Comparatif des performances d'une application en fonction des paramètres de compilation utilisés et analyse des performances

TER M1 Informatique

Pierre Tassel sous la direction de Sid Touati

Département Informatique
Université Nice Sophia Antipolis

Master Informatique

18 juin 2018

Plan

Présentation du programme

MinMax et le jeux de l'Awale

- MinMax avec coupes Alpha Beta
- Variante du jeux de l'Awale avec 6 cases et 4 graines par case
- Ecrit en C++ par M Régin

Figure – Le jeu de l'Awalé.

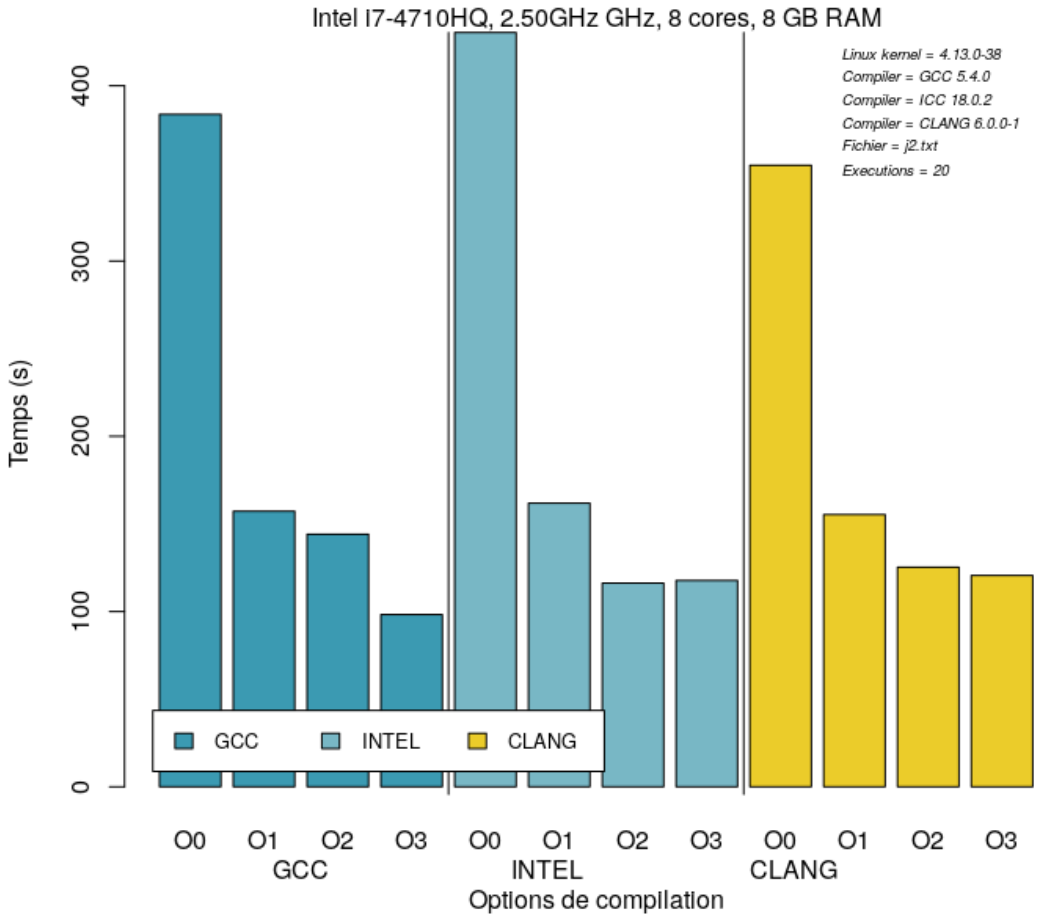
Démarche expérimentale

Environnement matériel et logiciel

- Processeur Scaling désactivé, processeur 8 coeurs 2,5GHz, 8Go DDR3, 250Go de SSD
- Environnement minimaliste en CLI, Arch Linux basé sur noyau 4.14.13-1
- 3 compilateurs analysés, GCC, ICC et CLang dernières versions disponible
- Fichier d'entrée généré en faisant jouer le programme contre lui même
- On répète les exécutions 20 fois pour chaque configurations testé

Analyse des performances séquentielles

Figure – Temps d'exécution séquentiel quand le programme commence.



