Modèle de performance des applications et de plates-formes parallèles pour le placement de processus

Nicolas Denoyelle

Encadrants :
Brice Goglin - Emmanuel Jeannot

Runtime - Bordeaux

25 juin 2014

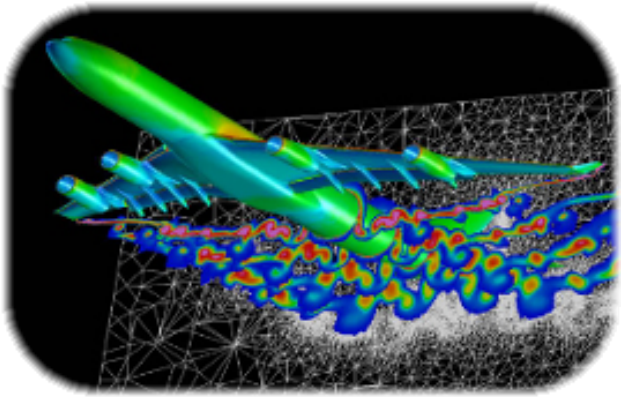
INTRODUCTION : Calcul Haute Performance

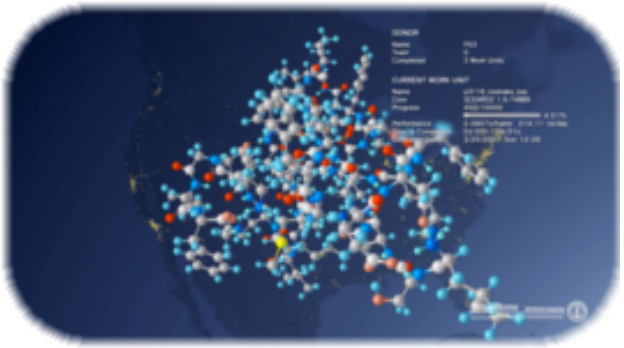
Besoins en calcul croissants

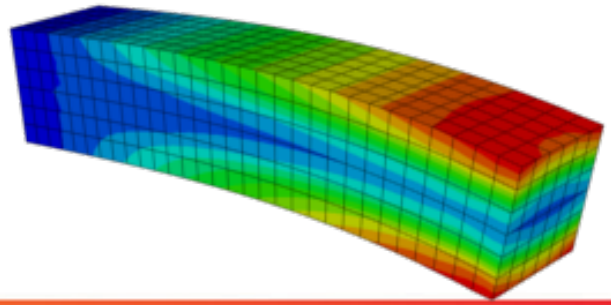
Ordinateurs de plus en plus puissants :
Aujourd'hui, top 500 : 33 PFlop/s

Problématiques nombreuses :

- Algorithmes performants
- Mise à l'échelle
- Portabilité
- Architectures et réseaux performants
- Exploitation des architectures
- Consommation ...







SOMMAIRE

Supports d'exécution pour le calcul haute performance

Le placement de tâches vecteur de performance

Observation de l'impact du placement de tâches

Modèle d'application pour le placement de tâches

Conclusion et perspectives

1 Supports d'exécution

et parallélisme de tâches

Supports d'exécution

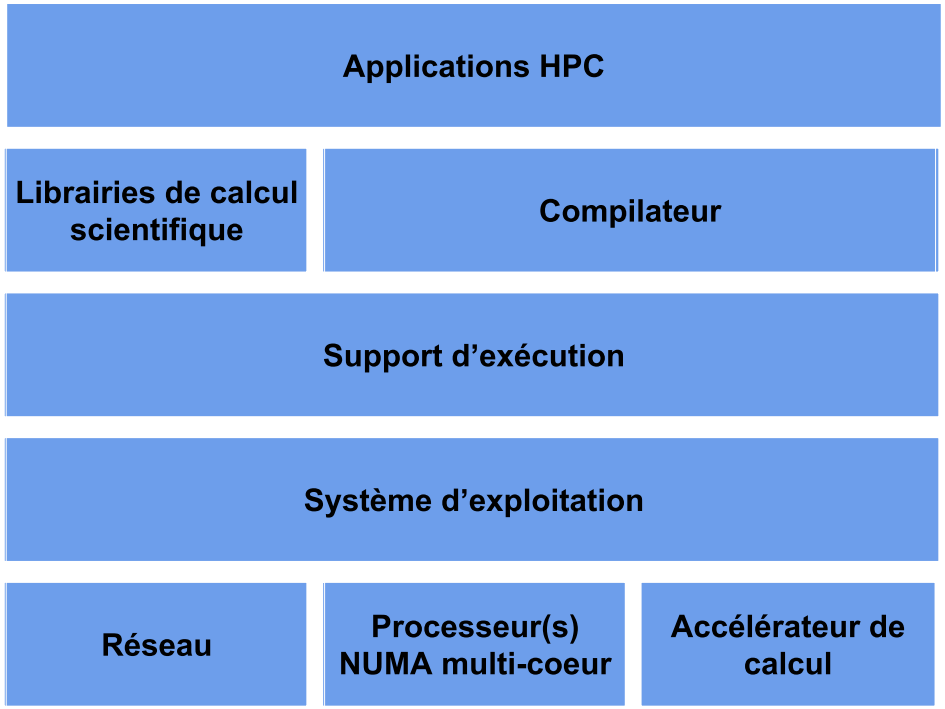
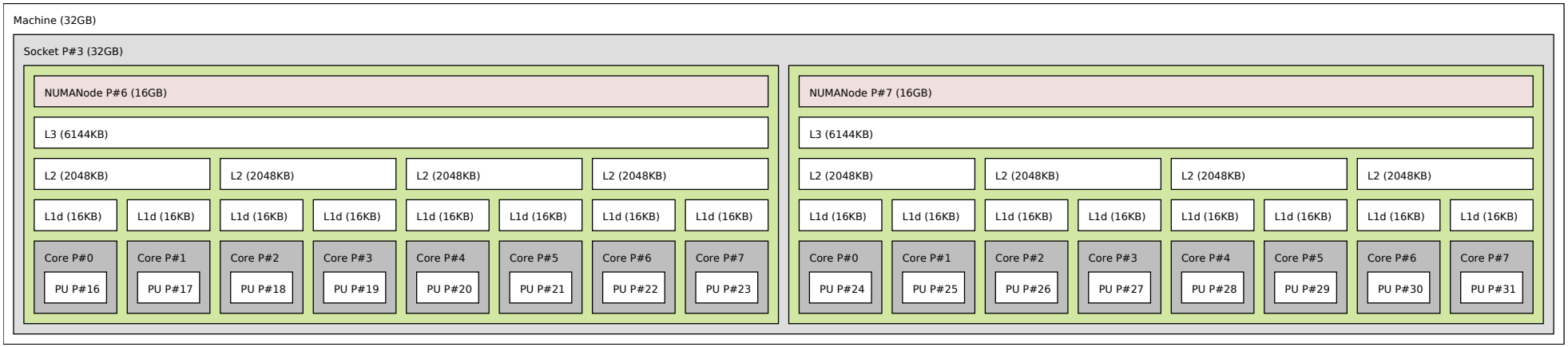


