERT, UPDATE ou DELETE) ;
* Les modifications faites sur les données sont détectées et stockées (dans une table, un fichier, une queue) en vue de leur propagation ;
* Un processus de réplication prend en charge la propagation des modifications à faire sur une seconde base dite esclave. Il peut bien entendu y avoir plus d’une base esclave.

Bien entendu il est tout à fait possible de faire de la réplication dans le deux sens (de l’esclave vers le maitre et inversement). On parlera dans ce cas de la réplication bidirectionnelle ou symétrique. Dans le cas contraire la réplication est unidirectionnelle (seulement du maitre vers l’esclave) et on parle de la réplication en lecture seule ou asymétrique.

1. Buts de la réplication

L’objectif principal de la réplication est de faciliter l’accès aux données en augmentant la disponibilité. Soit parce que les données sont copiées sur différents sites permettant de répartir les requêtes, soit parce qu’un site peut prendre la relève lorsque le serveur principal s’écroule.

1. Quand utiliser les bases de données répliquées

La mise en œuvre de la réplication des bases de données n’est pas une mince affaire c’est pour ça qu’il ne faut l’appliqué que dans des scénarios où il sera vraiment utile tel que :

* Le partage des données à travers des postes distants connectées via un réseau étendu
* Améliorer l’accessibilité aux données du serveur
* Distribution des schémas de BDD
* Sauvegarde des données

1. Point fort et point faible de la réplication
   1. Point fort

* Elle améliore la performance
* Elle diminue la probabilité de la panne
  1. Point faible
* Gestion des mises à jour (surcout : échange de message inter – site)
* Cohérence d’une donnée répliquée par rapport à la donnée de référence

1. Mono maître / multi maître

* Le maître peut exécuter des requêtes de type lecture ou écriture.
* L’esclave ne peut exécuter que des requêtes de type lecture et des requêtes de réplication.

On parle alors de mono maitre pour désigner un système où il n’y a qu’un seul maitre.

Et de réplication multi maitre pour des systèmes possédant plus d’un maitre.

1. Type de propagation

Selon le sens de propagation de la réplication on parlera de :

* Réplication bidirectionnelle ou symétrique :

Si la réplication se passe dans le sens de l’esclave vers le maitre ;

* Réplication unidirectionnelle ou asymétrique :

Si la réplication se passe seulement dans le sens du maitre vers l’esclave

1. Type de mise à jour
   1. Mise à jour synchrone

Aussi appelée « Réplication en temps réel »

La synchronisation est effectuée en temps réel puisque chaque requête est déployée sur l’ensemble des bases de données avant la validation (commit) de la requête sur le serveur où la requête est exécutée.

Ce type de réplication assure un haut degré d’intégrité des données mais requiert une disponibilité permanente des serveurs et de la bande passante.

Ce type de réplication, fortement dépendant des pannes des systèmes, nécessite de gérer des transactions multi sites coûteuses en ressources.

* + 1. Réplication synchrone asymétrique

Elle utilise un site primaire qui propage les mises à jour en temps réel vers un ou plusieurs sites secondaires.

La table répliquée est immédiatement mise à jour pour chaque modification par utilisation de trigger sur la table maitresse.

* + 1. Réplication synchrone symétrique

Dans ce cas, il n’y a pas de table maitre, mais chaque table possède ses triggers, déclenchés lors d’une modification. Il est alors nécessaire de définir des priorités et de gérer les blocages des tables en attendant qu’une modification soit entièrement propagée. D’autre part, les triggers doivent différencier les mises à jour issue d’une réplication des mises à jour de requête directes.

* 1. Mise à jour asynchrone

La réplication asynchrone (aussi appelée « store and Forward ») stocke les opérations intervenues sur une base de données dans une queue locale pour les propager plus tard à l’aide d’un processus de synchronisation. Ce type de réplication est plus flexible que la réplication synchrone.

