

Laboratory Notebook for a Multi-Threaded Version of Quicksort

Maxime Chevalier

January 19, 2017

Contents

1	Indroduction	1
2	Caractéristiques de la machine	1
2.1	Version du système d’exploitation	1
2.2	CPU	2
2.3	RAM	2
3	Test 1	2
3.1	Analyse des résultats	3
3.2	Retestons en branchant !	3
3.3	Analyse	3
4	Test 2	3
5	Test 3	4

1 Indroduction

Ce journal est le compte rendu d’un élève de RICM4 à Grenoble.

2 Caractéristiques de la machine

2.1 Version du système d’exploitation

```
uname -a;
```

```
Linux chevamax-Satellite-Pro-R50-B 4.4.0-21-generic #37-Ubuntu SMP Mon Apr 18 18:33:37
```

2.2 CPU

```
lscpu;
```

```
Architecture:      x86_64
Mode(s):           opérateur(s) des processeurs :32-bit, 64-bit
Byte Order:       Little Endian
CPU(s):           4
On-line CPU(s) :   list: 0-3
Thread(s) par cœur : 2
Cœur(s) par socket : 2
Socket(s):        1
Nœud(s) NUMA :    1
Identifiant constructeur :GenuineIntel
Famille de processeur :6
Modèle : 69
Model name: Intel(R) Core(TM) i5-4210U
Révision : 1
Vitesse du processeur en MHz :997.875
CPU max MHz: 2700,0000
CPU min MHz: 800,0000
BogoMIPS: 4788.54
Virtualisation : VT-x
Cache L1d : 32K
Cache L1i : 32K
Cache L2 : 256K
Cache L3 : 3072K
NUMA node0 CPU(s): 0-3
Flags: fpu vme de pse
```

2.3 RAM

```
free -m;
```

```
total      utilisé      libre  partagé  tamp/cache  disponible
Mem:      7898      1702      1133      366      5062  5751
Partition d'échange:  7996          0      7996
```

3 Test 1

Réexécution du code sur ma machine

```
"exports both"body
```

3.1 Analyse des résultats

On voit ici que les résultats sont complètement différents (et même plutôt incohérents...). A se demander si la machine utilise ses différents coeurs!

3.2 Retestons en branchant !

```
"exports both"body
```

3.3 Analyse

C'est pas mieux ...

4 Test 2

Nous allons garder les mêmes valeurs en mélangeant l'ordre d'exécution

```
OUTPUT_DIRECTORY=data/'hostname'_ 'date +%F'
mkdir -p $OUTPUT_DIRECTORY
OUTPUT_FILE=$OUTPUT_DIRECTORY/measurements_'date +%R'.txt

touch $OUTPUT_FILE

for rep in `seq 1 5`; do
for i in 100 1000 10000 100000 1000000; do
echo "Size: $i" >> $OUTPUT_FILE;
./src/parallelQuicksort $i >> $OUTPUT_FILE;
done
done

FILENAME="data/chevamax-Satellite-Pro-R50-B_2017-01-19/measurements_14:57"
perl scripts/csv_quicksort_extractor2.pl < "$FILENAME.txt" > "${FILENAME}_wide.csv"
echo "
    set terminal png size 600,400
    set output '${FILENAME}_wide.png'
    set datafile separator ','
    set key autotitle columnhead
    plot '${FILENAME}_wide.csv' using 1:2 with linespoints, '' using 1:3 with linespoints
" | gnuplot
echo [[file:${FILENAME}_wide.png]]
```

5 Test 3

Dans ce premier test nous allons affiner la courbe en faisant des tests sur des tableaux de taille 10, 50, 100, 500, 1000, 5000, 10000, 50000, 100000, 500000 et 1000000.

Nous allons effectuer le test 10 fois pour chaque valeur.

```
:::::::::::::
scripts/run_benchmarking.sh
:::::::::::::

OUTPUT_DIRECTORY=data/‘hostname’_‘date +%F’
mkdir -p $OUTPUT_DIRECTORY
OUTPUT_FILE=$OUTPUT_DIRECTORY/measurements_‘date +%R’.txt

touch $OUTPUT_FILE
for i in 10 50 100 500 1000 5000 10000 50000 100000 500000 1000000; do
    for rep in ‘seq 1 10’; do
        echo "Size: $i" >> $OUTPUT_FILE;
        ./src/parallelQuicksort $i >> $OUTPUT_FILE;
    done ;
done
```

Pour transformer le fichier texte en fichier on utilise le script fourni et la commande :

```
FILENAME="data/chevamax-Satellite-Pro-R50-B_2017-01-18/measurements_19:40"
perl scripts/csv_quicksort_extractor2.pl < "$FILENAME.txt" > "${FILENAME}_wide.csv"
echo "
    set terminal png size 600,400
    set output '${FILENAME}_wide.png'
    set datafile separator ','
    set key autotitle columnhead
    plot '${FILENAME}_wide.csv' using 1:2 with linespoints, '' using 1:3 with linespoints
" | gnuplot
echo [[file:${FILENAME}_wide.png]]
```