Figure : Pile HPC

- système : topologie, compteurs matériels, état de la machine, ...
- compilateur : instructions, parallélisme, besoins en mémoire . . .

Modèle de représentation de la machine

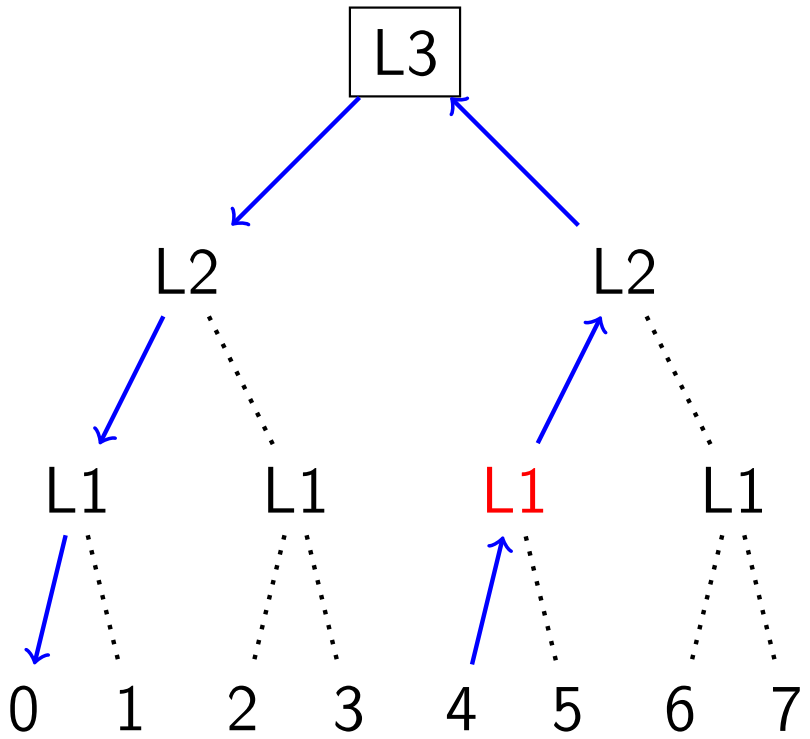


- Loi de Moore $\Longrightarrow$ multiplication du nombre de coeurs dans les processeurs.
- Hiérarchisation des ressources de calcul.
- Découpage en tâches pour exploiter le parallélisme des machines.

2
Le placement de tâches vecteur de performance

Illustration du problème

Placement par affinité

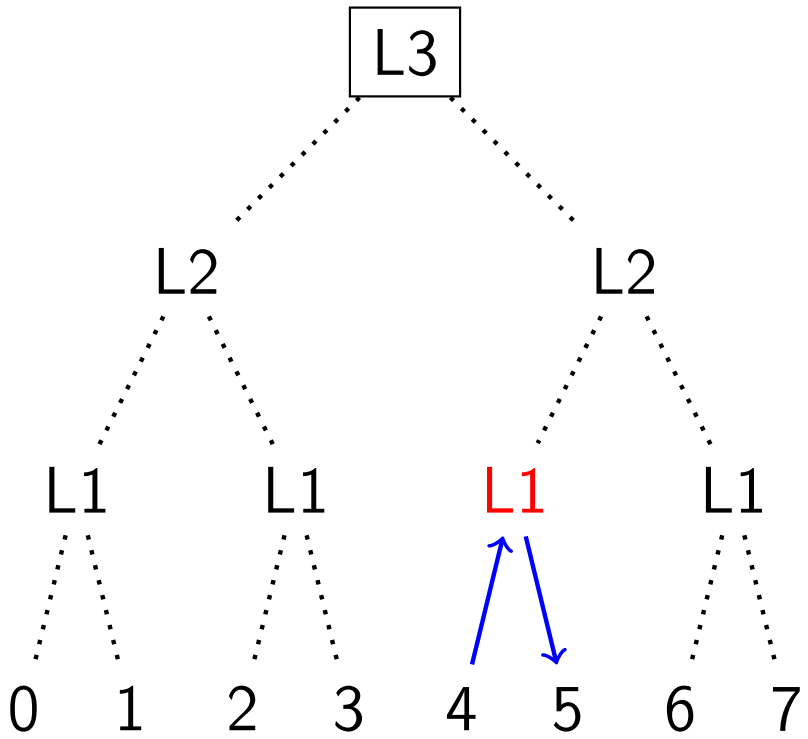


Améliore

- communications
- partage de données
- synchronisation entre les tâches

Illustration du problème

Placement par affinité



Améliore

- communications
- partage de données
- synchronisation entre les tâches

Illustration du problème

Améliore

L'utilisation des caches, pour des tâches plutôt indépendantes.

