

M1 SESI 2017-2018  
Architecture Multi-Processseurs  
TP4 : Caractérisation et Dimensionnement des Caches

Kevin Mambu

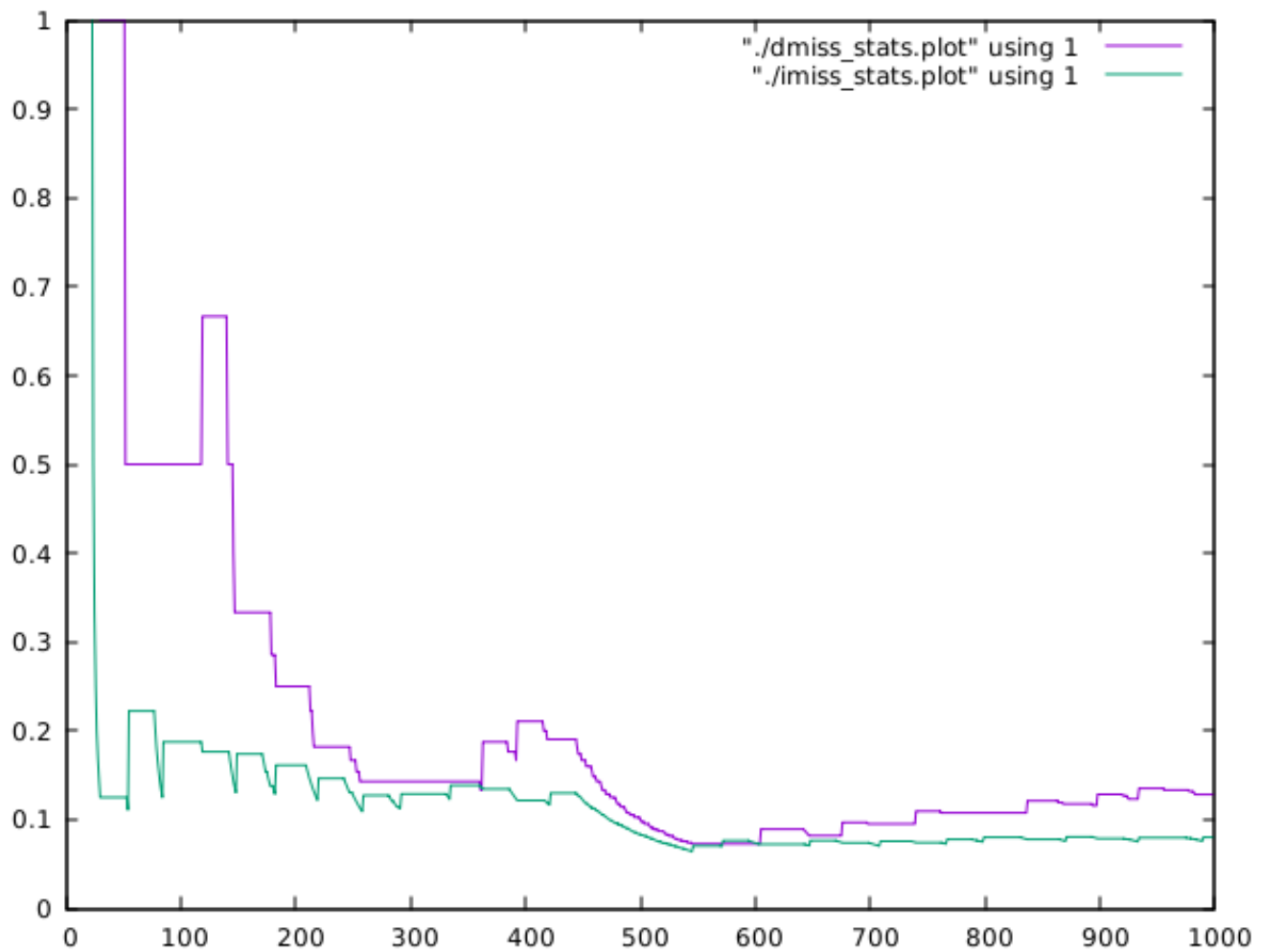
March 9, 2018

### Question C1

Avec cette configuration, l'exécution de la simulation dure 75725 cycles.

