

Haute disponibilité avec MS SQL Server



MVP SQL Server

Expert langage SQL, SGBDR, modélisation de données Enseigne à l'ISEN Toulon et aux Arts & Métiers

par Frédéric Brouard ([SQLpro](#))

Date de publication : Avril 2006

Dernière mise à jour :

La haute disponibilité est le fait de s'assurer des conditions optimales de fonctionnement en continu d'un système abritant un serveur SQL. Aujourd'hui ce genre d'exigence est au niveau des "5 neufs", c'est à dire une disponibilité du système de 99,999 %. Seules quelques grandes marques s'engagent sur de tels chiffres, comme HP, car avec un tel taux de disponibilité, vous n'avez droit qu'à un seul arrêt du système d'environ 5 minutes par an, soit juste le temps de passer un "Service Pack"

La haute disponibilité

La perte d'exploitation

Contexte de l'incident

1 - Avant tout se prémunir de la défaillance

2 - La solution de clusterisation

3 - Les solutions logiques

3.1 - Basculement avec sauvegardes

3.2 - Log Shipping

3.3 - Réplication

3.4 - Mirroring

3.5 - Solutions externes

De plus amples informations vous sont nécessaires ?

La haute disponibilité

La notion de haute disponibilité doit être établie par rapport à l'exigence de performance de la solution informatique.

La question fondamentale est :

quelles sont les données que l'on accepte de perdre en fonction du contexte de survenance d'un incident ?

La question sous-jacente est :

Quel coût financier de la solution de continuité doit être envisagé en regard de la perte de production ?

Elle pose directement le problème d'un retour sur investissement, même si l'investissement a des chances de ne jamais être consommé (absence d'incident). C'est donc à un calcul de probabilité calqué sur les modèles des compagnies d'assurance qu'il faut se prêter.

La perte d'exploitation

La mesure de la perte se chiffre généralement en terme de durée d'exploitation perdue. A une durée d'exploitation perdue peut correspondre une durée de remise en état beaucoup plus longue afin de rétablir le système comme à l'origine.

Par exemple, à une perte d'exploitation des trois dernières minutes de production peut correspondre une durée de remise en état d'une heure.

Contexte de l'incident

Il convient de prendre en compte la force et la gravité de l'incident comme critère à intégrer dans le processus de haute disponibilité. Il n'est pas toujours possible, de prendre en compte tous les incidents sans tenir compte du coût ou de la logistique globale de mise en oeuvre de la solution de dépannage.

Par exemple en cas d'incendie ou doit considérer le redémarrage "global" du système : nouveaux locaux, information du personnel... autant de préalables qui finalement font du coût de la solution de haute disponibilité, un élément presque anecdotique.

On comprend donc qu'assurer une haute disponibilité avec une perte nulle de la production et pour tous les types d'incident est relève d'une gageure financière pas toujours en adéquation avec les budgets informatiques.

Ce document présente donc différentes approches pour ce faire en détaillant les avantages et les inconvénients de chacune des méthodes. Il ne s'occupe pas du volet "sécurité" qui relève du domaine des administrateurs de systèmes informatiques et des stratégies de l'entreprise.

1 - Avant tout se prémunir de la défaillance

Se prémunir de la défaillance consiste à utiliser un serveur configuré de telle façon que les principaux incidents n'ait que peu ou pas d'influence sur la continuité de la production.

Dans ce cadre, il faut considérer au niveau matériel, un serveur configuré de la sorte :

- Une alimentation redondante
- Mémoire RAM autocorrective
- Disque RAID hot plug avec 3 agrégats (un RAID niveau 5 et deux RAID niveau 1)
- Onduleur online

On veillera de plus à stocker un disque hot plug en "spare" de façon à pallier immédiatement à la panne.


En outre il faut entretenir un serveur de secours capable de reprendre les mêmes disques hot plug que le serveur principal. Le serveur de secours étant configuré à l'identique au niveau hard et soft (OS + SQL Server).

Les fichiers seront répartis comme suit afin de réparer les défaillances :


- RAID 5 : fichiers contenant les données des bases (DATA)
- RAID 1 (grappe 2) : fichiers contenant les journaux des bases (LOG)
- RAID 1 (grappe 2) : autres fichiers (OS, exe, mémoire virtuelle...)

Pannes possibles	Remède
Coupure réseau électrique	Onduleur
Défaut de RAM	Auto correction soft
Défaut du sous système de contrôle disque (RAID, SCSI, SATA...)	Enficher * les disques DATA et LOG du serveur en panne dans le serveur de secours. Copier les fichiers nécessaire au redémarrage. Lancer les procédures adéquates pour reprendre la main sur SQL Server et la base de production.
Défaillance d'un disque	Remplacement à chaud du disque
Défaillance d'un fichier	<p>Si fichier OS ou exe : enficher * les disques DATA et LOG du serveur en panne dans le serveur de secours. Copier les fichiers nécessaires au redémarrage. Lancer les procédures adéquates pour reprendre la main sur SQL Server et la base de production.</p> <p>Si fichier DATA SQL : procédure de reprise depuis fichier LOG SQL.</p>

	Si fichier LOG SQL : procédure de reprise depuis fichier DATA SQL.
Défaillance processeur	Enficher * les disques DATA et LOG du serveur en panne dans le serveur de secours. Copier les fichiers nécessaire au redémarrage. Lancer les procédures adéquates pour reprendre la main sur SQL Server et la base de production.

