BERLAND Cédric
HENAFF Brewal
MARAVAL Nathan
Groupe 1

THEORIE DES SYSTEMES REPARTIES
RESEAU PAIR A PAIR

INTRODUCTION :

Nous avons mis en place un protocole pair à pair permettant de se connecter à un réseau pour communiquer avec d'autres pairs, sans avoir à passer par un serveur. Le protocole mis en place est fortement inspiré de Chord (notamment la disposition en cercle et les fingers). Un serveur est cependant utilisé lors de l'arrivée d'un nouveau pc sur le serveur pour attribuer un hash. Enfin une extension permet de gérer la disparition non annoncé d'un pair dans le réseau.

IMPLANTATION DE BASE :

TRANSMISSION D'UN MESSAGE :

Pour pouvoir communiquer avec les autres pairs, il suffit de connaître l'IP du hash suivant. De ce fait, on peut lui envoyer un paquet contenant le hash visé ainsi que le message. Le pair suivant va alors vérifier s'il est le destinataire avant de l'envoyer au suivant si ce n'est pas le cas. On ajoute au paquet le hash de l'envoyeur direct pour contrer le cas où le hash n'existe pas, on peut supprimer le paquet si le hash de destination est compris entre l'envoyeur et son successeur.

ARRIVEE D'UN PAIR DANS LE RESEAU :

Lorsqu'un pair se connecte il doit d'abords demander son hash au serveur, afin que les hash soit uniques. Ensuite il va s'insérer dans le réseau en demandant en prévenant le hash suivant. Dans cette optique chaque pair connaît l'IP de son prédécesseur. Ainsi dans un premier temps le nouveau pair prévient le successeur de son arrivé, et les 2 mettent leurs données à jour. Ensuite le successeur va prévenir son ancien prédécesseur du nouveau venu afin que celui-ci se remette à jour. Ainsi chacun connaît finalement le bon successeur, et la boucle est bouclée.

TABLES DE ROUTAGES DES PAIRS :

Le problème ici c'est qu'il faut à chaque fois parcourir tout le réseau pair par pair pour faire une action. Pour y remédier nous avons mis en place des fingers : une table de routage propre à chaque pair. Les fingers sont représentés par des HashMap, connaissant une dizaine de hash et d'IP du réseau. Ainsi pour envoyer un paquet il suffit de l'envoyer au pair avec l'hash le plus proche de l'hash du destinataire.

Notre politique est de prendre les hash suivants en calculant les puissances de 2. Par exemple le hash 0 va avoir dans sa table l'IP des hashs suivants : 1, 2, 4, 8, 16, 32… Dans notre cas le serveur est de taille 50, on va donc chercher jusqu'au 2^5 suivant. Il est cependant possible que le hash voulu ne soit pas attribué (par

exemple personne ne correspond au hash 16). On va alors prendre le pair le plus proche possédant un hash inférieur à celui recherché. Cette politique nous permet de bien repartir la table sur tout le réseau.

Afin de mettre son finger à jour, un pair va demander au successeur le plus proche connu de chaque puissance qui est le pair correspondant. Ce dernier va tout simplement renvoyer son IP au pair de base. Il faut noter que le finger perd de sa valeur dans le temps, car le réseau change. On va donc lancer le processus pour le mettre à jour toutes les 5 minutes automatiquement.

GERER LE DEPART, ATTENDU OU NON, D'UN PAIR.

Pour cette partie, 2 cas peuvent être décris.

DEPART ATTENDU D'UN PAIR :

Si un pair décide de partir de lui-même, la commande "exit" lui ai proposé, pour pouvoir sortir correctement du réseau.

Lorsque cette commande est rentré par l'utilisateur, le pair va d'abord envoyer à son prédécesseur un message contenant le successeur courant, et à ce successeur il envois le prédécesseur. Ainsi ils auront les bonnes valeurs pour pouvoir continuer les communications.

Après avoir fait ceci, le pair va envoyer au WelcomeServer un message contenant son hash, afin de le retirer de la table du WelcomeServer pour qu'un nouveau pair ne le choisisse pas comme IP entrante.

Enfin, avant de quitter, le pair envoi un dernier message à son successeur, qui n'attends pas de réponse et qui effectuera un tour complet du cercle, pour indiquer à tout le monde présent que le pair à quitter, et nécessite d'être retiré des tables de hachages.

DEPART IMPREVU D'UN PAIR :

Le premier cas est donc facile à mettre en œuvre, car le pair envoi lui-même les informations à ses voisins avant de partir. Cependant, si le pair venait à être déconnecter du réseau à cause d'une panne, les méthodes mises en œuvres seront donc bien différentes.

Tout d'abord, pour toujours s'assurer que le pair suivant est présent, un pair va donc lui envoyer toutes les 30 secondes un message, qui attendra une réponse, pour s'assurer que la connexion est bien stable.

Dans le cas où ce message n'est pas reçu, le pair peut donc assumer que son successeur à eu une déconnexion abrupte. A ce moment-là, il va donc vérifier si l'id de son successeur et l'id de son prédécesseur son égaux. Si c'est le cas, cela veut donc dire qu'avant la déconnexion, il n'y avait que deux personnes sur le network. Ce cas est donc facile, il n'y a qu'à annoncer au WelcomeServer le départ, et indiquer le successeur et le prédécesseur comme étant soi-même.

Cependant, dans l'autre cas, cela devient plus compliqué. Pour pouvoir connaître le "successeur du successeur", nous pourrions utiliser la table de routage, cependant nous avons remarqué que dans certains cas, certains successeur sont passés (dans le cas par exemple ou mon hash est 1, mon successeur 50 et son successeur 51. La plupart de la table de routage sera constitué de 50, et jamais n'apparaîtra 51). Pour pallier à cela, la méthode utilisée est celle d'envoyer à son prédécesseur le hash qui vient de partir, et de lui demander s'il est identique à son propre prédécesseur. Si c'est le cas, ce pair récupérera l'IP et le hash de son nouveau prédécesseur, et renverra à ce nouveau prédécesseur ses propres ip et hash, pour que celui-ci puisse les rentrer en tant que successeur. Sinon, il renverra le message à son prédécesseur, jusqu'à ce que ce message ai fait le tour de la boucle.

Cette méthode est donc peux viable dans de très gros réseaux, car le message devra donc faire le tour complet du cercle avant de retrouver le hash déconnecté.

N.B. : Avec l'utilisation du serveur d'accueil, lorsqu'un pair quitte le réseau il faut le signale au serveur d'accueil afin d'éviter que le pair quittant le réseau serve de point d'entrée à un futur arrivant.