

Extrait du Easter-eggs - Spécialiste GNU/Linux

<http://www.easter-eggs.com>

# Cluster Tomcat/MySQL à la Sonacotra

- Études de cas - Intégration systèmes et réseaux -

Date de mise en ligne : jeudi 9 novembre 2006

---

Easter-eggs - Spécialiste GNU/Linux

---

## Sommaire

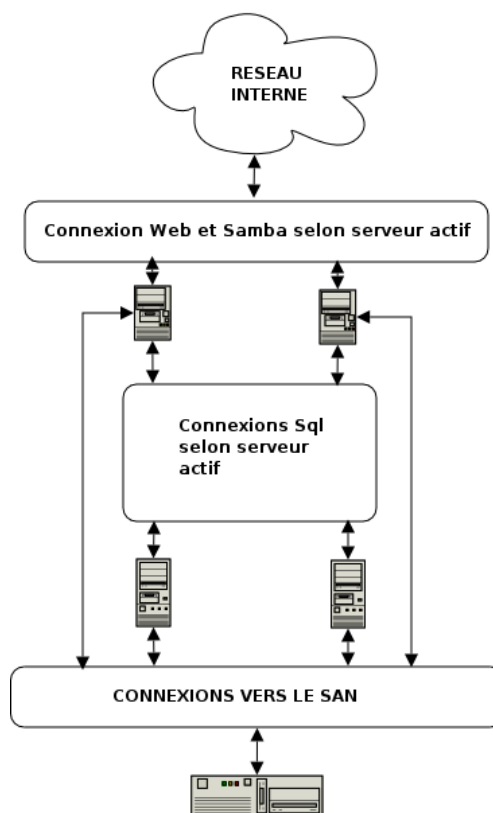
- [Contexte](#)
- [Architecture mise en place](#)
- [Mise en place du clustering](#)
- [Améliorations et orientations](#)

## Contexte

Le but de cette prestation était de migrer l'application intranet du client d'une plate-forme "Répartition de Charge / Haute Disponibilité" sous système d'exploitation [Debian](#) vers une plate-forme "Haute-Disponibilité" d'après un cahier des charges fourni par le client et non-modifiable, sous système d'exploitation [RedHat 4 Enterprise Linux](#).

## Architecture mise en place

<!-- htmlA -->



<!-- htmlB -->

L'ensemble est configuré en haute-Disponibilité avec l'outil [ServiceGuard](#), système de clustering offrant des outils d'administration en ligne de commande ou par le biais d'un client web Java fourni par [HP](#).

### Les serveurs SQL

Sur chaque serveur, un serveur [Mysql en multi-instances](#) fournissant deux instances dont une seule est active. Ce système de multi-instance permet d'installer un seul paquet Mysql mais de disposer de deux serveurs distincts l'un de l'autre utilisant chacun un port et un socket spécifiques

### Les serveurs Web

- un serveur [Apache2](#) dédié à Php4
- un serveur [Samba](#) dédié aux répertoires des sites Php4
- un serveur Apache2 dédié à Php5
- un serveur Samba dédié aux répertoires des sites Php5

Afin de pouvoir faire fonctionner deux serveur Apache et deux serveurs Samba sur le même serveur il a été nécessaire d'installer deux fois les paquets sur chaque machines et de les faire écouter chacuns sur une adresse ip dédiée ainsi Php4 écoute sur IP\_PHP4 et PHP5 écoute sur IP\_PHP5.

- Un serveur [Java/Tomcat](#)
- [Winbind](#) pour les uid/gid des utilisateurs du domaine.

### Espace de stockage

L'espace de stockage de toutes les données communes aux services est situé sur le SAN.

Les différents services (Mysql1, Mysql2, [Php4](#), [Php5](#) et Java) ont chacun un volume spécifique.

Ces volumes sont basculés automatiquement d'un serveur à l'autre lorsque le service de monitoring du cluster détecte une défaillance de service.

## Mise en place du clustering

L'outil déployé est ServiceGuard de HP. Il a l'avantage :

- d'être packagé pour RHEL4 et est donc directement installable sur le système,
- d'avoir une version pour le 64bits
- de posséder des scripts de monitoring pour des services en multi-instances comme dans notre cas.

Il a tout de même été nécessaire de modifier ces scripts fonctionnant dans un système de package, chaque package étant prévu pour monitorer un seul et unique service comprenant le service, l'espace disque qui lui est associé et une adresse IP virtuelle attribuée.

Dans le cas de Php4 par exemple, le volume disque de Samba-Php4 étant le même que celui de Apache-Php4 il n'était pas possible de monitorer les services séparément. En effet, si le service Samba-Php4 est défaillant le gestionnaire du cluster va démonter le volume locale correspondant au service, remonter ce volume et redémarrer le service sur le deuxième serveur du cluster. Cela implique que le service Apache-Php4 local n'a plus de données.

Nous avons donc modifié les scripts afin qu'un seul package puisse monitorer plusieurs services et faire basculer

simultanément l'ensemble des services monitorés d'un serveur à l'autre.

Il également fallu modifier le système de monitoring en lui même afin qu'il ne se base pas uniquement sur l'absence d'un processus pour décider la migration d'un service depuis un serveur vers l'autre.

Nous avons pris en compte la notion de processus père et fils qui n'étaient pas gérés correctement.

## Améliorations et orientations envisageables

- Remplacer RHEL4 par une [Debian Sarge](#). Cela permettrait de s'affranchir de paquets inutiles comme toute la gestion des interfaces graphiques, sources de failles possibles de sécurité et de bugs.
- Ne pas utiliser ServiceGuard mais la version 2 de [HeartBeat](#) qui possède une interface d'administration en mode graphique.
- Vérifier la réelle présence du service car un processus peut être présent mais le service indisponible.
- Vérifications du service web par requêtes http sur une page spécifique afin de contrôler que l'on obtient bien le contenu de cette page.
- Contrôle du bon fonctionnement de Mysql à l'aide de mysqladmin et l'option ping ou avec mysqlcheck, outils disponibles lors de l'installation d'un serveur Mysql.
- Vérification de Samba à l'aide de smbclient fourni lors de l'installation de Samba.
- Vérification de Winbind en recherchant un uid spécifique appartenant au domaine Windows.
- Ne plus utiliser le module [Ntlm](#) pour apache Linux qui rencontre des problèmes de communication réseau vers un contrôleur de domaine Windows et décentraliser l'authentification sur un serveur [OpenLdap](#) en utilisant le module ldapauth(24) d'Apache.
- Vérifier l'état des connexions réseau à l'aide de ping vers différentes adresses IP internes et externes au réseau afin de déterminer si une rupture de connexion vers un service est due à une défaillance du réseau ou de la carte ethernet du serveur et migrer les services vers l'autre serveur seulement si le problème est local à la machine.
- Faire un contrôle de l'état des disques à l'aide de [smartmontool](#) lorsque l'espace de stockage n'est pas un SAN.