Placement par concurrence

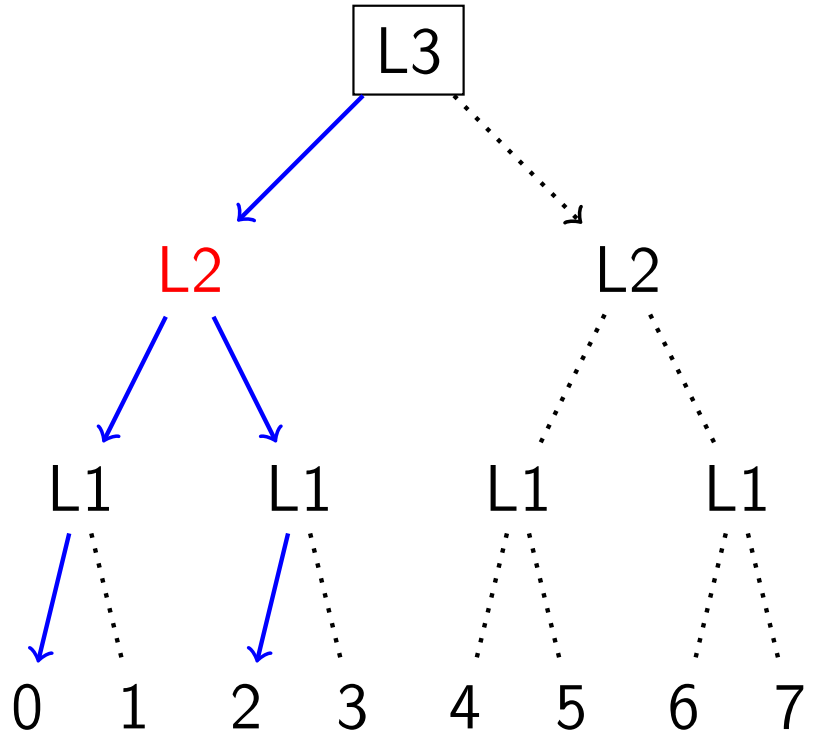
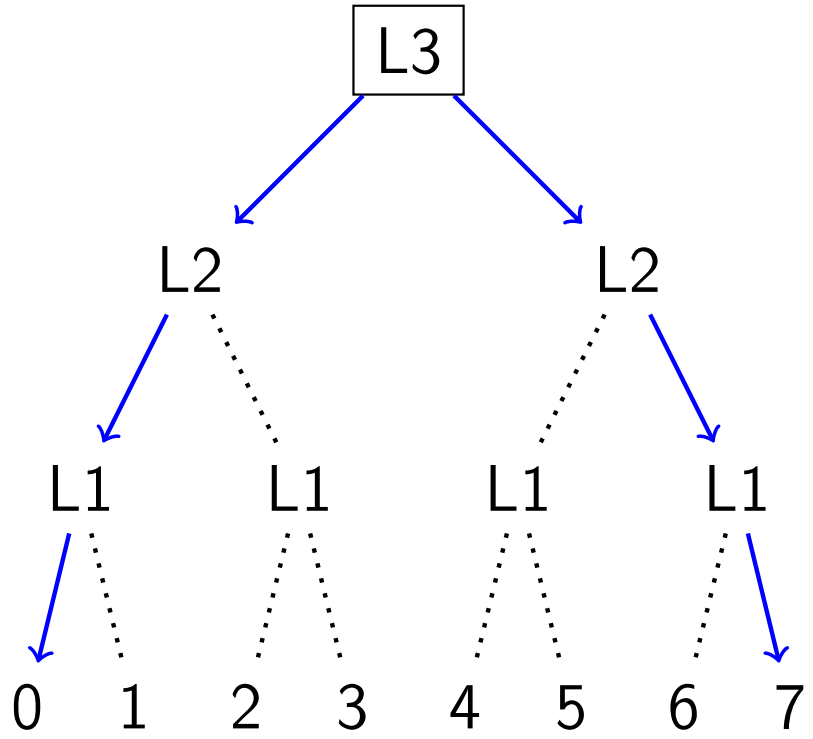


Illustration du problème

Améliore

L'utilisation des caches, pour des tâches plutôt indépendantes.

