Systèmes Distribués
Master M1 Info
2014-2015

TP
Distributed Bag of Tasks
(Ensemble de tâches)
Assistants et Coordinateur de Tâches

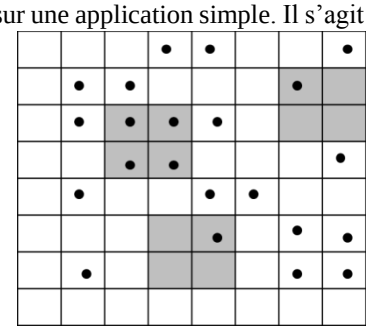
Nous allons étudier et illustrer un modèle d'interaction entre processus appelé "Bag of Distributed Tasks". Ce modèle organise une application ou un système en un ensemble distribué de tâches qui peuvent être indépendantes ou non. Il existe d'autres modèles d'interaction de processus parallèles ou distribués. Parmi ces modèles, nous pouvons citer : le producteur-consommateur, les lecteurs-rédacteurs, le client-serveur, et les P2P (peer to peer).

L'idée centrale du paradigme "Bag of Tasks" est de définir a) un ensemble de processus-assistant (worker process) qui partagent ou collaborent à l'exécution d'un ensemble ou bag de taches et b) un processus coordinateur chargé de décomposer une application en tâches et d'affecter les taches aux processus-assistant. Le processus coordinateur doit également synchroniser les exécutions des processus-assistant, collecter ou regrouper les résultats partiels des exécutions de tâches pour produire le résultat final et enfin déterminer la terminaison de l'ensemble des tâches. Un avantage du modèle "Bag of Tasks" est de faciliter la variation nombre de processus utilisés pour l'exécution d'une application. En effet, chaque processus assistant peut de façon itérative obtenir du coordinateur la tâche suivante à exécuter. Ceci continue jusqu'à épuisement de tâches.

Une variante du modèle "Bag of Tasks" peut être mise en œuvre sur un ensemble de taches dépendantes. Dans ce cas particulier, les données sont divisées en groupes ou blocs de données traités par les processus-assistant. A chaque étape du traitement, un processus assistant calcule et met à jour son groupe de données en recevant des données d'un ou plusieurs processus voisins. Par exemple, si les processus assistant traitent des données représentées par un vecteur ou une matrice de valeurs, le fonctionnement générique d'un processus assistant peut être décrit par le pseudo code suivant :

```
Processus-assistant Pi,  i=  0, 1, 2 …
Begin
  Déclaration des variables locales (un bloc du vecteur ou de la matrice de données)
  Initialisation de tache locale
  While (PAS-FINI) {
        Envoi  de  données à un plusieurs (processus) voisins
        Réception de données des voisins
        Calcul et mise à jour de données locales}
End
```

Nous allons illustrer le fonctionnement du paradigme "Bag of tasks " sur une application simple. Il s'agit du "JEU DE LA VIE" qui simule une automate cellulaire. L'application est représentée par un espace à deux dimensions décomposé en cellules. L'espace est une matrice nXn contenant n^2 cellules. Une cellule est dite vivante si elle contient un organisme biologique vivant ou morte si elle est vide. Chaque cellule interne a 8 voisins qui sont les cellules : au-dessus, en dessous, à gauche, à droite et les cellules voisines selon les 4 diagonales. Une cellule dans un coin de l'espace a trois voisins alors que celles situées sur les bords de l'espace ont chacune cinq voisins. La figure ci-contre montre un espace 8X8 avec des cellules vivantes et mortes. La division

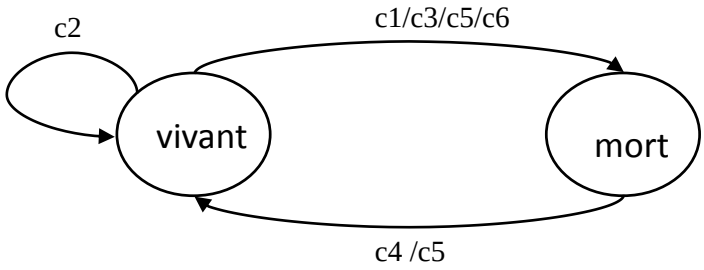


de l'espace en groupes de données est faite par bloc carré de cellules. Dans l'exemple de la figure, chaque bloc, en grisé, est composé de 4 cellules. La mise à jour de chaque bloc de 4 cellules (disparition ou apparition d'organisme vivant constitue une tâche.

Le jeu de la vie se déroule par vagues successives. Chaque vague correspond à une génération et est utilisée pour déterminer l'état suivant du système de cellules : une partie des cellules meurent alors que d'autres survivent et sont présentes à la génération suivante. L'ensemble des valeurs des cellules représente un état du système. La division du travail en vague est gérée par le processus coordinateur qui détermine le début et la fin de chaque vague.