Profilage du code

GProf

```
 %   cumulative          total
time   seconds   calls   s/call  name
48.83    167.72 5172852992  0.00  jouer_coup(Next*, Pos*, Pos*, int, int)
18.25    230.39 3262157329  0.00  copier(Pos*, Pos*)
17.22    289.56 3262157262  0.00  est_affame(Pos*, int)
 8.79    319.75 1457826958  0.00  calculer_coup(Next*, Pos*, int, int, int, int, bool)
 5.31    337.98 3252245939  0.00  valeur_minimaxAB(Next*, Pos*, int, int, int, int, bool)
 0.87    340.99 1775297034  0.00  evaluer(Pos*)
 0.52    342.77     67      0.03  test_fin(Pos*)
 0.09    343.65     33     10.34  decisionAB(Next*, Pos*, int, bool)
```

Figure – Analyse du code avec GProf.

Elapsed Time : 118.666s

CPU Time :	117.755s
Instructions Retired:	588,377,500,000
CPI Rate :	0.499
CPU Frequency Ratio :	1.000
Total Thread Count:	1
Paused Time :	0s

Top Hotspots

This section lists the most active functions in your a

Function	Module	CPU Time
jouer_coup	advisor.out	70.132s
calculer_coup	advisor.out	15.441s
copier	advisor.out	14.165s
est_affame	advisor.out	7.176s
valeur_minimaxAB	advisor.out	5.755s
[Others]		5.087s

Figure – vTune analyse usage processeur.

Elapsed Time : 120.174s

CPU Time :	118.454s
Memory Bound :	2.7%
L1 Bound :	6.6%
DRAM Bound :	
DRAM Bandwidth Bound :	0.0%
LLC Miss :	0.1%
Loads:	157,360,120,662
Stores:	100,867,825,944
LLC Miss Count :	900,054
Average Latency (cycles) :	8
Total Thread Count:	1
Paused Time :	0s

