



Systemes Concurrents

TP REGULATION

Deuxième Année, Systemes Logiciels

YOUNES SAOUDI

2020-2021

Calcul du Maximum d'un Tableau

1. Schéma de parallélisation le plus naturel/efficace :

Puisqu'il s'agit d'un problème régulier, le schéma le plus naturel et efficace serait celui du Pool de Threads.

2. Comparaison :

```
12 processeurs disponibles pour la JVM
Execution avec :
  nbEssais = 1
  nbTaches = 1000
  tailleTroncon = 20000
  nbOuvriersPool = 12
-----
Essai [0] : maximum = 49, duree (mono) 63850µs
-----
Essai [0] : maximum = 49, duree (pool) 50238µs
-----
Essai [0] : maximum = 49, duree (FJ) 99731471µs
-----
```

Figure 1 – Recherche du max d'un tableau de 10 millions d'éléments

Comme prévu, le schéma de parallélisation le plus efficace est celui du Pool de Threads. Ce problème, quoique élémentaire, peut être décomposé en sous-problèmes de taille équivalente, ce qui garantit une bonne répartition des tâches entre ouvriers.

Voir Implémentation MaxTab.java

Tri Fusion

1. Comparaison

```
12 processeurs disponibles pour la JVM
Essai [0] : duree (mono) 102159µs
-----
Essai [0] : duree (PF) 782782µs
-----
Essai [0] : duree (FJ) 60819µs
-----
```

Figure 2 – Tri fusion d'un tableau d'un million d'éléments

Le schéma de parallélisation le plus efficace cette fois-ci est celui du Pool Fork/Join. Effectivement, ce schéma est parfait pour les tâches qui peuvent être récursivement décomposées en tâches plus petites, ce qui convient au tri de fusion même s'il est régulier.

Ce schéma est surtout beaucoup plus facile à concevoir que la version séquentielle ou le schéma de Pool de Threads.

Voir Implémentation `TF.java`