TPs : Chord

M1 CSR

Le but de cette série de TPs est d'implémenter un système de type Chord.
On procédera par étapes:

1. Mise en place du cercle initial
   - Ajout d'un noeud : mise à jours uniquement du précédent et suivant
   - Fonction `value = get(key)` et `update(key, value)`
2. Gestion du véritable voisinage lors de l'ajout
   - Tenir compte du fait qu'on ne connaît pas à priori le nombre final de noeuds
3. Gérer le départ des noeuds
4. Tenir compte de la réplication (tester avec k=2)

Dans tous les cas, on pourra obtenir le nombre de communication moyen par type de requêtes (get, update) ainsi que pour la gestion de l'infrastructure (communications dues à l'arrivée/au départ des noeuds). Il y a donc 4 valeurs moyenne qu'il est nécessaire de pouvoir récupérer.

D'un point de vue code, il s'agira de faire deux programmes. Il peut s'agir d'un seul programme tant que toutes les communications entre objets se font à travers d'XML-RPS. Un programme lançant les différents objets chord, un programme faisant les requêtes. Toutes les communications entre objets seront faites par XML-RPC. En utilisant un argument sur la ligne de commande on utilisera un voisinage simple (étape 1) ou un voisinage complet (étape 2).

Le code client pourra demander à distance l'ajout de noeuds (dont la clé est prise au hasard) ou la fin de noeuds.

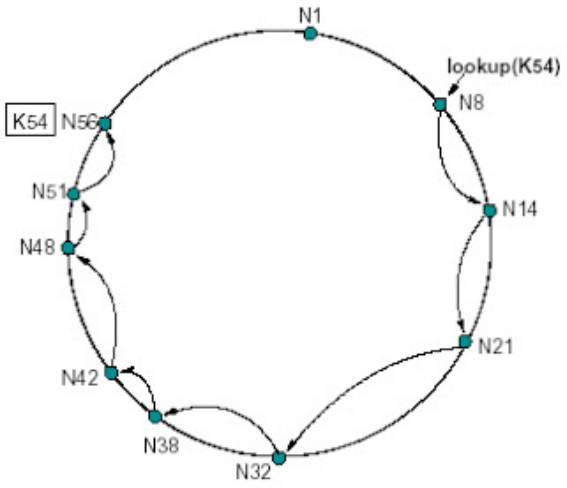
Le code client fera la démonstration des possibilités des objets Chord et donnera des statistiques du nombre de communications en fonction de la taille du réseau.

Remarques

On gèrera les couples `<clé,valeur>` par une table de hachage, les clés seront des `int` comme les valeurs. Les clés seront entre 0 et 65535.

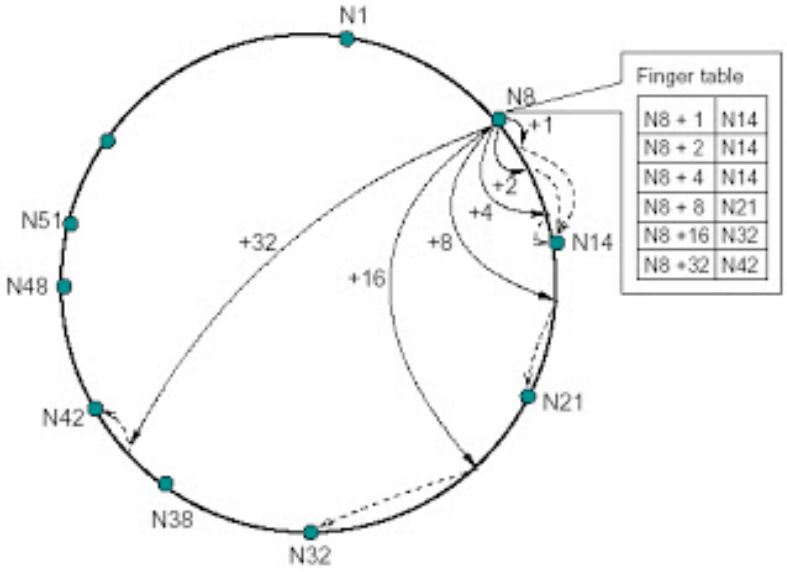
Le noeud responsable de la valeur i sera le noeud dont la clé est égale ou plus proche supérieur (modulo 65536).

Pour la première partie, on aura le type de routage suivant:



Pour la seconde partie, on rappel que chaque table de voisinage est constituée comme suit: Le noeud i a comme voisin

- $i + 65536/2 \ \% \ 65536$
- $i + 65536/4 \ \% \ 65536$
- $i + 65536/8 \ \% \ 65536$
-
-
- $i + 1 \ \% \ 65536$



Si une clé de noeud n'est pas trouvée, on utilisera le noeud responsable pour cette clé à la place. On fera attention à la gestion des doublons.

Le rendu du code aura lieu à la fin du dernier TP au mail suivant: **dacosta@irit.fr** avec le sujet **[TPCSR] nom prenom**