Placement par concurrence



Dualité entre affinité et concurrence

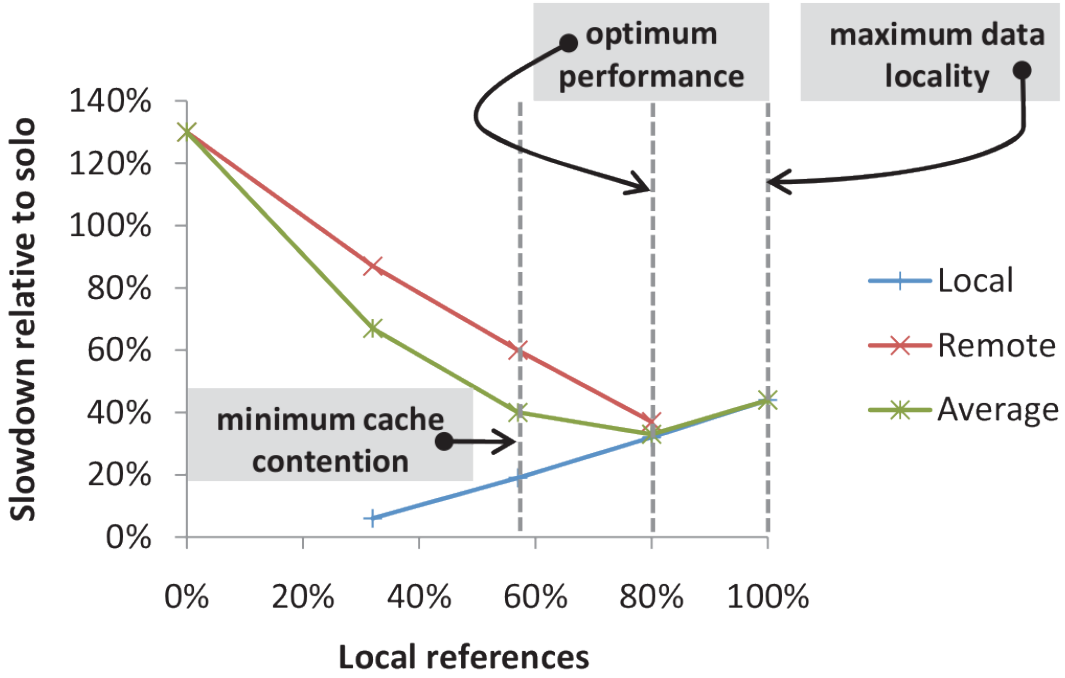


Figure : [Majo, Zoltan and Gross, Thomas R., 2011]

Lequel des deux critères précédents choisir ?

Problématique restreinte

- Il est impossible d'essayer toute les applications, nous avons besoin d'un modèle quantitatif.

- Choix : problème de concurrence d'accès aux caches.

Comment devons nous placer les tâches dans ce cas ? Quel gain de temps pouvons nous espérer ?

3
Exemples de placements de tâche

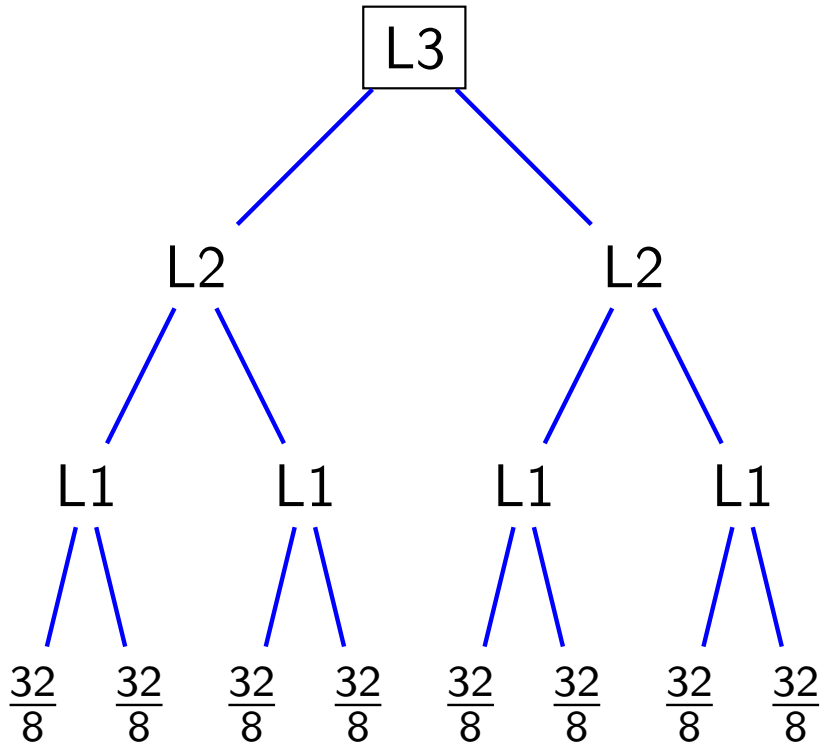
Pour se convaincre de l'intérêt de réduire la contention

ddot taille 32

64 opérations, 96 lecture, 32 écritures

contention level 0

- 8 processus, 1 ddot de taille 4 par processus.



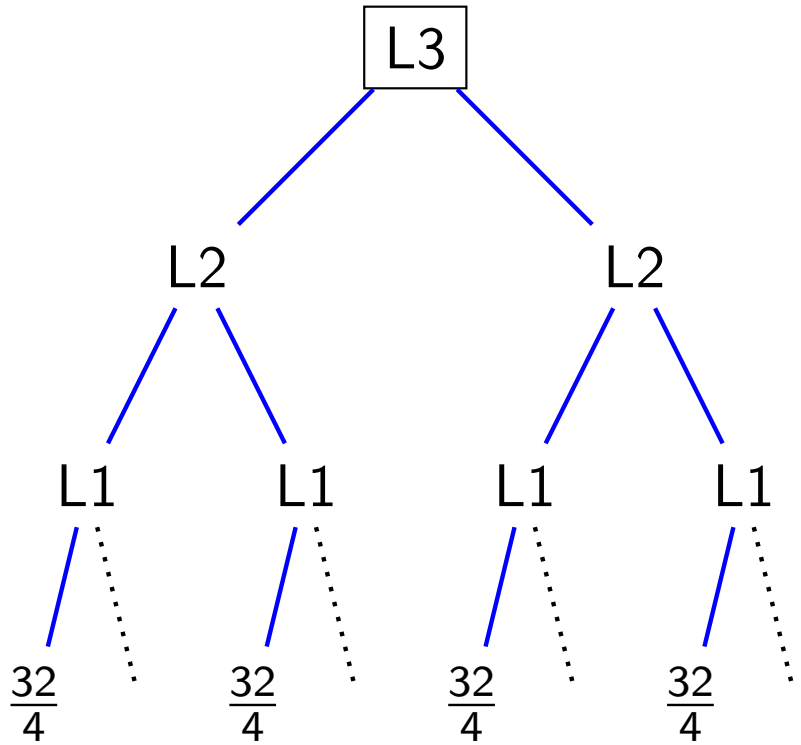
Pour se convaincre de l'intérêt de réduire la contention

ddot taille 32

64 opérations, 96 lecture, 32 écritures

contention level 1

- 4 processus, 1 ddot de taille 8 par processus.