 * cette pratique n'est pas garantie par Microsoft. En effet, les fichiers des données comme ceux des journaux peuvent ne pas pouvoir être repris tel quel parce qu'il sont ouvert à l'usage exclusif de MS SQL Server tant que le serveur SQL tourne (en principe 24h/24).

Bien entendu il est toujours possible de repartir d'une sauvegarde mais la perte de production dans ce cas est lié au delta entre deux sauvegardes.

 **NOTA** : MS SQL Server est capable de sauvegardes très légères (différentielles, JT) que l'on peut programmer à fréquences plus ou moins élevées (jusqu'au 1/4 d'heure par exemple).

2 - La solution de clusterisation

C'est la seule solution actuellement garantie par MS en terme de reprise d'exploitation sans perte de données.

Pour cela il faut :

- Deux serveurs identiques, choisis dans une liste des matériels approuvés par Microsoft,
- Une baie de disque partagée, (SAN) choisie dans une liste des matériels approuvés par Microsoft,
- La solution MS Windows 2003 Server Clustering

Cette solution est couteuse en deux endroits :

- Le matériel approuvé par MS est très limité et ce sont des serveurs hauts de gammes (cette exigence disparaît avec la version 2005 sous Windows Server 2003).
- L'installation du clustering nécessite des compétences étendues au niveau système.

Néanmoins il s'agit d'une solution simple en exploitation car elle est transparente et permet un basculement automatique sans même qu'aucune intervention humaine soit nécessaire.

Le seul inconvénient technique de cette solution est un temps de latence du basculement qui est de quelques dizaines de secondes.

Article en français expliquant les principes du clustering :

- <http://www.itpro.fr/article.asp?mag=3&th=9&ss=8&id=2064>

Document en anglais de la solution MS :

- <http://download.microsoft.com/download/d/d/7/dd75ece7-83de-45da-8bb1-cb233decf595/BDMTDM.doc>

Avantages	Inconvénients
<ul style="list-style-type: none">• Seule solution garantie sans pertes de données.• Pas ou peu d'administration• Basculement automatique• Faible coût d'exploitation• Faible consommation des ressources du serveur	<ul style="list-style-type: none">• Couteuse en matériel et logiciel.• Installation complexe nécessitant du personnel qualifié• Concerne tout le serveur• "single point of failure" : le SAN...

Mais à bien y regarder, cette solution n'est pas toujours aussi couteuse qu'on le croit parce qu'un SAN économise du disque. En revanche, le SAN reste le talon d'Achille car en cas de sérieux problème sur ce dispositif, c'est l'ensemble du système qui est en panne !

3 - Les solutions logiques

3.1 - Basculement avec sauvegardes

La solution de basculement avec sauvegarde consiste à mettre en exploitation sur un serveur de secours la plus récente sauvegarde de la base de données et repartir de cette sauvegarde pour commencer une nouvelle exploitation.

Les pertes engendrées dépendent donc :

- du temps de latence des sauvegardes pour la perte des données
- des points de reprise fonctionnels pour la perte d'exploitation (vague en cours ?)

Pour réduire la perte des données on peut augmenter les fréquences des sauvegardes. Dans ce cas un plan de sauvegarde alternant des sauvegardes complète la nuit, des différentielles en production et des sauvegardes du journal de transaction au fil de l'eau peuvent réduire considérablement les temps de latence et faire en sorte que la perte des données soit réduite à une moyenne de quelques minutes d'exploitation.

Avantages	Inconvénients
<ul style="list-style-type: none"> • Très faible coût matériel (ne nécessite pas l'acquisition préalable d'un serveur de secours) • Faible coût d'installation • Faible consommation des ressources du serveur • Concerne une ou plusieurs bases 	<ul style="list-style-type: none"> • Pertes importantes possibles • Basculement manuel et long • Nécessite une administration au quotidien

3.2 - Log Shipping

Le log shipping (littéralement "envoi de journaux") est une technique qui consiste à envoyer de manière planifiée et régulière la partie du journal de la base qui contient les dernières transactions achevées.

En quelques sortes, le Log Shipping introduit de manière automatique la solution vue précédemment.

Les scripts de mise en oeuvre de cette technique figurent dans le CD de l'édition Entreprise de MS SQL Server et sa simplicité fait qu'elle peut être mise en oeuvre sur n'importe quel serveur de n'importe quelle édition.

Il est possible de descendre le temps de latence entre deux envois de journaux à quelques minutes, afin de minimiser les pertes.