Il permet en effet de définir les intervalles de synchronisation, ce qui permet à un site de fonctionner même si un site de réplication n’est pas accessible. Si le site distant est victime d’une panne, l’absence de synchronisation n’empêche pas la consistance de la base maitresse.

* + 1. Mise à jour asynchrone asymétrique

Elle propage les mises à jour en temps différé via une file persistante. Les mises à jour seront exécutées ultérieurement, à partir d’un déclencheur externe, l’horloge par exemple.

* + 1. Mise à jour asynchrone symétrique

Dans ce cas, la mise à jour des tables répliquées est différée. Cette technique risque de provoquer des incohérences de données.

1. Accès au réseau

Afin que le processus de réplication se déroule sans encombre, certaines conditions élémentaires doivent être satisfaites sur l’accès au réseau :

* Si les serveurs SQL participant à la réplication se situent dans des domaines différents, des relations d’approbation doivent être établies entre ces domaines.
* L’agent SQL Server doit s’exécuter dans le contexte d’un compte d’utiliser du domaine. Si possible, l’agent SQL Server des différents serveurs SQL participant à la réplication utilise toujours le même compte d’utilisateur.

L’agent SQL Server ne doit utiliser ni un compte local ni un compte d’utilisateur local, car ces deux comptes ne permettent pas d’accéder aux ressources du réseau.

1. Méthode des réplications des bases de données

Dans le sql server, les différents modèles de réplication utilisent la métaphore ‘’Editeur – Abonné’’ afin de concevoir au mieux les modèles de réplication.

1. L’Editeur

Comme un éditeur de livres ou de journaux, un serveur Editeur met à la disposition des autres des données pour mettre en œuvre la réplication. L’éditeur conserve toute les données publiées (celle qui participent à la réplication) et tient à jour les modifications intervenues sur ces données. Pour les données publiées, l’éditeur est toujours unique.

1. Le distributeur

Le serveur de distribution est le serveur qui contient la base de données de distribution et qui fonction comme un magasin pour les données spécifiques de la réplication associées à un ou plusieurs serveurs de publication. Chaque serveur de publication stocke les données d’état de la réplication.

Rôle :

* Le rôle de serveur de distribution varie en fonction du type de réplication implémenté.
* En règle générale, il joue un rôle beaucoup plus important pour la réplication transactionnelle que pour la réplication de la fusion et la réplication d’instantané pour deux raison :

1. Toutes les transactions répliquées sont lues et écrites dans la base de la base de données de distribution
2. Les topologies de la réplication de fusion utilisent généralement des abonnements extraits de sorte que les agents s’exécutent sur chaque abonné, plutôt qu’ils s’exécutent tous sur le serveur de distribution.

Fonctionnalités :

Dans de nombreux cas, un même serveur de base de données fonctionne à la fois comme serveur de publication et comme serveur de distribution

Mémoire :

Le serveur sélectionné comme serveur de distribution doit avoir suffisamment d’espace disque et un processeur assez puissant pour prendre en charge la réplication et toutes les autres activités effectuées sur ce serveur.

1. Les Abonnés

Ce sont les serveurs SQL qui stockent une copie des informations publiées puis reçoivent les mises à jours de ces données.

Comme pour un magazine, les abonnés doivent passer des abonnements pour recevoir des données publiées. Il existe deux types d’abonnement :

* Abonnement envoyé

Dans un tel cas, c’est le distributeur qui se charge d’envoyer la mise à jour des données distribuées à tous les abonnés. Ce type d’abonnement est particulièrement bien adapté lorsque le temps de mise à jour des abonnés doit être réduit au minimum.

* Abonnement extrait

C’est l’abonné qui décide de souscrire ou non un abonnement. C’est par la suite l’abonné qui va demander régulièrement des mises à jour. Ce type d’abonnement convient bien lorsque les abonnés sont très nombreux, car la charge de travail serait trop importante pour le serveur distributeur. Les abonnements extraits sont également bien adaptés aux utilisateurs nomades, car lorsque l’utilisateur vient se reconnecter au réseau de l’entreprise, c’est son propre serveur qui demande la mise à jour de ses données.