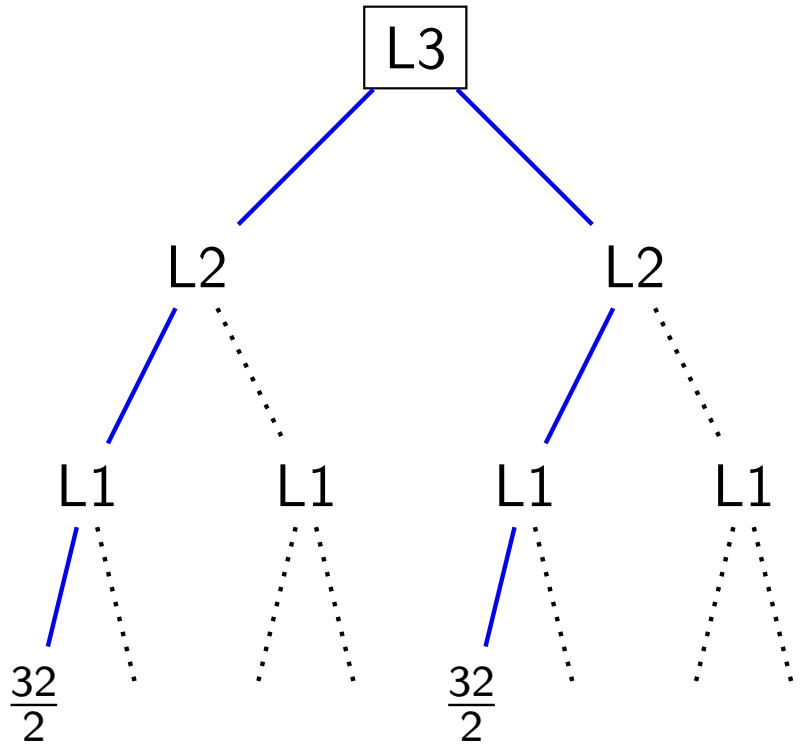
Pour se convaincre de l'intérêt de réduire la contention

ddot taille 32

64 opérations, 96 lecture, 32 écritures

contention level 2

- 2 processus, 1 ddot de taille 16 par processus.



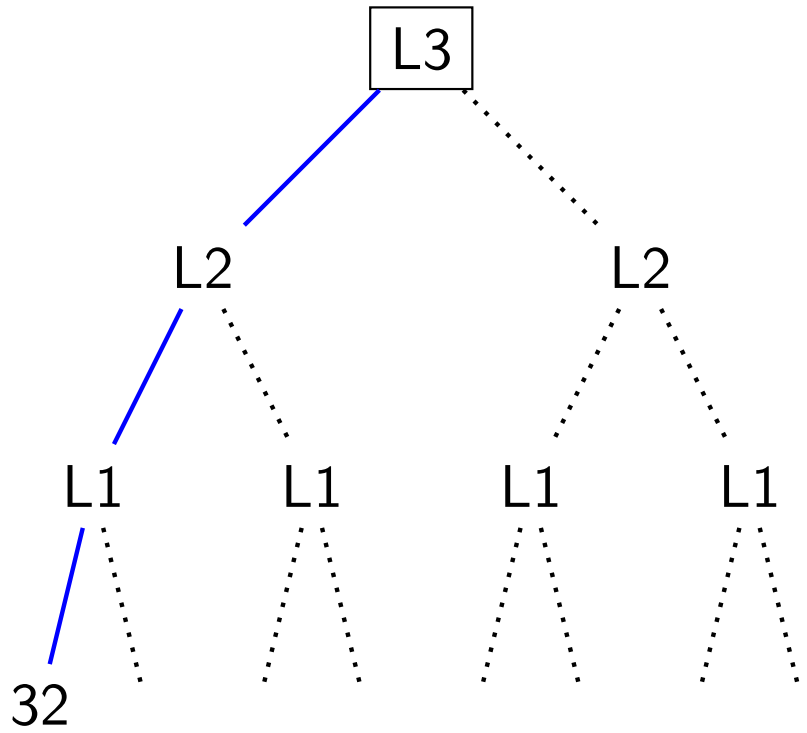
Pour se convaincre de l'intérêt de réduire la contention