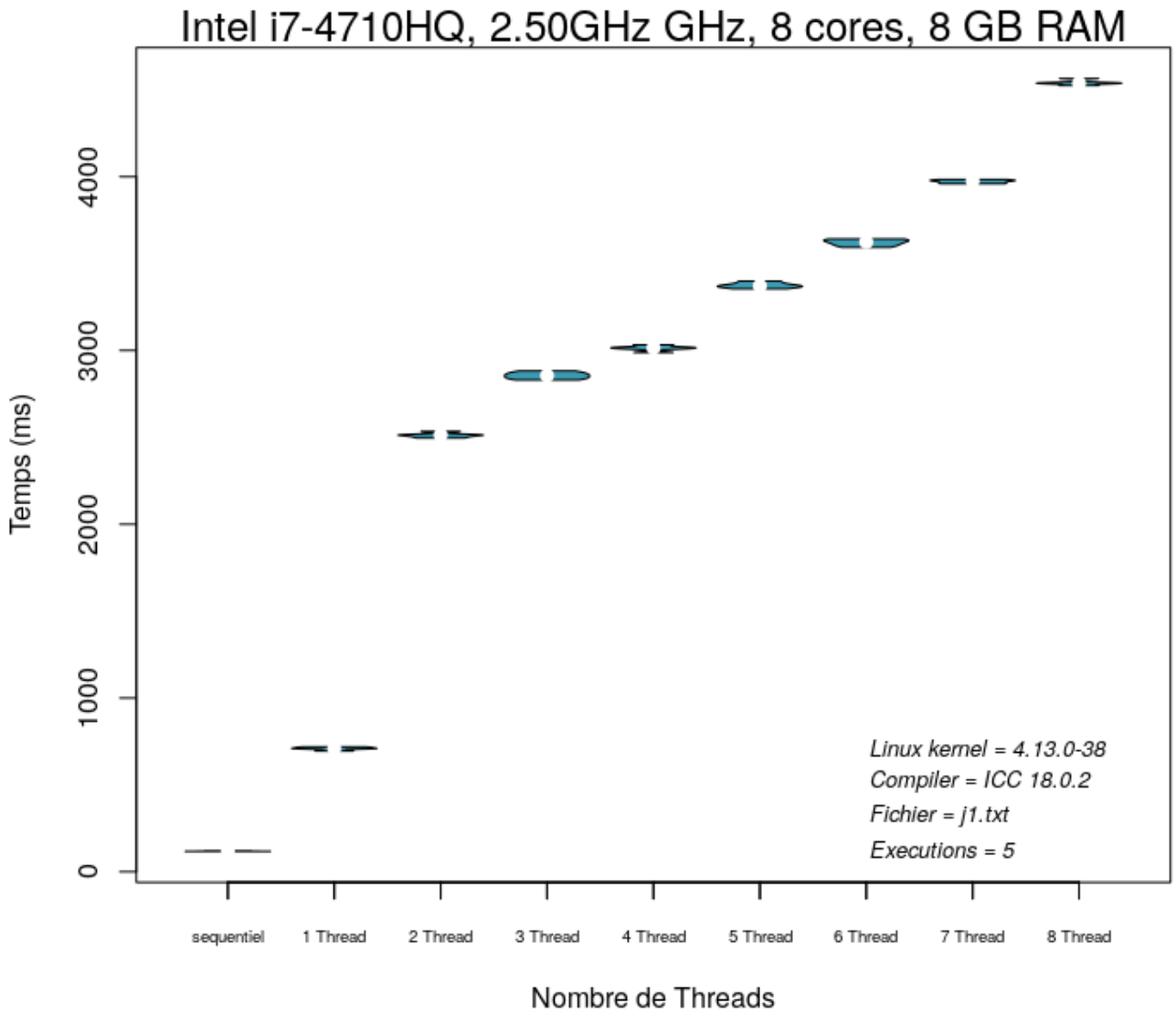
Figure – vTune analyse usage mémoire.

Difficultés de l'optimisation du code

- Algorithme très séquentiel
- Nombre d'itérations faibles
- Usage de pointeurs qui pourraient aveugler les compilateurs

Approche naïve

Figure – Résultat ajout parallélisme avec compilateur Intel.