1. Les Agents

Pour fonctionner correctement, la réplication nécessite que certains programmes s’exécutent de façon récurrents. Ces programmes portent le nom d’agent et leur exécution répétitive est possible car ils apparaissent sous forme de tâche planifiée. Les agents de la réplication fonctionnent donc sous le contrôle de SQL Server Agent.

Les agents de réplication sont :

* L’agent de capture instantanée : son rôle est de fournir une image exacte de la base répliquée à un instant. Cette image prend en compte les structures et les données. La totalité de cette capture est stockée dans des fichiers de captures instantanée et des informations de synchronisation sont stockées dans la base de distribution. Cet agent s’exécute sur le distributeur.
* L’agent de lecture du journal : cet agent, actif uniquement dans le cadre de la réplication transactionnelle, scrute le journal de base de données à laquelle il est attaché et copie les transactions qui concernent la réplication vers la base de distribution. Si des réplications transactionnelles sont définies sur plusieurs bases, alors chacune d’entre elle possède son propre agent de lecture du journal. Cet agent s’exécute sur le distributeur.
* L’agent de distribution : exécuté sur le serveur de distribution (abonnement poussé ») ou bien sur l’abonné (abonnement extrait), l’agent de distribution a en charge d’appliquer la capture instantanée et les transactions enregistrées dans la base de distribution. Chaque abonné dispose d’une instance spécifique de l’agent de distribution.
* L’agent de fusion : spécifique à la réplication de fusion, il se charge tout d’abord d’appliquer la capture instantanée sur l’abonné. Par la suite, il se connecte au serveur de publication et à l’abonné pour rapprocher les informations. Par défaut, il télécharge les modifications de l’abonné sur le serveur de la publication, puis il reporte sur l’abonné les modifications intervenues sur le serveur de publication. Chaque abonné à une réplication de fusion possède son propre agent de fusion.
* L’agent de lecture de la file d’attente : spécifique à la réplication transactionnelle avec l’option de mise à jour en attente, cet agent s’exécute sur le serveur de distribution. Il n’nen existe qu’une seule instance quel que soit le nombre d’abonnés. L’agent de lecture de la file d’attente, va reporter les modifications effectuées au niveau des abonnés sur le serveur de publication.

Les agents de réplications sont gérés soit par SQL Server Management Studio, soit par le moniteur de réplication d SQL Server.

* 1. La réplication intégrale

La réplication intégrale (ou snapshot replication ou encore réplication de capture instantanée) consiste à prendre une image instantanée des données publiées dans la base. Ce type de réplication demande une surcharge de travail peu importante pour le serveur éditeur car l’opération est ponctuelle. Les abonnés sont mis à jour en recopiant la totalité des données publiées plutôt que d’effectuer uniquement les modifications (INSERT, UPDATE et DELETE) qui sont intervenues. Cette réplication convient bien pour des publications de petit volume sinon les mises à jours des abonnés peuvent nécessiter des ressources réseau importantes.

La réplication de capture instantané est souvent utilisé lorsque les abonnés ont besoin d’accéder aux informations uniquement en lecture et qu’ils n’ont pas besoin de connaître les informations en temps réel.

* 1. Réplication par fusion

Il s’agit ici de surveiller les modifications d’une base de données source et de synchroniser les valeurs entre l’éditeur et les abonnés. Ces derniers peuvent tous effectuer des opérations de mise à jour sur les données distribuées. Si l’éditeur conserve la maitrise de la publication, ce n’est pas toujours les opérations effectuées sur l’éditeur qui prennent le pas sur celles effectuées sur l’abonné. Toutes les modification apportées à la base cible sont reportées dans la base source.

La réplication de fusion ne commence toujours pas par une capture instantanée.

* 1. Réplication transactionnelle

Elle peut être utilisée pour répliquer des tables (intégralement ou partiellement) et des procédures stockées.