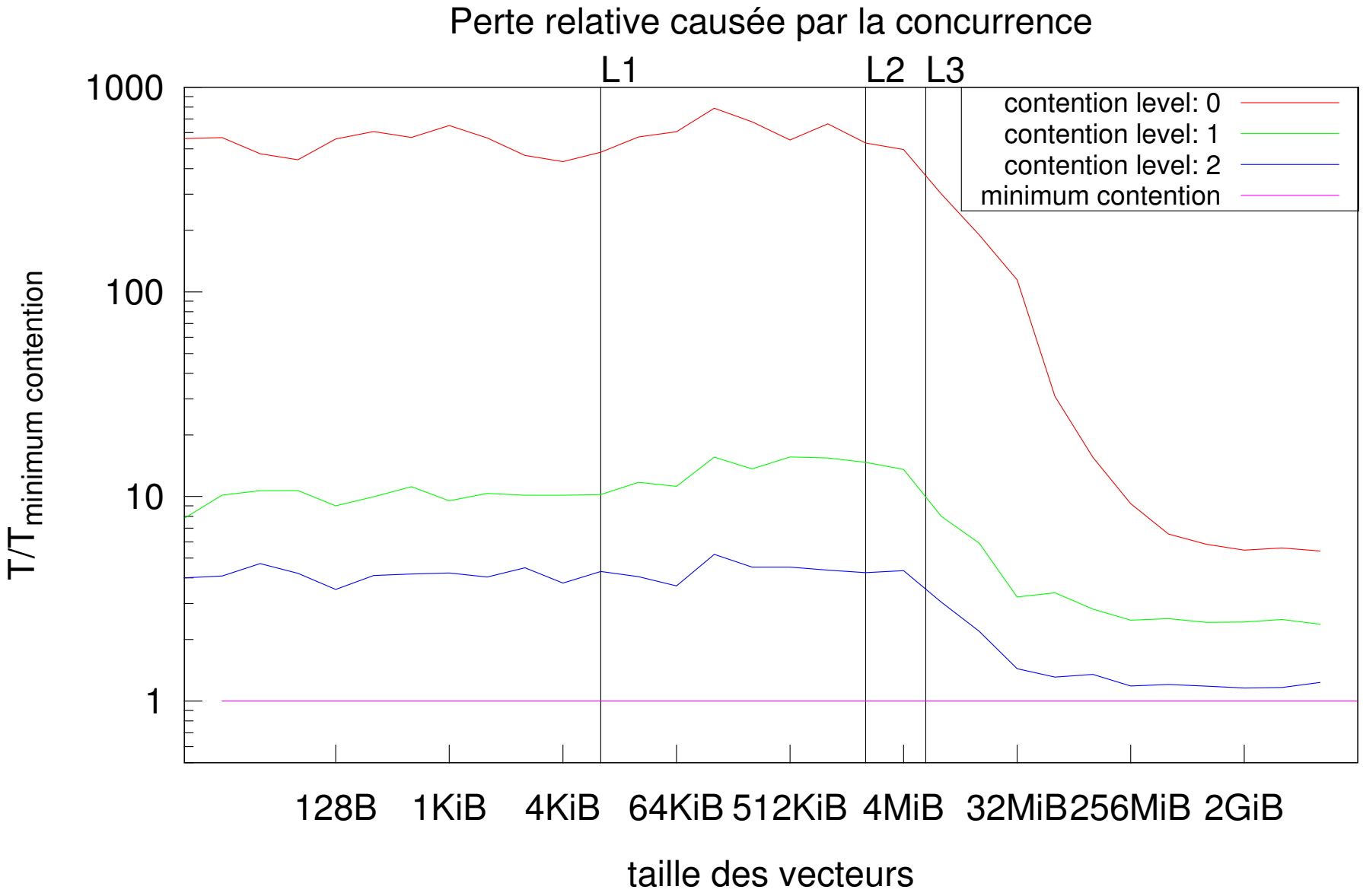
ddot taille 32

64 opérations, 96 lecture, 32 écritures

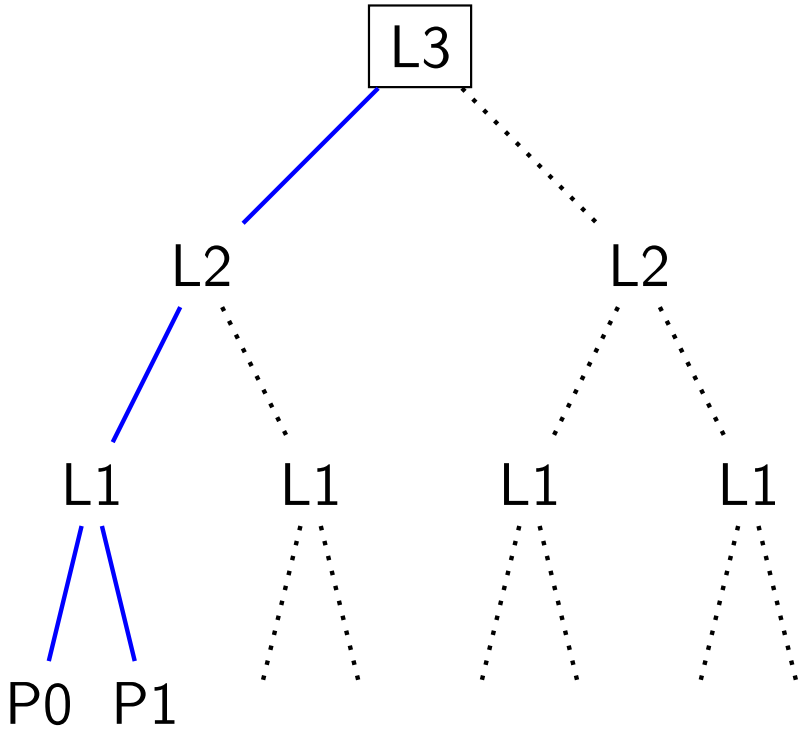
minimum contention

- 1 processus, 1 ddot de taille 32 par processus.




Perte relative causée par la concurrence
L1
L2
L3
1000
100
10
1
T/T minimum contention
contention level: 0
contention level: 1
contention level: 2
minimum contention
128B
1KiB
4KiB
64KiB
512KiB
4MiB
32MiB
256MiB
2GiB
taille des vecteurs

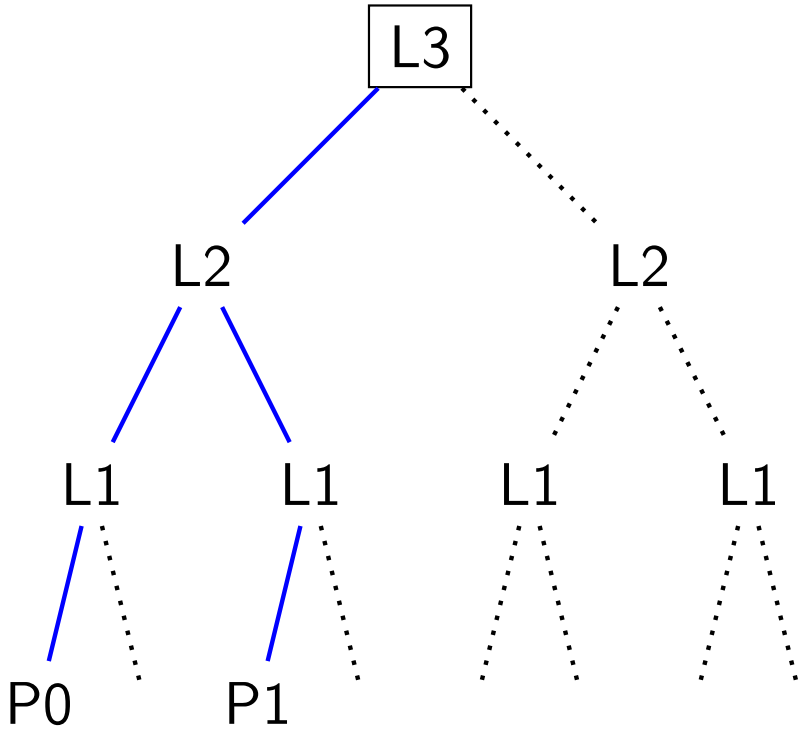
Avec les NAS, on ajoute de l'affinité entre les tâches



