ESISAR

NE441 – Programmation répartie

TDM numéro 5

Table des matières

1 Objectifs du TDM

Ce TDM va permettre de découvrir la gestion des threads, en Java.

2 Exercice 1 : Calcul de la constante PI en mono thread

Réalisez un programme qui calcule la constante PI de façon classique , avec la formule suivante, avec N = 5 000 000 000.

$$\frac{1}{1} - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \cdots + \frac{(-1)^k}{2k+1} + \cdots = \sum_{k=0}^{\infty} \frac{(-1)^k}{2k+1} = \frac{\pi}{4}$$

Afficher le temps de calcul. Visualisez la consommation de CPU de la machine. Pour cela, il existe divers outils :

- top
- vmstat
- htop

L'outil **htop** est conseillé car il permet de visualiser la consommation CPU sur chaque coeur de la machine. Pour installer cet outil , il faut lancer la commande suivante en étant root :

```
apt-get install htop
```

Vous pouvez aussi visualiser les caractéristiques matérielles de votre machine (nombre de CPU, mémoire, …).

Pour cela , installer le paquet hwloc avec la commande suivante

```
apt-get install hwloc
```

Ensuite, lancez les deux commandes suivantes

```
lscpu
```
et
```
lstopo
```

3 Exercice 2 : Calcul de la constante PI en multi thread

Faites le même calcul mais avec un programme avec 2 threads.

Afficher le temps de calcul. Visualisez la consommation de CPU de la machine.

Faites de même avec 4 threads, 8 threads, 16 threads, 32 threads, 64 threads , 512 threads, 2048 threads.

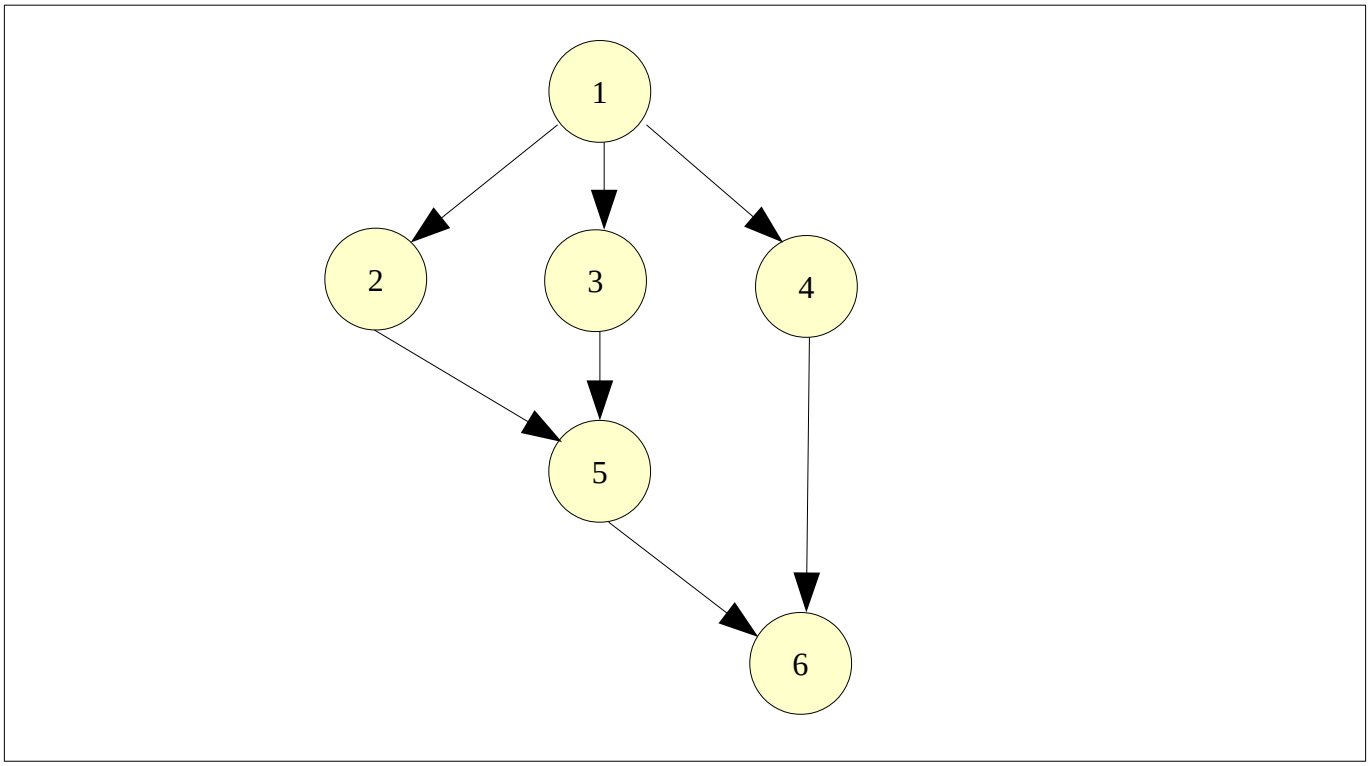
Pour chaque cas, afficher le temps de calcul et visualisez la consommation de CPU de la machine.