False Sharing

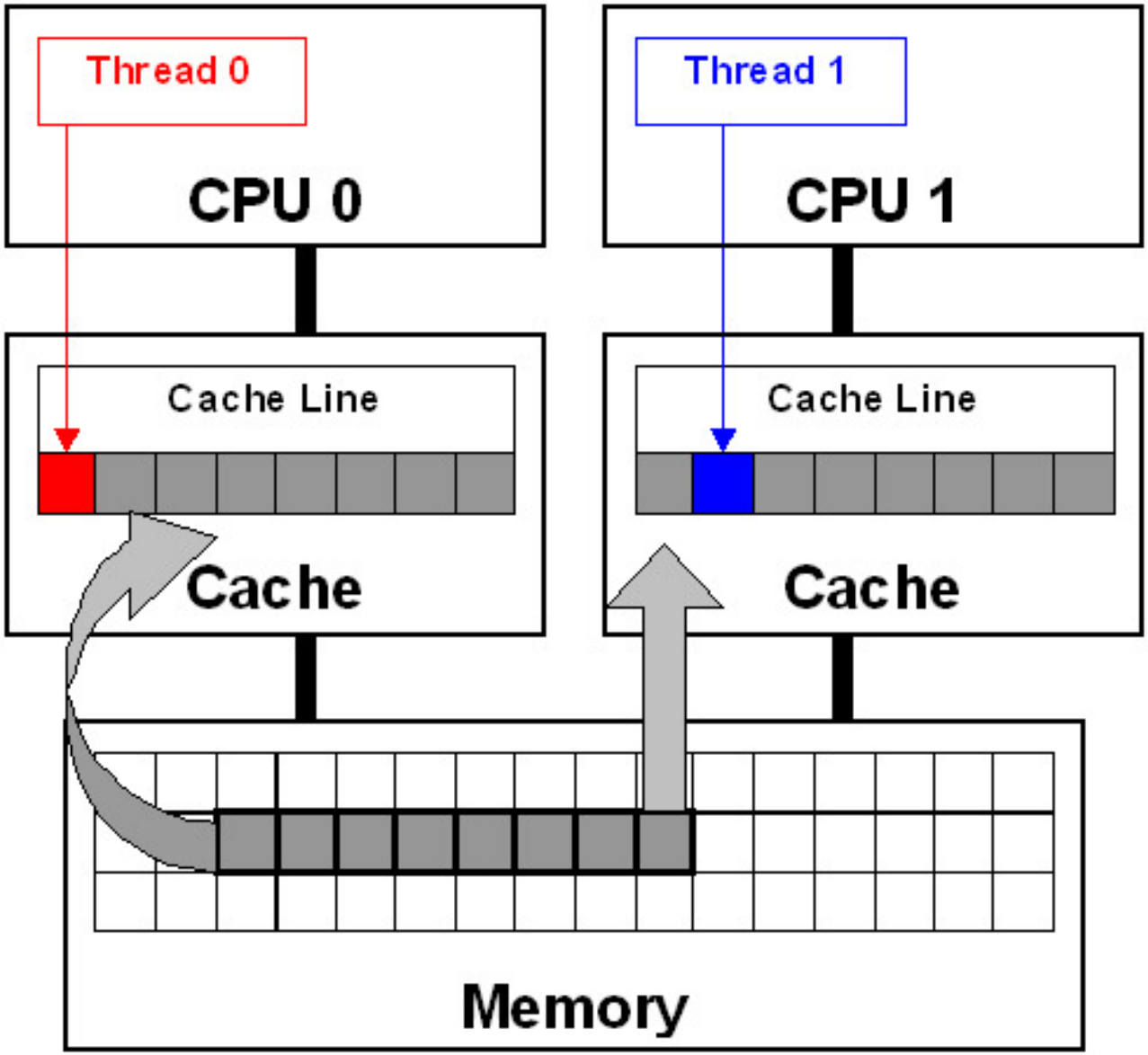