Avantages	Inconvénients
<ul style="list-style-type: none"> • Faible coût d'installation • Très faible cout d'administration • Faible consommation des ressources du serveur • Concerne une ou plusieurs bases 	<ul style="list-style-type: none"> • Pertes moyennes possibles • Basculement manuel

Dans la version SQL Server 2000, le Log Shipping constitue souvent la solution de meilleurs compromis entre le coût à tous niveaux et la perte possible.

3.3 - Réplication

La réplication consiste à dupliquer des informations d'une base à une autre. La granularité de la réplication permet de descendre jusqu'à la donnée unitaire (une colonne d'une ligne d'une table). Le système repose donc sur des articles qui sont des parties de tables (filtrage horizontal et vertical), qui sont mis à disposition sur un distributeur que les abonnés viennent reprendre.

Suivant les différents modèles de réplication, la reprise des données se fait par "push" ou "pull".

La réplication nécessite en fait 3 serveurs SQL : le répliqué (éditeur), le serveur de publication et le serveur abonné. Mais en fait le serveur de publication peut être l'un des deux serveurs, et dans le cas de solution de continuité, le mieux est que le serveur de publication soit aussi le serveur abonné.

Les différents modèle de réplifications sont les suivants :

- capture instantanée
- transactionnelle
- fusion

Ces réplifications peuvent être combinées avec une mise à jour immédiate de l'abonnée ou différée.

Dans le cas qui nous préoccupe (solution de continuité) il convient de choisir la solution de réplication dont le temps de latence peut être le plus court. Dans ce cadre il convient donc de préférer la réplication transactionnelle avec mise à jour immédiate. Le gros inconvénient dans ce cas est que les modifications du schéma ne sont pas prises en compte et nécessite un script à jouer simultanément sur les deux serveurs ainsi qu'une modification de la définition des objets répliqués.

Avantages	Inconvénients
<ul style="list-style-type: none"> • Ne nécessite pas de basculement • Concerne une ou plusieurs bases, voire même un élément (sous ensemble d'une table) 	<ul style="list-style-type: none"> • Cout d'installation élevé • Consommation élevée des ressources du serveur suivant la position du distributeur • Cout d'exploitation élevé • Pertes faibles à moyenne possibles • Les modifications de schéma peuvent ne pas être prise en compte suivant le mode de réplication

Cette solution est celle qui minimise les pertes après le clustering, mais son coût, lié à sa complexité est très élevé, en particulier dans les bases de données dont la structure est sujette à modification.

3.4 - Mirroring


Le mirroring est un concept nouveau introduit avec la version 2005 de MS SQL Server. Ce technique de mise en miroir est simple et pratique. Elle consiste avant tout à dupliquer intégralement une base de données en temps réel, base qui sera alors accessible en lecture seule tant que le miroir est actif et en lecture écriture si le miroir est brisé. Comme dans le cas du clustering, la reprise peut être automatisé et en pratique le basculement s'opère en

quelques dizaines de secondes, à condition d'avoir mis en place un serveur de scrutation, qui peut être constitué par une machine base de game (simple PC) avec le run time SQL Server (Express 2005).

Cette solution n'exige en outre aucun matériel spécifique.

En terme de richesse, souplesse et coût, cette solution s'avère la mieux adaptée pour la haute disponibilité d'un faible nombre de bases de données (quelques unités).

Avantages	Inconvénients
<ul style="list-style-type: none">• Pas de surcoût matériel• Basculement automatique possible• Concerne une ou plusieurs bases• Pas de pertes de données• Pas ou peu d'administration• Faible coût d'installation• Faible coût d'exploitation• Faible consommation des ressources du serveur	<ul style="list-style-type: none">• Disponible uniquement en version 2005

 *S'il y avait une solution à retenir dans le cadre d'epsilon, ce serait certainement celle-là qui possède tous les avantages mais nécessite la validation de notre logiciel dans la version MS SQL Server 2005.*

3.5 - Solutions externes

Diverses solutions externes plus ou moins couteuses sont disponible afin d'assurer une haute disponibilité avec reprise automatique. Les plus éprouvées sont dans l'ordre :

Double Take

- <http://www.nsisoftware.com/what-we-offer/double-take/>

Never Fail :

- <http://www.neverfailgroup.com/products/nfsql.asp>

WANSyncHA :

- http://www.xosoft.com/products/f_WANSyncHASQL.shtml

SQL backup de redGate software :

- http://www.red-gate.com/products/sql_backup/technical_papers/log_shipping.htm

Seule une étude comparative poussée permettrait d'en présenter avantages, inconvénients et coûts.

De plus amples informations vous sont nécessaires ?

Venez en discuter sur le forum public français de Microsoft SQL Server :

<news://msnews.microsoft.com/microsoft.public.fr.sqlserver>

Ou sur le forum SQL Server de developpez.com :

<http://www.developpez.net/forums/forumdisplay.php?f=49>

