Examen SE 205

avril 2019

1h30 - Barème indicatif (complété par QCM et TP), **sans document**
Les téléphones portables doivent être éteints et rangés dans vos sacs.
Les calculatrices sont interdites.

Les différentes parties de l'examen sont indépendantes.

Il sera tenu compte **de la présentation et de la clarté** dans la rédaction.
Seules les réponses **précises et justifiées** seront considérées

REPONDRE AUX QUESTIONS SUIVANTES SUR UNE FEUILLE SEPAREE
N'OUBLIEZ PAS D'Y INSCRIRE VOS NOM ET PRENOM.
VOUS POUVEZ REPONDRE EN ANGLAIS

1 Modèle d'exécution C/POSIX (2 points)

On considère en C/POSIX la fonction pthread_cond_wait. Décrivez le fonctionnement de cette fonction et notamment ses paramètres. Rappelez également les états et transitions intermédiaires suivis par un thread exécutant une telle fonction. Seules les réponses **précises et justifiées** seront considérées.

2 Modèle d'exécution alternatif (3 points)

Nous souhaitons concevoir un système qui reçoit et traite des requêtes en tirant profit d'une architecture multi-cœurs. Nous utilisons un patron Executor comme celui vu en cours et en TP. On dispose d'un quadri-cœurs (4) et souhaitons que le système accepte (i) de traiter en parallèle et sans attente au minimum 4 requêtes concurrentes, (ii) de traiter en parallèle au plus 8 requêtes simultanées sans en mettre en attente et (iii) de détruire les threads créées, s'il y en a plus de 4 disponibles, après 15s d'inactivité. Détaillez la structure (notamment les patrons intermédiaires) et le fonctionnement du patron Executor. Précisez la configuration de ses principaux paramètres pour le cas décrit.

3 Modèle d'exécution Java (2 points)

Donnez au moins trois manières de mettre en œuvre des threads afin de permettre l'exécution de tâches logiques de manière concurrente.

REPONDRE AUX QUESTIONS SUIVANTES SUR UNE FEUILLE SEPAREE
N'OUBLIEZ PAS D'Y INSCRIRE VOS NOM ET PRENOM.
VOUS POUVEZ REPONDRE EN FRANCAIS

4 Dépendances

4.1 Loi d'Amdahl (1 point)

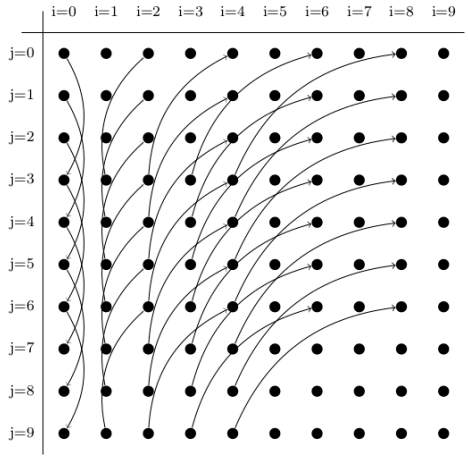
Expliquez les implications de la loi d'Amdahl pour la programmation parallèle.

4.2 Iteration Space Graphs (1 point)

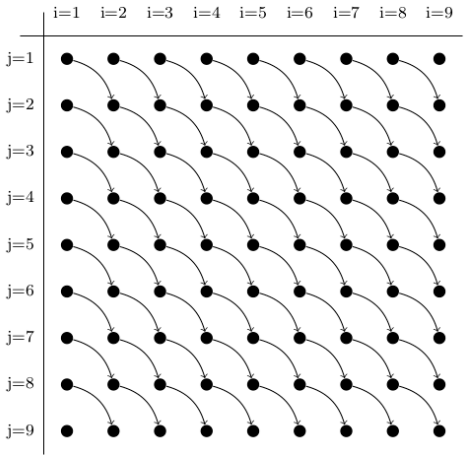
Quel iteration space graph correspond au code ci-dessous ? Justifiez votre choix.

```
1 for(i = 0; i < N; ++i) {
2   for(j = 0; j < N; ++j) {
3     A[i][j] = A[i][j+3] + B[i][j];
4   }
5 }
```