### Question C2

Le graphe ci-dessous est l'évolution des taux de MISS sur les caches en fonction du temps sur une durée de 1000 cycles :



- cycle 20 : Chargement du code du reset depuis la ROM  
IMISS RATE = 1.
- cycle 30 : Chargement de la première ligne de cache du tableau HexaTab.
- cycle 51 : Fin lecture dans le code du reset DMISS RATE = 0.5.
- cycle 76 : Fin chargement de la première ligne de cache du code du main  
IMISS RATE = 0.22
- cycle 117 : Fin chargement ligne de cache du main après branchement  
IMISS RATE = 0.1875
- cycle 140 : Fin lecture de la pile mémoire  
Chaque paire de crête IMISS RATE / DMISS RATE peut être apparenté au chargement des arguments de sum depuis la pile mémoire
- cycle 363 : Lecture deuxième ligne de cache du tableau HexaTab.
- cycle 393 : lecture de données non-cachable (tty\_printf)

## Question C3

Il y a un signal interne *run\_cycles*, correspondant aux nombres de cycles d'exécutions actives du processeur. Il est égal à  $\frac{c\_total\_cycles}{c\_frz\_cycles}$ .

Le CPI est égal à  $\frac{c\_total\_cycles}{run\_cycles}$ .

Le taux de MISS d'instructions IMISS RATE est égal à  $\frac{c\_imiss\_count}{run\_cycles}$ .

Le taux de MISS de données DMISS RATE est égal à  $\frac{c\_dread\_count - c\_dunc\_count}{run\_cycles}$ .  
c.dread\_count comptabilise toutes les lectures.

## Question D1

```

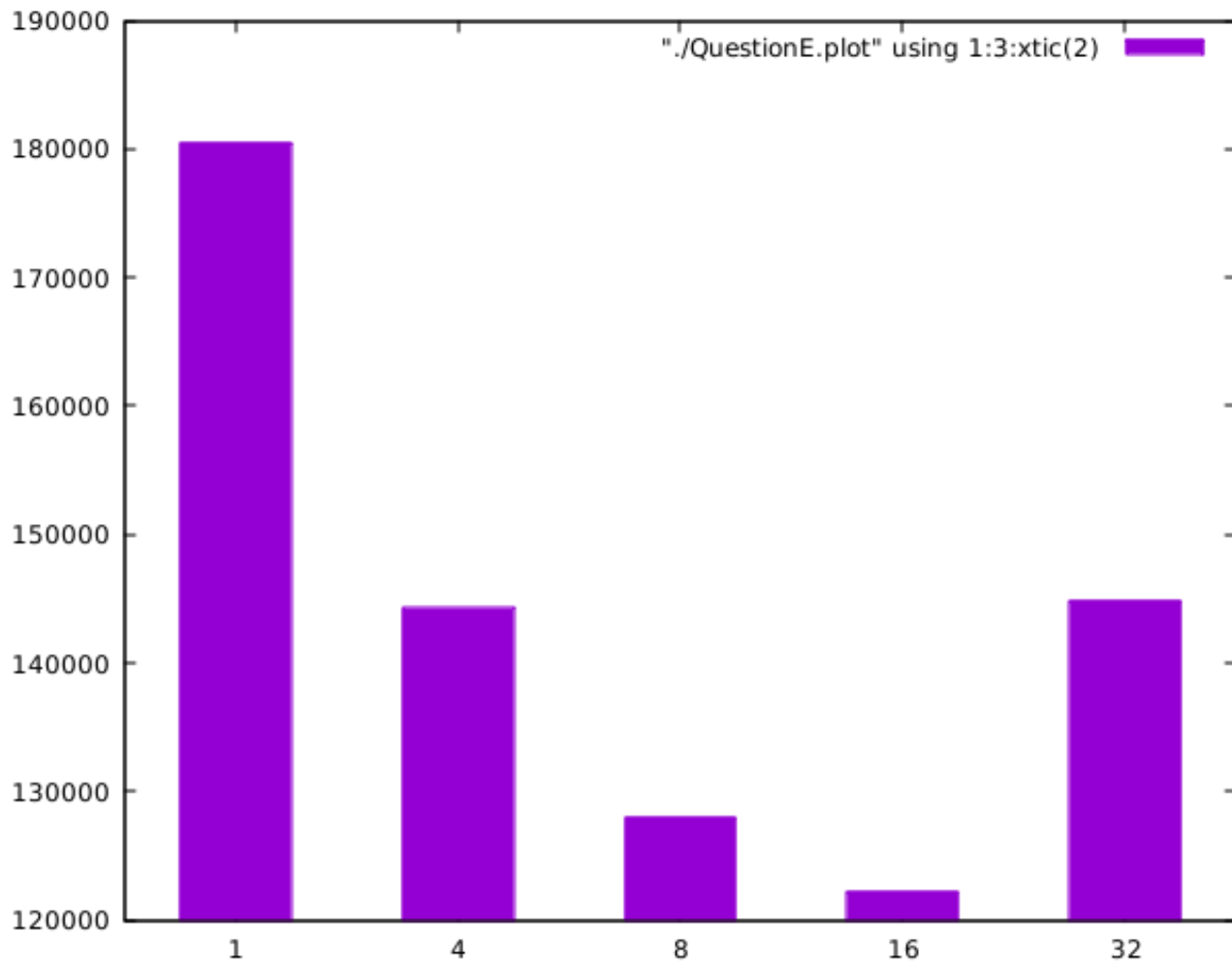
1 ISETS = 256 :
3 - IMISS RATE = 0.00108192
  - IMISS COST = 16.2105
5 - CPI
  NCYCLES = 75888
7 ISETS = 256 :
  - IMISS RATE = 0.0236548
9 - IMISS COST = 15.8833
  - CPI = 1.71993
11 NCYCLES = 94320
  ISETS = 256 :
13 - IMISS RATE = 0.111868
  - IMISS COST = 15.5302
15 - CPI = 3.17477
  NCYCLES = 150817
17 ISETS = 256 :
  - IMISS RATE = 0.142406
19 - IMISS COST = 15.4564
  - CPI = 3.67259
21 NCYCLES = 170389
  ISETS = 256 :
23 - IMISS RATE = 0.242413
  - IMISS COST = 15.0683
25 - CPI = 5.22878
  NCYCLES = 238648

