Implémentation du protocole pair à pair CAN

Mini-projet AR à réaliser sur deux semaines en binôme

Introduction

Le système pair à pair CAN se base sur une table de hachage distribuée (DHT). Une DHT permet de stocker chaque donnée en un point unique du réseau, défini à partir de l'identifiant de la donnée. Cela s'avère extrêmement performant lorsqu'il est question de localiser des données spécifiques dans un réseau à grande échelle.
La spécificité de CAN est de diviser l'espace en coordonnées cartésiennes. Chaque nœud se voit attribuer un point unique dans l'espace, ce qui détermine ensuite sa position par rapport aux autres nœuds. Chaque donnée est elle aussi associée à un point unique grâce à une fonction de hachage ; on peut ainsi créer de manière très simple un lien entre une donnée et le nœud sur lequel elle doit être stockée.

Objectif

Le but de ce projet est de simuler une implémentation bi-dimensionnelle de CAN en MPI. L'espace des coordonnées est le sous-ensemble des entiers naturels [0, 1000] x [0, 1000].
Votre implémentation devra permettre les opérations suivantes :

- Insertion d'un nœud,
- Insertion d'une donnée,
- Localisation d'une donnée.

Travail à réaliser

Dans l'ensemble de ce projet, chaque processus représentera un nœud et le rang du processus représentera l'adresse du nœud. Le processus de rang 0 fait exception : il ne participe pas à l'overlay mais sert de coordinateur. Son rôle sera dans un premier temps de synchroniser les opérations d'insertion des nœuds, puis d'agir en tant que client pour insérer/rechercher des données.
L'overlay contiendra N nœuds en tout. Le protocole de routage et les données conservées localement sur chaque nœud sont laissées à l'appréciation des étudiants concepteurs.

Etape 1 : Insertion de nœud

Chaque processus tire aléatoirement un identifiant : un couple de valeurs dans l'espace des coordonnées. Le coordinateur contacte un premier processus (celui de rang 1) pour l'inviter à s'insérer dans l'overlay. Chaque nœud une fois inséré notifie le coordinateur, qui invite alors le nœud suivant dans l'overlay.
Le premier processus à s'insérer (rang 1) hérite de l'ensemble de l'espace et devient également le nœud de *bootstrap* par lequel tous les autres nœuds passeront pour s'insérer.
Chaque nœud invité d'identifiant I émet une requête d'insertion de nœud qui est routée jusqu'au nœud R responsable de l'espace [a, a'] x [b, b'] contenant I. Le partage de l'espace des coordonnées entre I et R se fait alors de la manière suivante.

1. On choisit d'abord l'intervalle le plus grand (max((a'-a);(b'-b))) ; si les deux intervalles sont de taille égale, on choisit l'intervalle [a, a'] par défaut.
2. On divise l'intervalle choisi en deux et on constitue deux sous-espaces issus de cette division.
3. R conserve la responsabilité du sous-espace qui contient son identifiant.
4. I acquiert la responsabilité de l'autre sous-espace. Si l'identifiant I n'appartient pas à ce sous-espace, I tire aléatoirement un nouvel identifiant dans ce sous-espace.

Etape 2 : Insertion de données
Le coordinateur tire aléatoirement N*10 coordonnées (a, b), et la valeur de la donnée à insérer est calculée par addition de a et b.
Le couple clé-valeur inséré dans la DHT est donc : ((a, b), (a+b)).
Le coordinateur insère ces données une par une, en émettant une requête d'insertion de donnée auprès du nœud de bootstrap et en attendant un acquittement du nœud qui stocke la donnée avant de demander une nouvelle insertion.
Le coordinateur conserve en mémoire les coordonnées des 5 premières insertions et celles des 5 dernières insertions.

Etape 3 : Recherche de données
Lorsque toutes les données ont été insérées, le coordinateur va émettre une par une des requêtes de recherche de donnée associées aux 10 coordonnées qu'il a conservées en mémoire.
Le coordinateur va également initier 4 recherches pour des clés distinctes tirées au hasard.

Etape 4 (facultative) : Suppression de nœud
Le coordinateur tire aléatoirement une clé. Le nœud responsable de l'espace contenant cette clé doit alors être retiré de l'overlay. Lorsque cette suppression est effective, le coordinateur lance une recherche de donnée associée à la même clé pour vérifier qu'un des nœuds restants a bien endossé la responsabilité de l'espace.

Références
- “A Scalable Content-Addressable Network”, thèse de Sylvia Ratnasamy, 2002.