A chaque itération l'ensemble des processus disponibles (coordinateur et assistants) recalcule les valeurs des différentes cellules. La mise à jour de la valeur d'une cellule peut être décrite par un diagramme de transition d'états (vivant ou mort) et se déroule de la façon suivante, voir figure ci-dessous. D'abord l'espace est initialisé en précisant les cellules vivantes par allocation d'organismes (valeur 1, T ou tout autre type de valeur), et les cellules mortes (valeur 0, F ou tout autre type de valeur). Ensuite, à chaque vague de traitement ou calcul, la valeur d'une cellule est mise à jour en fonction de l'état de la cellule et des états des voisins selon les règles ou conditions suivantes :

- (c1) Une cellule vivante ayant 0 ou 1 voisin vivant meurt de solitude à la génération suivante
- (c2) Une cellule vivante ayant 2 ou 3 voisins vivants survit et est présente à la génération suivante
- (c3) Une cellule vivante ayant 4 ou plus voisins vivants meurt à la génération suivante, victime de surpopulation
- (c4) Une cellule morte avec 3 voisins vivants est ressuscitée à la génération suivante
- (c5) Un processus aléatoire permet d'injecter de la vie dans l'espace par génération spontanée d'organisme dans une cellule choisie au hasard : a) s'il s'agit d'une cellule morte, alors elle devient vivante et contient l'organisme nouvellement créé; b) s'il s'agit d'une cellule vivante, elle devient morte car deux organismes ne peuvent pas partager la même cellule.
- (c6) un processus aléatoire d'épidémie virale peut se réveiller et à partir d'une cellule c(x, y) tuer toutes cellules (c(x+1, y+1), c(x+2, y+2), c(x+3, y+3) etc.) situées sur diagonale issue de la cellule c(x,y).



Le diagramme de transition d'état ci-contre présente les évolutions d'états d'une cellule du jeu de la vie. Les transitions d'état sont étiquetées par les conditions déclenchant la transition d'un état vers un autre.

A faire : Vous devez proposer une solution pour le jeu de la vie sur une grille 16X16 constituée de 256 cellules divisées en 64 blocs de 4 cellules chacun. Votre solution doit, dans un premier temps, utiliser 16 processus-assistant et un processus coordinateur qui répartit les tâches aux assistants et qui réalise aussi les processus aléatoires (injection de vie et épidémie virale) décrits ci-dessus. Ensuite le nombre de processus-assistant est varié de 16 à 32, puis 64 par pas de 4. L'initialisation du jeu se fera par le processus coordinateur en lisant un fichier "fic-init" qui définit les paramètres suivants:

-n : la dimension de la grille nXn (n=16)

-b : la taille en cellules d'un bloc (b=4)

-p : le nombre de processus-assistant (p=16)

-la liste des cellules vivantes initiales de la grille, précisées par leurs coordonnées.

A chaque itération, les échanges entre processus (assistant-assistant ou coordinateur-assistant) se font à l'aide de canal de communication **send-Xchange (i, j, valeurs)** qui permet au processus i d'envoyer des données au processus j et **recv-Xchange (i, j, valeurs)** qui permet à i de recevoir des données de j.

On dispose donc d'un tableau ou ensemble de canaux d'échange qui seront implémentés suivant le modèle client-serveur si les processus sont sur des machines distantes ou suivant d'autres modèles (tube, tube nommés etc.) si les différents processus sont des processus ou threads situés sur la même machine.

Délivrables :

- Les échanges de données ou messages représente une part importante du projet. Pour les séances de TP de la semaine du 17/11, vous devez préparer un rapport intermédiaire qui précise a) les différents types d'échanges d message et les protocoles (règles d'interactions) utilisés, b) les conditions d'initialisation et de terminaison de chaque vague de calcul, c) les conditions de terminaison du jeu, d) l'architecture générale de votre projet, et d) l'affichage des résultats intermédiaires (qui affiche quoi?)

- Vous devez rendre un rapport final comprenant l'analyse et les détails de l'architecture de votre solution, et les codes source de votre implémentation.
    - Votre rapport doit montrer les évolutions de la grille suivant plusieurs itérations ou générations du jeu de vie.
    - Enfin, vous devez discuter de la performance (temps d'exécution) de votre solution en variant le nombre de processus-assistant pour p=16, 32, 64.
    - Le rapport final doit être rendu par email le 20/12/2014 à minuit.
    - Des séances de démonstration de votre solution seront prévues dans la dernière semaine de cours de décembre.
- La note du projet sera basée sur le rapport intermédiaire et sur les présentations et discussion de votre projet en séances de TP