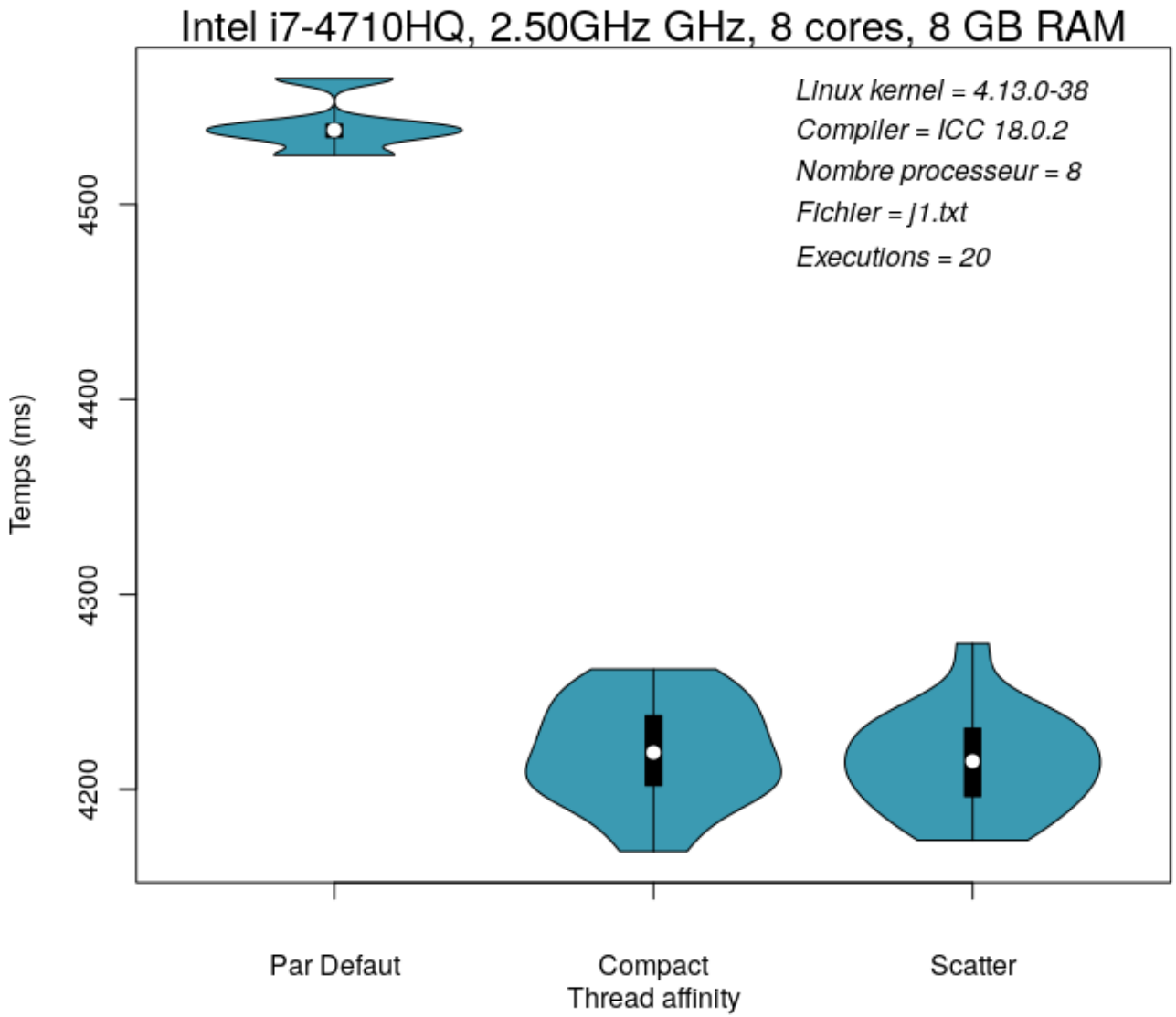


