LIU Zhuying 11306849

FLORES Isabelle 11102272

**Rapport Projet Content Addressable Network, Mif32 Parallélisme**

Pour **stocker un espace**, nous avons simplement une structure qui contient quatre entiers:

typedef struct espace {

int xdebut;

int xfin;

int ydebut;

int yfin;

}Espace;

Pour **stocker les voisins et les données**, nous avons hésité de passer en C++ pour utiliser la librairie STL. Finalement nous avons trouvé GSList qui est une librairie Open-Source en c. Celle-ci est une liste générique et simplement chaînée.

typedef struct voisins{

GSList \*voisinsHaut;

GSList \*voisinsBas;

GSList \*voisinsGauche;

GSList \*voisinsDroite;

}Voisins;

typedef struct donnee{

int pos[2];

int valeur;

} Donnee;

typedef struct donnees {

GSList \*contenu;

}Donnees;

**Insertion des nœuds:**

Le processus 0 envoie l’autorisation d’insertion à chaque processus et attendre la fin de l’insertion, ensuite il envoie l'autorisation d’insertion au prochain processus.

Chaque processus attend l’invitation d’insertion et envoyer aux processus déjà inséré son identifiant, et le processus qui a la responsabilité de cet identifiant lui attribue un espace. Une fois un processus a eu son espace distribué, il annonce au processus 0 la fin d’insertion.

**Connaissance des voisins:**

Quand tous les processus sont insérés, chaque processus envoie son espace à tout le monde. Et chacun calcul si l’espace reçu est son voisin.

**Calcul des voisins:**

Pour le nœud rose, un de ses voisins en haut a trois situations.

*rose.xdebut <= bleu.xdebut < rose.xfin*

*rose.xdebut < bleu.xfin <= rose.xfin*

*rose.xdebut >= bleu.xdebut && rose.xfin <= bleu.xfin*

Dans ces trois cas*, bleu.yfin == rose.ydebut – 1*.

**Insertion des données:**

Le processus 0 envoie chaque coordonnée à un processus aléatoire, et attendre la réponse du processus qui a la responsabilité de cette coordonnée, ensuite il lui envoie la donnée.

Chaque processus attend un message de deux entiers de n’importe quelle source. Si ces deux sont positifs, il s’agit d’une coordonnée. Le processus vérifie si cette coordonnée est dans son espace, si oui il répond au processus 0, sinon il trouve un voisin de la bonne direction et envoyer cette coordonnée à ce voisin. Si ces deux entiers sont (-1,-1), cela indique la fin d’insertion des données.

**La recherche des données est similaire à l’insertion des données.**

**Suppression d’un nœud:**

Le nœud qui va partir demande à ses voisins de lui envoyer leurs espaces. Ce nœud décide à quel(s) voisin(s) donner ses données et son espace. Les voisins reçoivent ses données et ses voisins, et son espace en partie.

**Trouver le(s) voisin(s) remplaçant(s) avant la suppression:**