```

QuestionD.txt

## Question D2

## Question E1



Lorsque la configuration est de 16 octets de largeur, sachant que ce programme utilisateur contient beaucoup de branchements et d'appels de fonctions, ce ratio semble optimisé pour le chargement des instructions sans prendre trop de temps. Sur la part de gauche du graphe, une à deux instructions par ligne seulement sont chargées, ce qui augmente la fréquence de lecture en mémoire. Sur la part de droite, les grandes lignes de cache permettent de charger plus d'instructions, 4 à 5 par ligne. Mais cela augmente le temps de chargement d'une ligne de cache en mémoire du coup.

Le juste milieu est à des cases de 16 octets.

## Question F1

```
2 ISETS = 256 :  
  - IMISS RATE = 0.00108192  
4  - IMISS COST = 16.2105  
  - CPI  
6 NCYCLES = 75888  
  ISETS = 256 :  
8  - IMISS RATE = 0.0236548  
  - IMISS COST = 15.8833  
10 - CPI = 1.71993  
  NCYCLES = 94320  
12 ISETS = 256 :  
  - IMISS RATE = 0.111868  
14 - IMISS COST = 15.5302  
  - CPI = 3.17477  
16 NCYCLES = 150817  
  ISETS = 256 :  
18 - IMISS RATE = 0.142406  
  - IMISS COST = 15.4564  
20 - CPI = 3.67259  
  NCYCLES = 170389  
22 ISETS = 256 :  
  - IMISS RATE = 0.242413  
24 - IMISS COST = 15.0683  
  - CPI = 5.22878  
26 NCYCLES = 238648
```

QuestionD.txt

## Question G1

- Le buffer d'écriture postées est une queue dans laquelle le contrôleur de cache peut stocker les requêtes (protocole FIFO)
- Lorsque le processeur veut poster une requête d'écriture alors que le buffer est plein, il doit attendre que ce dernier ne soit plus plein (stall)
- Lorsque le processeur fait une requête de lecture qui fait MISS alors que le BEP est non-vide, les requêtes postées sont prioritaires sur le bus, pour s'assurer que les données les plus récemment modifiées soient bien présentes en mémoires (problème de consistance mémoire)