Figure – Le phénomène de False Sharing.

Thread Affinity

Figure – Analyse de l'impact du placement des threads.



Intel Advisor

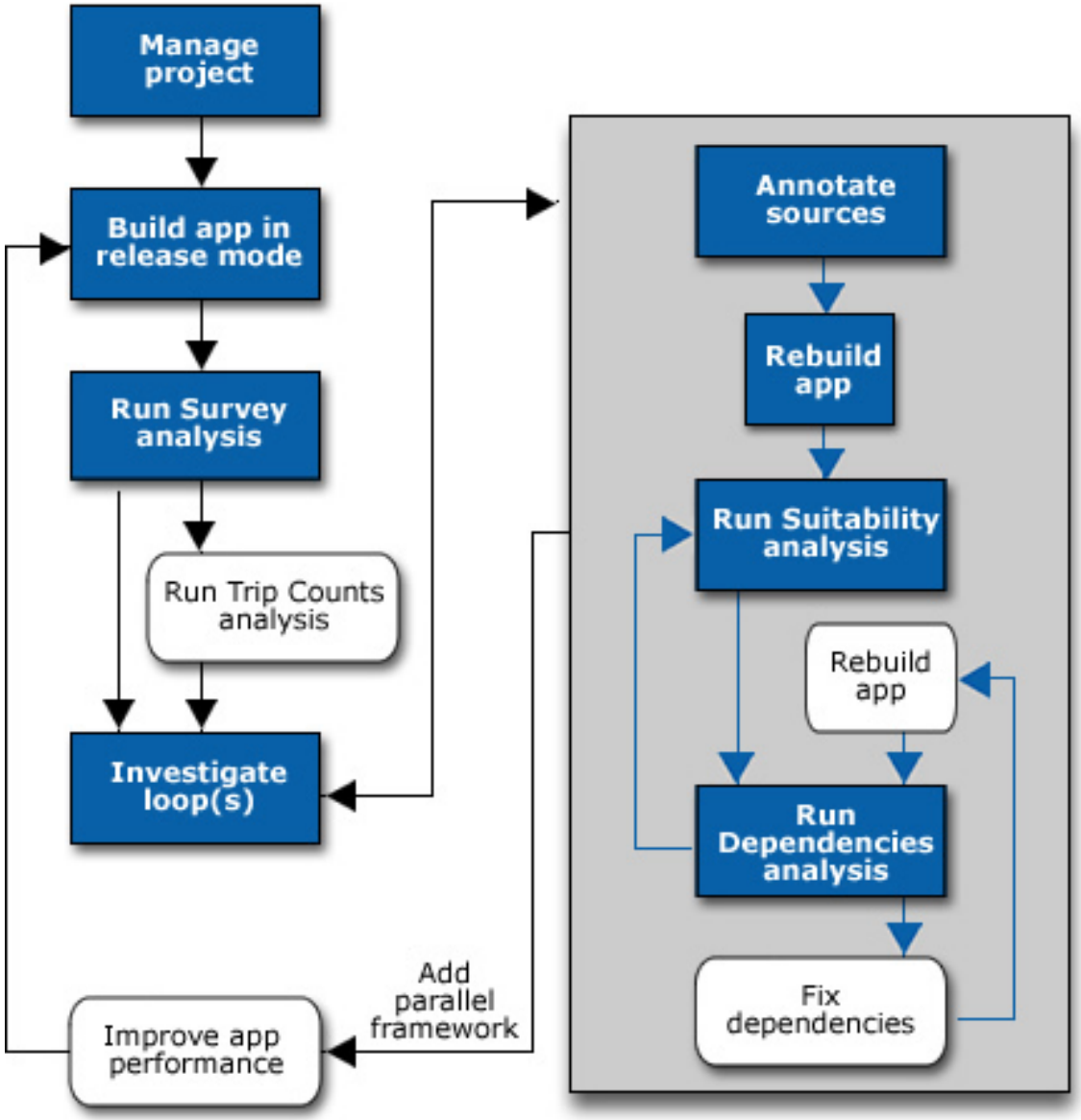


Figure – Processus itératif d'Intel pour ajout parallélisme.

Modification Algorithmique

- Dictionnaires ouvertures.
- Trie des coups.
- Table de transposition.

Trie coups

Figure – Analyse de l'impact du trie des coups.