is	time
no bind	1.61
0,1,2,3	3.00
0,2,4,6	1.74
0,8,16,24	1.58
0,16,32,48	1.55

cg	time
no bind	42.77
0,1,2,3	45.34
0,2,4,6	34.90
0,8,16,24	33.12
0,16,32,48	32.94

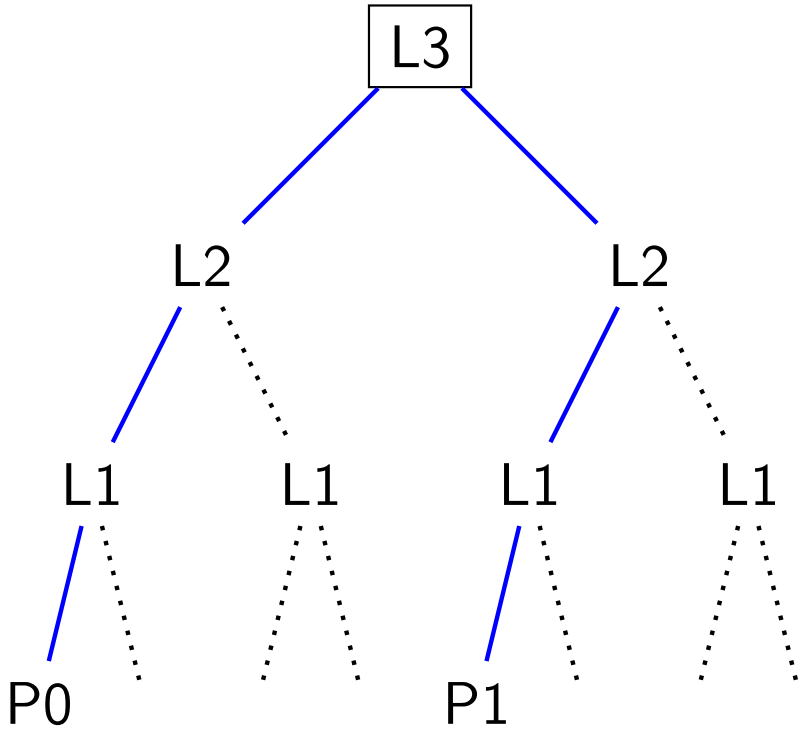
Avec les NAS, on ajoute de l'affinité entre les tâches



is	time
no bind	1.61
0,1,2,3	3.00
0,2,4,6	1.74
0,8,16,24	1.58
0,16,32,48	1.55

cg	time
no bind	42.77
0,1,2,3	45.34
0,2,4,6	34.90
0,8,16,24	33.12
0,16,32,48	32.94

Avec les NAS, on ajoute de l'affinité entre les tâches



is	time
no bind	1.61
0,1,2,3	3.00
0,2,4,6	1.74
0,8,16,24	1.58
0,16,32,48	1.55

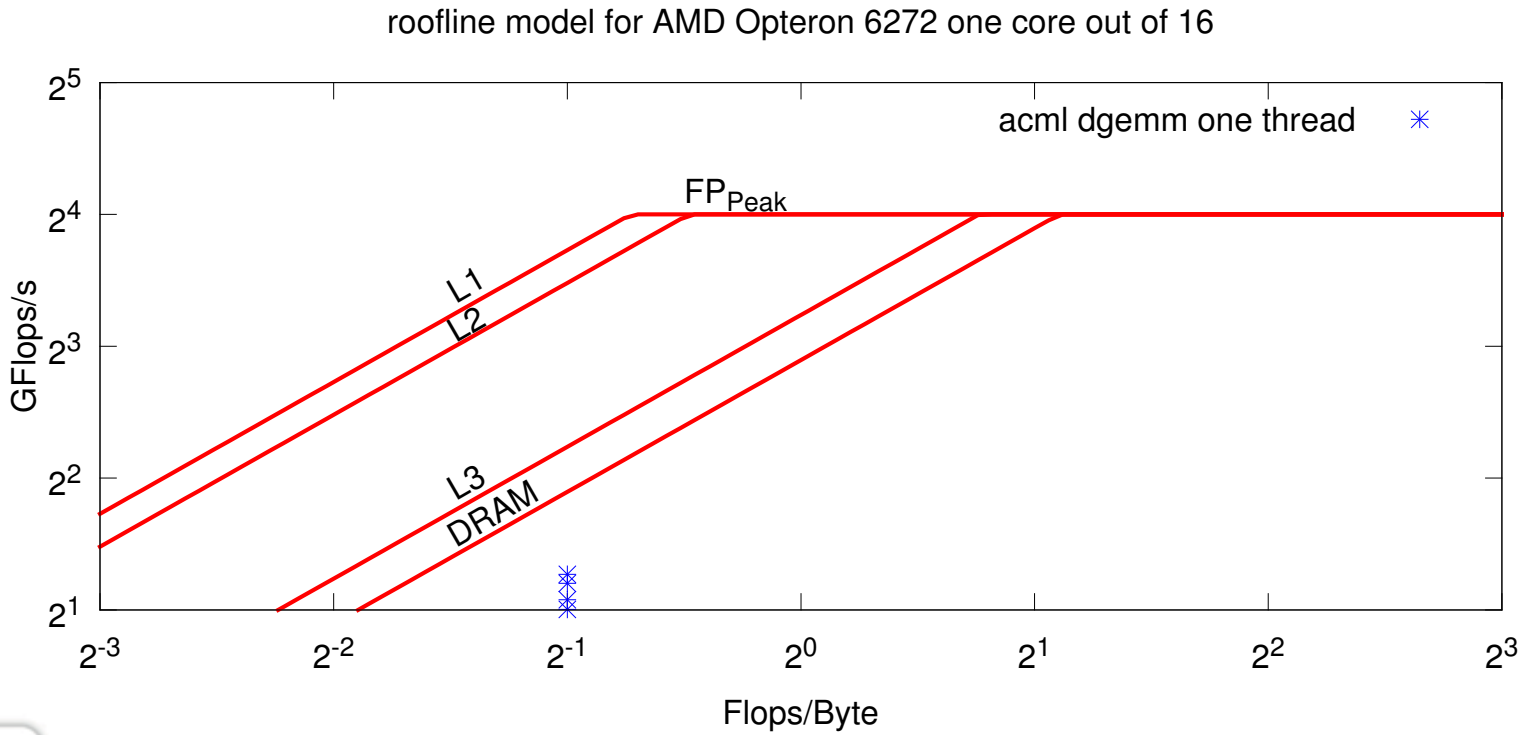
cg	time
no bind	42.77
0,1,2,3	45.34
0,2,4,6	34.90
0,8,16,24	33.12
0,16,32,48	32.94

