

Projet #1 : construction d'une DHT

Le but de ce projet est de concevoir et implémenter une DHT en dessus de PeerSim.

Les premières étapes sont obligatoires. Elles sont là pour vous guider, vous lancer...

Ensuite, libre à vous d'imaginer des solutions / scénarios...

Vous devez rendre votre code (ou un lien sur votre code) et un compte rendu expliquant votre projet, les choix de conception. Si vous avez quelques courbes à montrer c'est

encore mieux .

Etape 1 (join/leave):

Dans un premier temps, votre DHT est un simple anneau où chaque nœud connaît uniquement ses voisins immédiats. Chaque nœud a un identifiant (aléatoire), les nœuds sont rangés dans l'anneau en fonction de leur identifiant).

Au démarrage il n'y a qu'un nœud (il est son propre voisin de droite et son propre voisin de gauche). Quand un nouveau nœud rejoint la plateforme, il contacte un des nœuds présents (n'importe lequel), il doit trouver sa place et s'insérer dans l'anneau. Quand un nœud quitte la plateforme, il contacte ses voisins de droite et gauche afin de les mettre en relation (son voisin de droite devient le voisin de droite de son voisin de gauche et inversement).

=> concevoir et coder ce système, l'implémenter et le tester.

Etape 2 (routing : Send/Deliver):

Si l'étape 1 est réalisée, celle-ci doit être assez rapide : vous avez déjà routé les messages de join des nouveaux nœuds.

Prévoir des messages (objets Java) qu'un nœud A peut envoyer à un nœud B.

Tester ! (penser à logger les envois / forward / réceptions de messages)

Etape 3 (storage : Put/Get):

Chaque nœud va maintenant être capable de stocker des données. (chaque nœud va posséder un tableau des données qu'il stocke).

Dans une DHT, chaque donnée se voit attribuer un identifiant, le nœud d'identifiant le plus proche est responsable de la donnée. La donnée doit être stockée sur le nœud responsable et sur les voisins immédiats (degré de réplication == 3).

Tester !

Etape 4 (Advanced routing):

Le routage de proche en proche n'est pas performant.

Proposer 2 moyens pour ajouter des liens longs :

- En "trichant" (en utilisant le fait que l'on a une vue globale au sein du simulateur)
- Sans tricher (en ajoutant des infos dans les messages qui passent 'piggybacking' et en les utilisant pour construire des tables de routage).

Pour aller plus loin

Prendre en compte la dynamicité...

a/ comment maintenir un routage correct / maintenir des tables de routage efficaces avec des départs arrivées de noeuds (départs propres ou crashes)

b/ comment maintenir le degré de réplication de chaque donnée