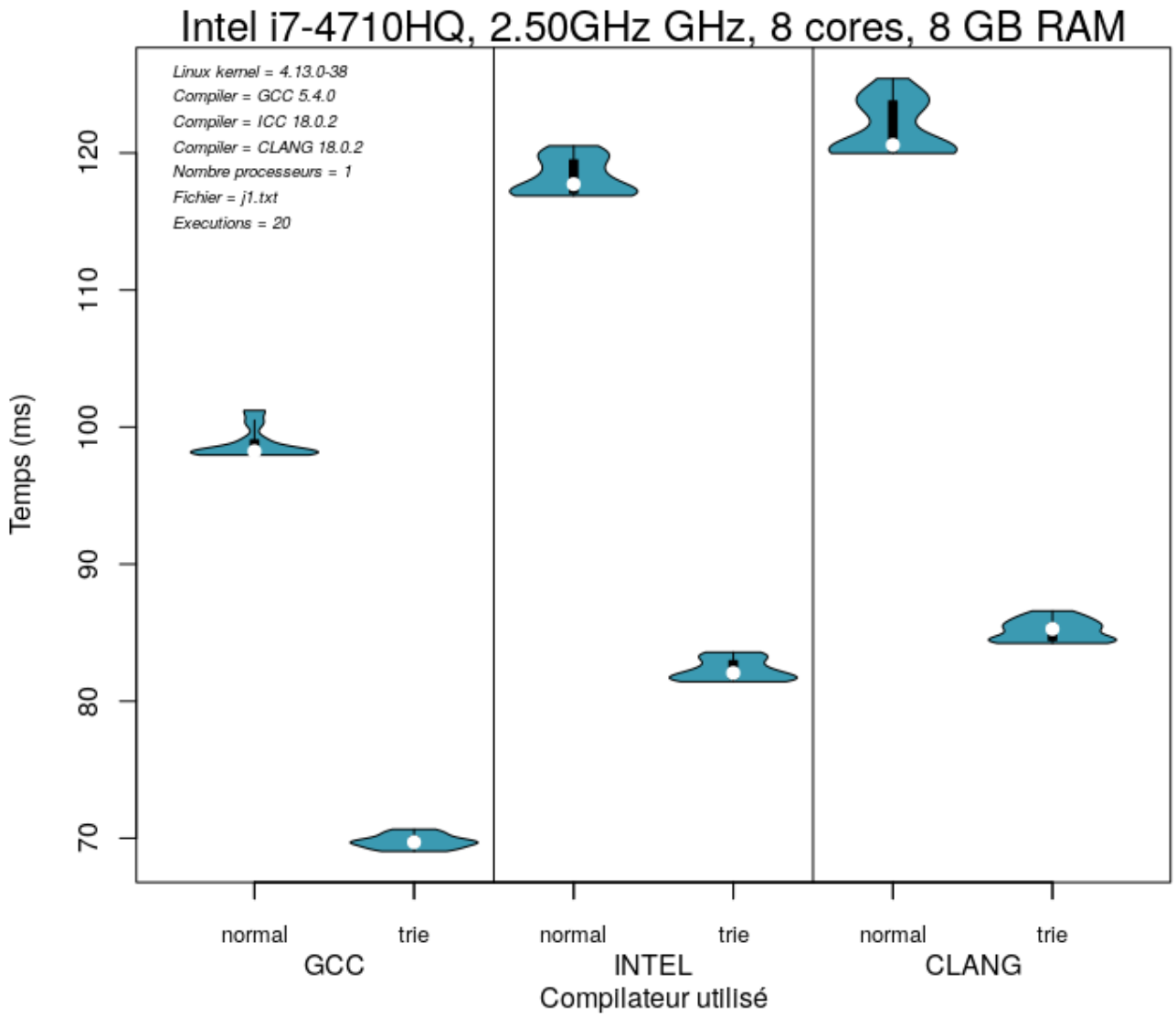


Table de transposition

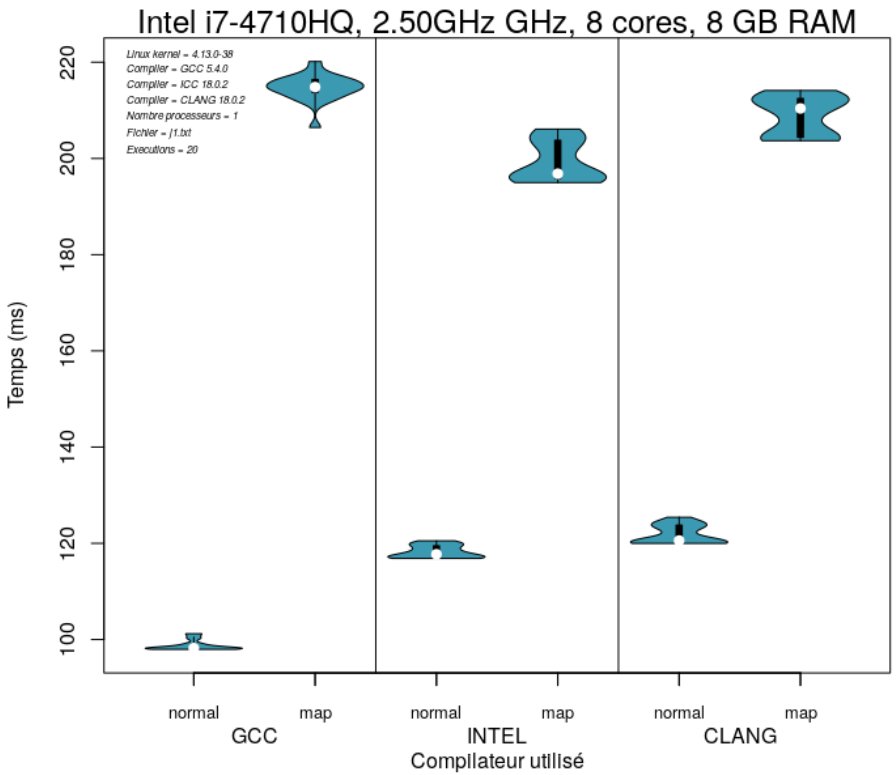


Figure – Arbre binaire de recherche.