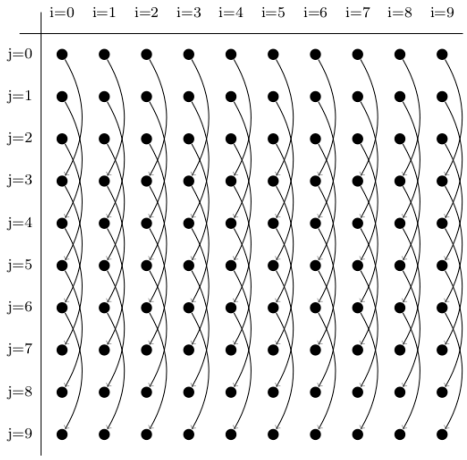
a)



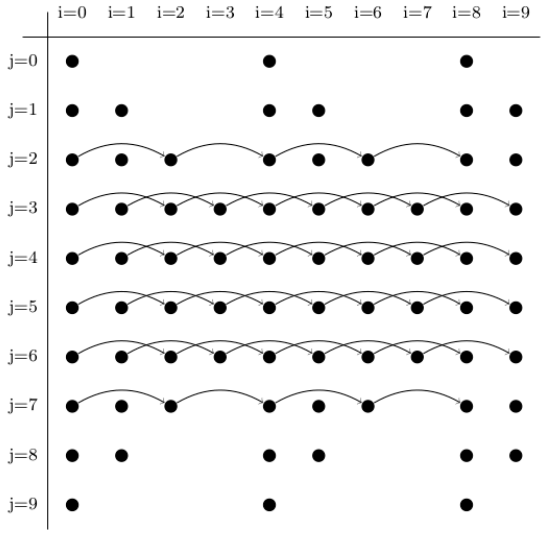
b)



c)



d)



5 Mémoire Partagée

Répondez aux questions suivantes en considérant le code ci-dessous, qui consiste en trois opérations atomiques, exécutées par deux processus légers (threads) :

```
          Thread α                          Thread β
1  x = 1                          1  y = 1
2  if (y == 0) printf("zero");    2  if (x == 0) printf("zero");
3  printf("done");                3  printf("done");
```

5.1 Consistance séquentielle (1 point)

Supposez un système qui garanti de la consistance séquentielle. Les quels, parmi les résultats et entrelacements indiqués ci-dessous, sont observables, si les deux taches exécutent touts leurs opérations. Justifiez votre réponse.

a) `donedone`
(α1, β1, α2, β2, α3, β3)

b) `donedonezero`
(α1, β1, α2, α3, β3, β2)

c) `donezerozerodone`
(α1, α3, α2, β2, β1, β3)

d) `zerodonezerodone`
(α1, α2, α3, β1, β2, β3)

e) `zerodonedone`
(α1, α2, α3, β1, β2, β3)

f) `done`
(α1, β1, α2, β2, α3)

5.2 Consistance faible (2 point)

En cours et dans un TP vous avez vous que ce code, sur des machines réels, permet d'observer le résultat `zerozerodonedone`. Les machines réels ne garantissent seulement un modelé de consistance faible (Total Store Order). Donnez un entrelacement qui explique cette observation.

5.3 Opérations atomiques (1 point)

En TP vous avec vu l'opération `atomic_thread_fence()`. Donnez un exemple où cette opération est nécessaire afin d'éliminer des comportements inattendus d'un programme C. Quel est l'effet de cette opération ?

6 Acteurs

6.1 Actions (1 point)

Quel sont les trois action (principaux) qui un acteur peut effectuer ? Expliquez les.

6.2 Location Transparency (1 point)

Expliquez le concept de *Location Transparency* qui est souvent associé aux acteurs.

6.3 Acteurs dans la bibliothèque Akka (1 point)

Quel est la différence entre un objet de type UntypedActeur et un objet de type ActorRef dans la bibliothèque Akka ?