Les réplications transactionnelles utilisent le journal pour connaître les modifications apportées aux données publiées. Ces modifications sont stockées tout d’abord sur la base de distribution avant d’être envoyées sur les abonnés.

Toutes les publications possèdent un éditeur. Les modifications apportées à l’éditeur sont recopiées dans la base de distribution avec un laps de temps plus ou moins long. Dans un environnement où les connexions sont optimales, le temps de latence entre l’éditeur et les abonnés peut être très faible.

Cette réplication fonctionne aussi bien avec des abonnements envoyés que des abonnements extraits.

1. Mise en place
   1. Configuration initiales

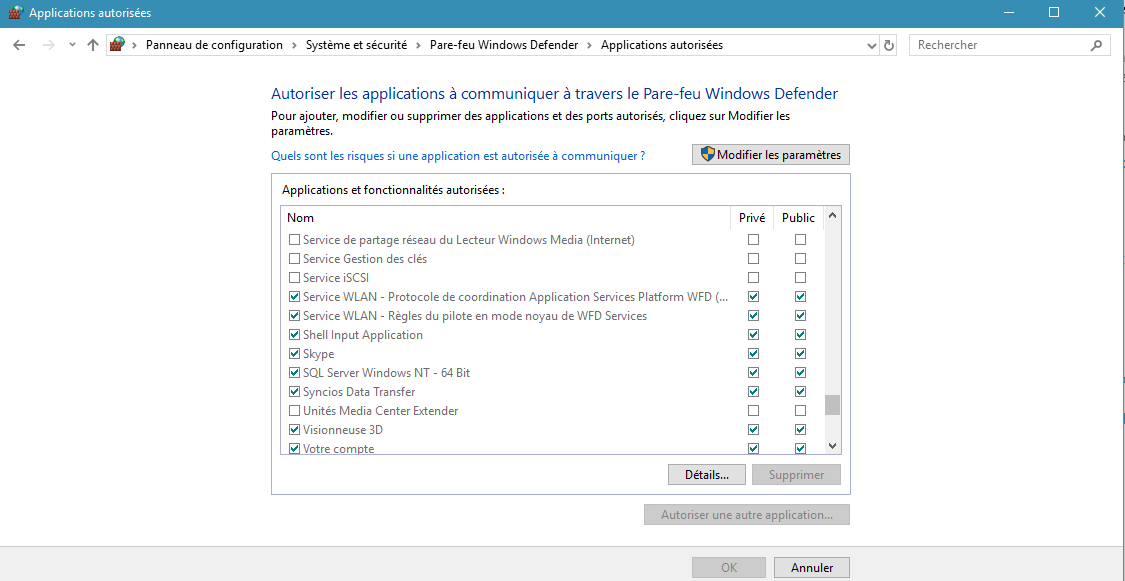
SQL server est un SGBD, quand il est installé sur une machine, il a la possibilité de pouvoir communiquer avec d’autres dans le réseau si la machine est connectée à un réseau. C’est ainsi que l’ensemble de SQL Server connecté forme un SGBDR, c’est – à – dire, donnant cette possibilité de déployer des bases de données réparties.

* + 1. Le pare – feu

Le pare – feu est un logiciel permettant de faire respecter la politique de sécurité du réseau, celle – ci définissant quels sont les types de communication autorisés sur ce réseau informatique.

Et celui installé à Windows empêche les utilisateurs ou logiciel non autorisés d’accéder à votre ordinateur depuis un réseau ou internet.