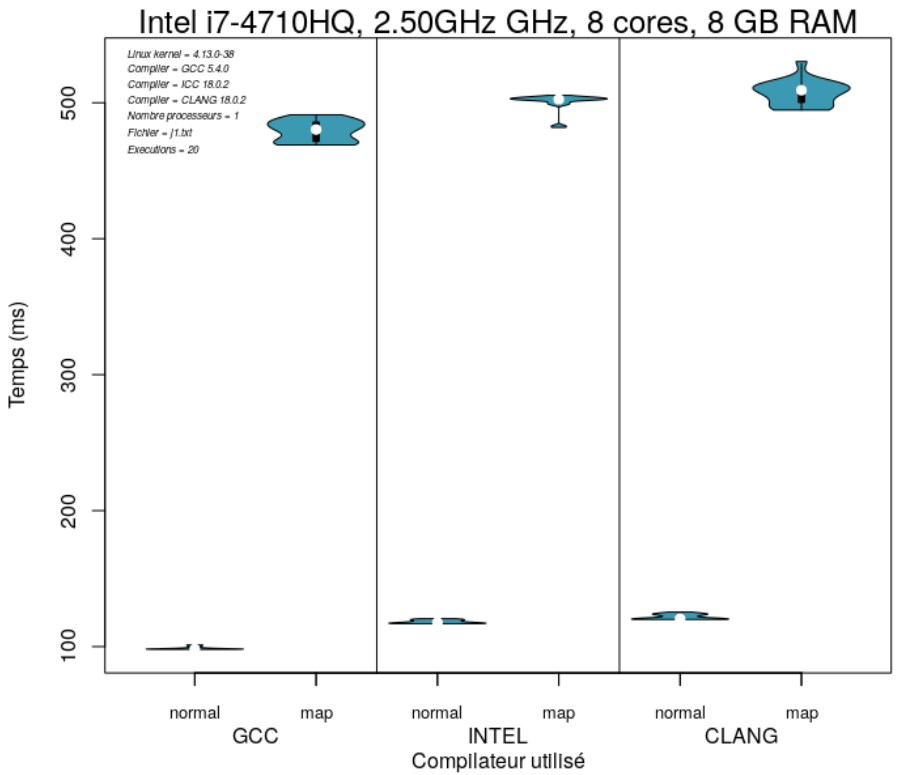


Figure – Table de hashage.

Conclusion et perspectives

- Stage à l'INRIA.
- Réduire le False Sharing.
- Maintenir coups trié aux sous noeuds.