

# Rendu programmation concurrente

## Simulateur echange thermique version 2

Martini Didier

March 23, 2015

### 1 Introduction

Le but de cette application est de fournir un simulateur de temperature multi thread.

### 2 Parametre

Dans le code java fourni, le programme fait 10 iteration et affiche cela sur la sortie standard. Le mur testé est composé de 20cm de brique puis 12cm de laine de verre. Les constantes utilisées sont:

- temps entre deux calcul: 600s
- epaisseur de tranche: 4cm

Avec ces parametre il faut 50 iteration pour que la couche la plus proche du mur interieur change de temperature.

### 3 Structure de données

Les données des temperature sont stocké dans un tableau, les temperatures sont des double.

### 4 Calcul des constantes

Le calcul des constantes s'effectue au lancement du programme.

Constantes:  $\text{ConductiviteThermique} * \text{TEMPS\_ENTRE\_CALCUL} / (\text{MasseVolumique} * \text{ChaleurMassique} * \text{EPAISSEUR\_TRANCHE} * \text{EPAISSEUR\_TRANCHE})$ ;

## 5 10 premiere iteration

110,20,20,20,20-20,20,20,20,20  
110,44,20,20,20-20,20,20,20,20  
110,55,26,20,20-20,20,20,20,20  
110,62,32,21,20-20,20,20,20,20  
110,67,37,24,20-20,20,20,20,20  
110,70,41,26,21-21,20,20,20,20  
110,73,45,29,22-22,20,20,20,20  
110,75,48,31,23-23,20,20,20,20  
110,77,51,34,25-25,21,20,20,20  
110,79,53,36,26-26,21,20,20,20  
110,80,55,38,27-27,22,21,20,20

## 6 Algorithme - Thread

Pour cette deuxième version il y a N thread (N étant le nombre de coeur du processeur). Une liste contient l'ensemble des index du tableau a calculer pour l'étape T. Chaque thread va retirer l'un de ces index de la liste, puis faire le calcul correspondant a cet index. Quand le calcul est fini, chaque thread recommence ces etapes jusqu'a ce que la liste soit vide. Quand la liste est vide, les thread se rejoignent a la barriere (await), la liste se re remplit, et on reprend les calcul pour l'étape T+1.

## 7 Temps d'exécution

Le programme met environs 60 milliseconde a effectuer 100 000 iterations sans Thread. Le programme met environs 1400 milliseconde a effectuer 100 000 iterations avec Thread. Pour mesurer ce temps, nous avons recupere le temps en milliseconde au debut de l'exécution du programme, puis le temps en milliseconde a la fin de l'exécution du programme.

## 8 Modifier le programme

Les fichiers interessant pour la comprehension des calcul de la simulation et des threads sont:

"models/Simulateur.java"

"models/threading/Barriere.java"

"models/threading/Threading.java"

## **9 Pas d'iteration**

Il faut mettre un pas (temps entre 2 calcul) de  $60*60*24*365$  pour simuler une année complete.