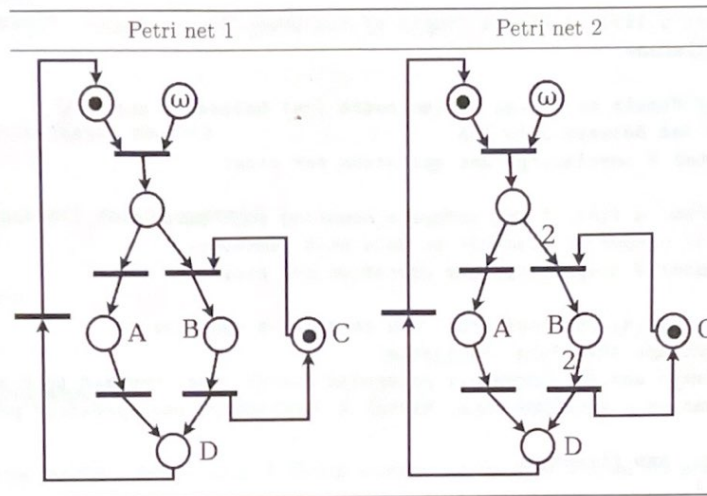


M1 - Parallélisme - Examen - Session 1

Documents non autorisés, durée 3h

Sujet Petri (répondre sur cette page)



1. Pour le réseau de Petri 1 donnez tous les déroulés possibles pour les actions A, B, C et D quand un message commence à être traité

2. Pour le réseau de Petri 2, donnez le graphe de marquage

3. Pour le réseau de Petri 2, que pouvez-vous dire sur l'action B ?

Sujet MPI (répondre sur une copie MPI)

Exercice 1

On veut faire un équilibrage (equalizer) de la luminance d'une image par la méthode des histogrammes.

L'algorithme séquentiel serait le suivant:

```
# loads an image into a list of Pixels (tuple of Red/Green/Blue values)
data = load_image(filename)

# Converts a list of Pixels to a list of luminance (int between 0 and 255)
# returns a list of int between 0 to 255
lumi = luminance(data) # complexity: one operation per pixel

# creates an histogram: a list of 256 integers counting each value :
# histo[5] counts the number of elements in data with luminance 5
histo = histogram(data) # complexity: one operation per pixel

# equalizes the image using the histogram (can be done on any list
# of pixels, but must use the global histogram)
# returns the new image and the number of saturated pixels (i.e. rounded to 0 or to 255).
new_image, nb_saturation = equalize(data, histo) # complexity: one operation per pixel

save_image(new_image, new_filename)
print(nb_saturation)
```

On propose de créer un algorithme parallèle utilisant les opérations collectives MPI. On considère que le processus de rang 0 se chargera du chargement, de la sauvegarde de l'image et de l'affichage du nombre de pixels saturés.

- 1) Faire le schéma de communication de données d'une version parallèle utilisant 4 processus
- 2) Ecrire le code MPI générique (ie pour un nombre quelconque de processus).
- 3) Exprimer la complexité nécessaire (en nombre d'opérations et en quantité de communications) pour traiter les données en fonction de la taille de l'image (en nombre de pixels) et du nombre de processus (hors fonction load_image and save_image).

On considère avoir les fonctions :

```
split(list_of_elements, nb)
# splits the list in argument in nb sub-lists

add_histogram(histo1, histo2)
# adds two histograms (i.e. adds element by element) and return an histogram

add_histograms([histo1, histo2, ..., histon])
# adds a list of histograms and returns one histogram
```

Exercice 2

On a le code suivant et on suppose que le nombre de processus est 4

```
from mpi4py import MPI

comm = MPI.COMM_WORLD
rank = comm.Get_rank()
size = comm.Get_size()

data = [0,2,4,8]

l_data = comm.scatter(data, root=1)
l_data2 = l_data / 2

resultat = comm.gather(l_data2, root=2)

if rank == 0:
    print(resultat)
```

1. Qu'affiche ce programme ?

2. Donnez l'état des variables *data*, *l_data*, *l_data2*, *resultat* à la fin pour chacun des processus dans le tableau suivant:

	rank = 0	rank = 1	rank = 2	rank = 3
data				
l_data				
l_data2				
resultat				

Moniteur de Hoare : gestion des appels d'un professionnel

On souhaite évaluer différentes politiques de gestion des déplacements d'un service de retrait à domicile de colis pour répondre aux appels émis par ses clients et les déposer au dépôt central localisé locDépôt. Cette évaluation se fait sur la base de la distance parcourue par les professionnels et du temps d'attente d'un client.

