

TP Systèmes concurrents – Concurrency et cohérence

2SN

ENSEEIH
Département Sciences du Numérique

4 octobre 2020



Exercice 1 : efficacité de la parallélisation

- A priori : réduction proportionnelle au nb de threads
- Différences

Exemple d'exécution de IncrMes :

```
[ps3:Corrigés/TP1/TP1A - Concurrence et cohérence] philippe% java IncrMes 3
Durée multi (ms) : 555
Durée mono (ms) : 1570
Surcoût (multi*nbactivités/mono) : 1.0605095541401275

[ps3:Corrigés/TP1/TP1A - Concurrence et cohérence] philippe% java IncrMes 5
Durée multi (ms) : 576
Durée mono (ms) : 2612
Surcoût (multi*nbactivités/mono) : 1.1026033690658499

[ps3:Corrigés/TP1/TP1A - Concurrence et cohérence] philippe% java IncrMes 10
Durée multi (ms) : 1112
Durée mono (ms) : 5232
Surcoût (multi*nbactivités/mono) : 2.125382262996942

[ps3:Corrigés/TP1/TP1A - Concurrence et cohérence] philippe% java IncrMes 15
Durée multi (ms) : 1828
Durée mono (ms) : 7843
Surcoût (multi*nbactivités/mono) : 3.4961111819456843

[ps3:Corrigés/TP1/TP1A - Concurrence et cohérence] philippe% java IncrMes 20
Durée multi (ms) : 2373
Durée mono (ms) : 10455
Surcoût (multi*nbactivités/mono) : 4.539454806312769


[ps3:Corrigés/TP1/TP1A - Concurrence et cohérence] philippe% java IncrMes 25
Durée multi (ms) : 2996
Durée mono (ms) : 13059
Surcoût (multi*nbactivités/mono) : 5.735508078719657

[ps3:Corrigés/TP1/TP1A - Concurrence et cohérence] philippe% java IncrMes 30
Durée multi (ms) : 3589
Durée mono (ms) : 15660
Surcoût (multi*nbactivités/mono) : 6.875478927203065
```

- capacité matérielle
- coût de l'ordonnancement
- Evaluation du surcoût : pas évidente

Note : `Runtime.getRuntime().availableProcessors()`

Exercice 2 : coût de la cohérence

- Sans préemption \Rightarrow valeurs en séquence 
- Cohérence – Exemple d'exécution : 3 activités, pauses de 100ms

```
[ps3:Corrigés/TP1/TP1A - Concurrency et cohérence] philippe% java IncrDemoSync  
Durée exécution mono : 311789
```

```
Thread t0 part de : 0  
Thread t1 part de : 0  
Thread t2 part de : 913  
Thread t1 finit à : 1233349588  
Thread t0 finit à : 1233349588  
Thread t2 finit à : 1233349588  
Durée exécution non synchronisée (ms): 104094
```

```
Thread t0 part de : 0  
Thread t1 part de : 0  
Thread t2 part de : 688  
Thread t0 finit à : 1000000228  
Thread t2 finit à : 2000000039  
Thread t1 finit à : 3000000000  
Durée exécution synchronisée (ms): 311940
```

```
Thread t0 part de : 0  
Thread t1 part de : 0  
Thread t2 part de : 217  
Thread t1 finit à : 2918391445  
Thread t2 finit à : 2935001685  
Thread t0 finit à : 3000000000  
Durée exécution synchronisée grain très fin (ms): 171699
```

```
Thread t0 part de : 0  
Thread t1 part de : 0  
Thread t2 part de : 1499  
Thread t0 finit à : 3000000000  
Thread t2 finit à : 3000000000  
Thread t1 finit à : 3000000000  
Durée exécution synchronisée à grain fin (ms): 104185
```

```
Thread t0 part de : 0  
Thread t2 part de : 0  
Thread t1 part de : 0  
Thread t0 finit à : 2985000000  
Thread t2 finit à : 2999000000  
Thread t1 finit à : 3000000000  
Durée exécution synchronisée atomique (ms): 113996
```

```
Thread t0 part de : 0  
Thread t1 part de : 0  
Thread t2 part de : 602  
Thread t1 finit à : 2369037995  
Thread t0 finit à : 2373345600  
Thread t2 finit à : 2375342243  
Durée exécution volatile (ms): 127368
```



tout n'est pas parallélisable




impact du grain de
l'exclusion mutuelle



volatile \neq atomique

Exercice 3 : producteur-consommateur

- observation d'un blocage 
- particularité : un seul producteur et un seul consommateur
→ gestion de la cohérence simplifiée 