Tracer la courbe du temps de calcul en fonction du nombre de thread.

Que pouvez vous en conclure ?

4 Exercice 3 : Séquencement de threads (version simple)

Votre programme doit réaliser 6 tâches qui sont dépendantes entre elles. Les dépendances entre les taches sont données ci dessous :



Explication :

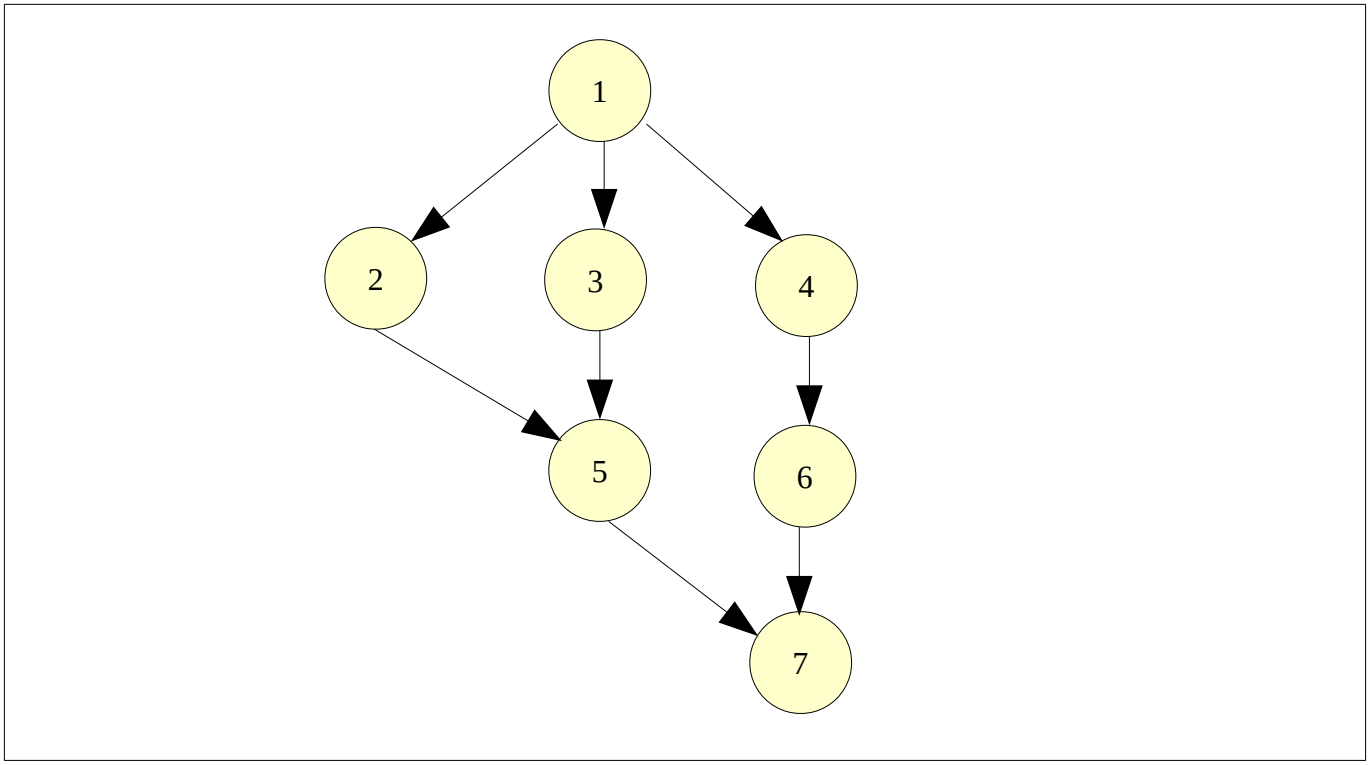
- la tâche 2 peut se faire si la tâche 1 est terminée
- la tâche 5 peut se faire si les tâche 2 et 3 sont terminées
- …

Faites un programme avec 6 threads, le thread 1 executera la tâche 1,le thread 2 la tâche, .. en respectant le séquencement des tâches.

Pour le contenu de la tâche, vous ferez simplement une attente d'un temps aléatoire, en indiquant le début de l'attente et la fin de l'attente.

5 Exercice 4 : Séquencement de threads (version complexe)

Maintenant, votre programme doit réaliser 7 tâches avec les dépendances suivantes:



Pourquoi est ce plus compliqué ?