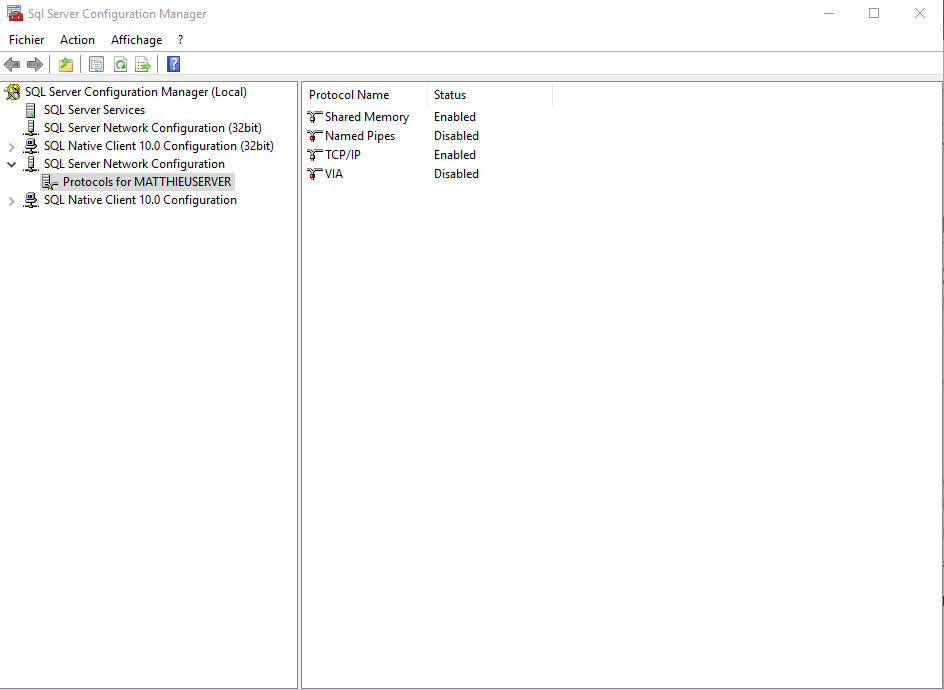
Il se pourrait qu’il empêche donc que des instances de SQL server installées sur différentes machines du réseau se communiquent. Pour ce faire, il faut soit le désactiver, soit créer un tunnel de communication, c’est – à – dire, autoriser un programme à recevoir et émettre bien que le Pare – Feu est activité



* + 1. Configurer SQL Server pour fonctionner dans le Réseau

Ici, il faut suivre un certain nombre d’étapes :

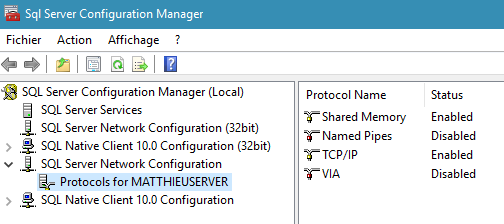
* Menu Démarrer -> Tous les programmes - > Microsoft SQL Server 2008 -> Outils de Configuration -> Gestionnaire de configuration SQL Server
* Voici la fenêtre qui va s’afficher



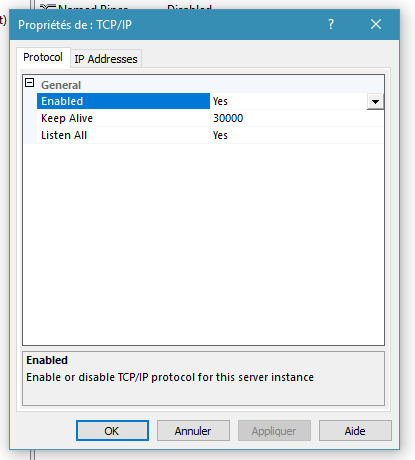
Ici, on peut voir toutes les services disponibles de SQL server, leurs états et autres information sont aussi affichés.

L’administration de services de SQL Server peut aussi se faire ici.

