

Développement d'un système de haute disponibilité par réplication de base de données

Type de contenu : Texte

Titre(s) : Développement d'un système de haute disponibilité par réplication de base de données ; BARTHEYE, Olivier ; DECKER, Hendrik ; García, José Luis ; SLT THIRIET, Adrien

Autre(s) responsabilité(s) : BARTHEYE, Olivier (Directeur de thèse)
DECKER, Hendrik ; García, José Luis (Directeur de thèse)
SLT THIRIET, Adrien Promotion Capitaine de Cacqueray (2009-2012) (Secrétaire)

Editeur, producteur : Ecoles Militaires de Saint-Cyr Coëtquidan

Description matérielle : 1 CD

Note sur le contenu : mémoire

Note de thèses et écrits académiques : Filière Scientifique - Option Informatique Promotion Capitaine de Cacqueray Date de soutenance : 01/01/2012

Résumé ou extrait : **PRESENTATION** : Aujourd'hui, les bases de données sont très utilisées, surtout dans le monde des entreprises. Certaines de ces bases sont connectées à Internet ou à un réseau afin d'être accessible à un client à partir d'un autre point. Pour une entreprise comme une banque ou même un réseau social, leurs bases de données doivent être accessibles tout le temps, même en cas de panne du serveur principal. Pour ce faire on utilise un système de haute, celui-ci est construit avec un système de réplication de bases de données. A l'ITI, j'ai été attaché à une équipe de développement, ils travaillent sur le projet ITICAB, un service de contrôle de présence à l'aide d'un dispositif à reconnaissance faciale. Ce produit doit être continuellement en service parce que s'il est arrêté, personne ne peut entrer dans l'entreprise. Mon travail consistait à construire ce système de haute disponibilité pour la base de données du projet. Ils travaillent sur ce projet depuis deux ans, mais avec la crise économique actuelle, l'équipe a été réduite pour privilégier des projets plus rentables. Alors ils attendaient de moi de les avancer au maximum. **CONSTRAINTES** : La base de données du projet comprend deux parties, les données biométriques et la table d'activité qui contient toutes les données concernant les heures d'arrivée et de sortie des employés. Les premières informations sont très longues à transférer par le réseau en raison de l'espace qu'il occupe sur le disque. La plus grosse contrainte est le problème de connexion, en effet le système comprend trois bases de données (1, 2 et 3) et deux interfaces client (A et B). Les connexions A-1 et B-2 sont très rapides, les connexions entre les trois bases de données sont très lentes. Avec la taille des données à transférer, il y a une forte probabilité de les perdre entre les interfaces clients et les bases. C'est pourquoi j'ai dû construire un système de réplication maître-esclave à partir du maître 2 aux esclaves 1 et 3. Pour résoudre le problème précédent, quand un client écrit dans l'interface A, par exemple, les données doivent être écrites d'abord dans la base de 1 et ensuite être envoyées au maître afin d'être répliqué. Avec cela, mon système doit inclure un basculement automatique et d'un programme de récupération qui peut être lancé par l'administrateur. **DEMARCHE** : J'ai commencé à construire un

système maître-esclave en utilisant la réplication incluse dans PostgreSQL depuis la version 9.1. Ensuite, j'ai ajouté un système de haute disponibilité sur lui avec un basculement et la récupération. Pour l'exécuter, j'ai utilisé un service de messagerie Web: Spread utilisé avec langage Python. Celui-ci s'occupe de la détection de tous les serveurs et fonctionnent comme un déclencheur. Si une base de données est fermée ou présente une défaillance, Spread va lancer le basculement. Le client peut envoyer ses requêtes à Spread, celui-ci va les rediriger au serveur principal actuel. Ce système était efficace, mais il ne peut pas résoudre le problème de connexion, donc j'ai dû trouver une autre solution. J'ai cherché sur internet un nouveau logiciel de réplication. J'en ai téléchargé et essayé certains, comme Bucardo, Slony ou Pg_Cluster. Enfin, j'ai décidé de construire un nouveau système, mais pas un maître-esclave, j'ai essayé un système maître-maître avec Pgpool II. Le client peut envoyer ses requêtes à pgpool et celui-ci les enverra à toutes les bases, sauf si c'est une requête de lecture. Dans ce cas, pgpool va lire dans la base la plus proche, et donc la base avec une meilleure connexion. Pgpool gère lui-même le basculement et la récupération. CONCLUSION : J'ai construit deux systèmes de haute disponibilité de base de données répliquées: l'une avec PostgreSQL et Spread et l'autre avec Pgpool II. La première est efficace mais ne peut pas être utilisée dans le projet ITICAB. La seconde le peut mais n'est pas terminée. J'ai aidé l'équipe du projet en les évitant de perdre beaucoup de temps à bâtir une base de

Sujet(s) : SQL

base de données

biométrie

employé : catégorie

entreprise

reconnaissance faciale

récupération des données

système de protection