Un processus client qui souhaite une intervention d'un professionnel a le comportement suivant :

```
Processus client(Identification idClient,
                  Localisation locClient){
    Demander au service de gestion des appels l'intervention d'un
    professionnel pour la localisation locClient
    -> identification du professionnel idProfessionnel + temps
        attendu;
    Le professionnel idProfessionnel est donc là : il prend le colis
}
```

Un processus professionnel prend en compte les demandes d'intervention d'un clients selon l'algorithme suivant :

```
Processus professionnel(Identification idProfessionnel) {
    Demander au service de gestion des appels un appel à traiter
    -> identification du client idClient et localisation locClient
        du client à servir
    Rouler pour aller chez le client
    Gérer l'appel du client;
    Indiquer au service de gestion des appels que l'appel a été traité
}
```

Le traitement d'un appel sera considéré immédiat (durée = 0).
Lorsqu'il n'y a plus d'appel, un professionnel revient au dépôt.

Le service de gestion des appels (SGA) sera implémenté en utilisant un moniteur de Hoare qui permettra de synchroniser les professionnels et les clients.

Ce service devra être capable de fournir à tout instant le nombre d'appels satisfaits, le temps cumulé des attentes des clients traités et la distance cumulée parcourue par les professionnels. L'opération qui permet d'obtenir ces informations devra avoir pour prototype :

```
void statistiques( unsigned *nbClientsTraités,
                  unsigned long *tempsCumuléAttentes,
                  float *distanceTotaleParcourue);
```

On supposera fournies les fonctions :

```
void roulerDeA(Localisation loc1, Localisation loc2);
    qui simule un déplacement de loc1 à loc2 par un professionnel.

float distance(Localisation loc1, Localisation loc2);
    qui estime la distance pour aller de loc1 à loc2.
```

Le moniteur comportera aussi la méthode ticTac, qui sera automatiquement appelée à chaque seconde par un processus horloge. Elle permettra d'actualiser l'heure du SGA.

```
Pocessus horloge() {
    while (true) {
        // attendre une seconde;
        SGA.ticTac();
    }
}
```

Plusieurs versions correspondant à différentes politiques de gestion des appels seront programmées.

1 Spécification du moniteur

Donnez la spécification du moniteur SGA (service de gestion des appels)

2 Codes d'un processus professionnel et d'un processus client

En utilisant la spécification du moniteur SGA et les fonctions fournies, donnez, en pseudo C, le code d'un processus professionnel et d'un processus client.

3 Variante « un seul professionnel »

On suppose que le SGA ne travaille que pour un seul professionnel.

Pour chacune des politiques définies :

3.1 Donnez les valeurs des statistiques évaluées par le SGA pour le scénario suivant :

Distances						
	P 1	C1	C2	C 3	C 4	C 5
P1	0					
C1	5	0				
C2	5	2	0			
C3	2	3.6	3	0		
C4	2	3	3.6	2	0	
C5	8	3.2	3.2	6	6	0

Ordre des événements

Temps

- | | |
|----|----------------------------------|
| 1 | C3 |
| 2 | C1 |
| 3 | Demande d'appel du professionnel |
| 4 | C4 |
| 5 | Fin de traitement de l'appel |
| 6 | C5 |
| 7 | Demande d'appel du professionnel |
| 8 | Fin de traitement de l'appel |
| 9 | C2 |
| 10 | Demande d'appel du professionnel |
| 11 | Fin de traitement de l'appel |
| 12 | Demande d'appel du professionnel |
| 13 | Fin de traitement de l'appel |
| 14 | Demande d'appel du professionnel |
| 15 | Fin de traitement de l'appel |

Au début de l'application, le professionnel est au Dépôt.

3.2 Précisez les conditions de blocage et de réveil du processus professionnel et d'un processus client.

3.3 En déduire les variables d'état et « condition » gérées dans le moniteur. On précisera s'il s'agit de condition avec ou sans priorité

3.4 Donner le code du moniteur.

Politique 1 : FIFO

On sert les appels des processus clients selon une politique FIFO (premier arrivé, premier servi).

Politique 2 : politique plus proche de la position actuelle

On sert les appels des processus clients en minimisant le déplacement du professionnel, relativement à sa localisation actuelle.

Politique 3 : politique « Scan »

Le professionnel veut servir les appels qui l'éloignent de plus en plus du dépôt, puis revenir vers le dépôt tout en répondant aux appels qui le rapprochent du dépôt. Cette stratégie est renouvelée en permanence.

4 Variante « plusieurs professionnels »

Le SGA gère à présent les appels pour le compte de NB_PROF professionnels qui peuvent répondre indifféremment aux appels de tous les clients.

Proposez les modifications à apporter au moniteur SGA, pour chacune des variantes, puis codez le moniteur pour chacune de ces variantes. Pour rappel, l'objectif est de minimiser les déplacements des professionnels et de réduire les temps d'attente des clients.