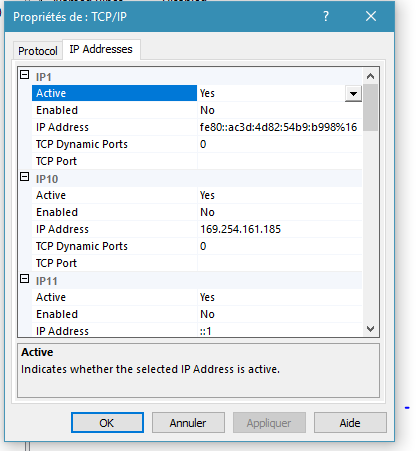
* Pour configurer la possibilité de communiquer dans un réseau, on a :



Faire un clic droit -> Propriété, et on aura la boite de dialogue



C’est ici qu’on peut activer ou désactiver les adresses et ports TCP/ IP et autres



C’est ici qu’il faut mettre l’adresse IP de la machine et le numéro de port, par lequel SQL Server de notre machine peut être repérable dans le réseau.

Il est à noter que si des serveurs qui participent à la réplication résident au sein de domaines Microsoft Windows NT Server distincts, vous devez établir des relations d’approbation entre ou parmi ces domaines.

Quelles données sont publiées ?

Il est important de répliquer uniquement les données nécessaires afin de réduire l’utilisation de l’espace disque, le temps du traitement et les E/S du réseau

Si vous disséminez des données, vous avez le choix entre les deux possibilités :

* Créer une publication de données globales à laquelle tous les sites d’abonneront et une publication de données partitionnées pour chaque abonné.
* Créer une publication pour chaque abonné qui contiendra les données globales et les données partitionnées.

Comment la synchronisation sera – t – elle effectuée : planifiée ou sur demande ?

Si vos besoins d’application autorisent une certaine latence, vous pouvez planifier des mises à jour peu fréquentes.

Quelle est votre topologie de réplication ?

L’établissement des rôles de serveur détermine le cadre physique de l’implémentation de la réplication. Nous devons :

* Sélectionner le modèle physique de la réplication ;
* Déterminer si le distributeur sera local ou distant ;
* Déterminer si la base de données distribution sera partagée ;
* Déterminer si les abonnements envoyés utilisent les ressources du distributeur, et les abonnements extraits celles de l’abonné

1. Modèle physique de la réplication

Tous les modèles physiques présentés ci – après sont indépendants du type de réplication choisi.

1. Editeur central – abonnés multiples

C’est le modèle par défaut dans SQL Server. Le serveur éditeur joue également le rôle de distributeur. L’éditeur et le distributeur peuvent s’exécuter sur deux serveurs différents. Il est également possible d’utiliser la même distribution pour plusieurs éditeurs.

Le serveur de publication (l’éditeur) est le propriétaire de toutes les données répliquées. le serveur de distribution stocke les données en attendant qu’elles soient envoyées sur les abonnés. Les données reçues sur les abonnés doivent être accessibles en lecture seule : tous les utilisateurs ne possèdent que l permission SELECT.

Il faut noter que si le serveur et le distributeur sont sur deux machines distinctes, la plus grosse partie du travail de la réplication est prise en charge par le distributeur.

1. Abonné central – éditeur multiples

Dans ce cas, plusieurs éditeurs répliquent les données vers un abonné central. L’abonné centralise les informations de tous les éditeurs. Il possède une vue globale de la situation tandis que chaque éditeur possède une vue locale de l’information. Étant donné que plusieurs éditeurs vont inscrire de l’information dans la même table d’abonnement, afin d’éviter la perte d’information, il est important que chaque donnée soit clairement possédée par un seul éditeur. Le moyen le plus simple à mettre en œuvre est un filtrage horizontal des données.

1. Éditeurs multiples – abonnés multiples

Dans ce cas, les éditeurs sont également des abonnés multiples. Ce cas pourrait se produire par exemple entre plusieurs points de vente travaillant indépendamment les uns des autre mais souhaitant connaître les différents stocks des points de ventes.

1. Conclusion

La réplication des bases de données est un concept très intéressant permettant de réglé beaucoup de problème dans le domaine du stockage des données. Son utilité est indiscutable et nécessaire à beaucoup de système.

Nous espérons qu’à travers notre travail avoir su vous initier à ce concept, son utilité, sa lise en œuvre, et ses domaines d’application.