3
Modèle quantitatif d'application pour le placement de tâches

Modèle d'application pour le placement de tâches

ddot (BLAS1) memory-bound

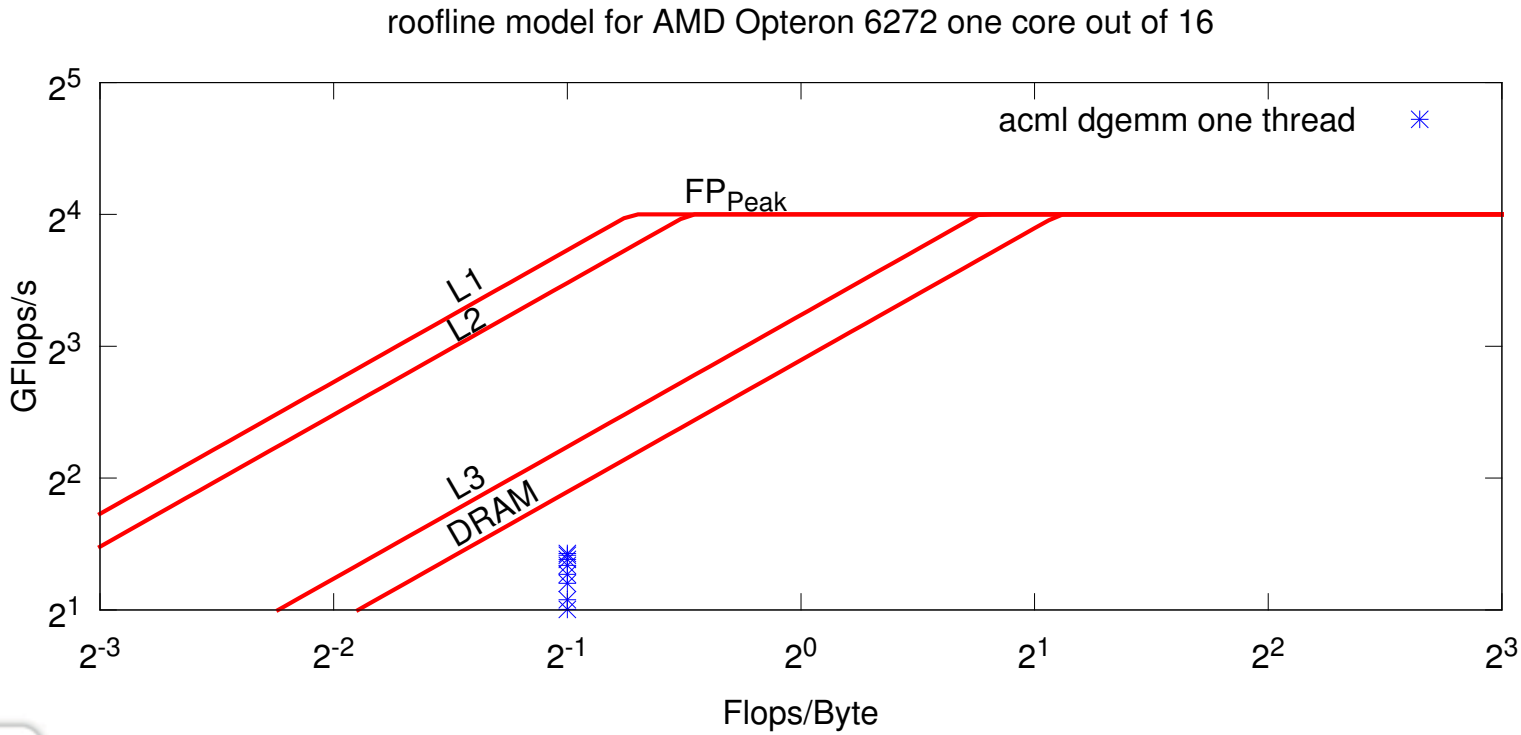
for $i \leftarrow 0 \ldots n-1$ **do**
  $result \leftarrow result + array_A[i] * array_B[i]$
end



Modèle d'application pour le placement de tâches

ddot (BLAS1) memory-bound

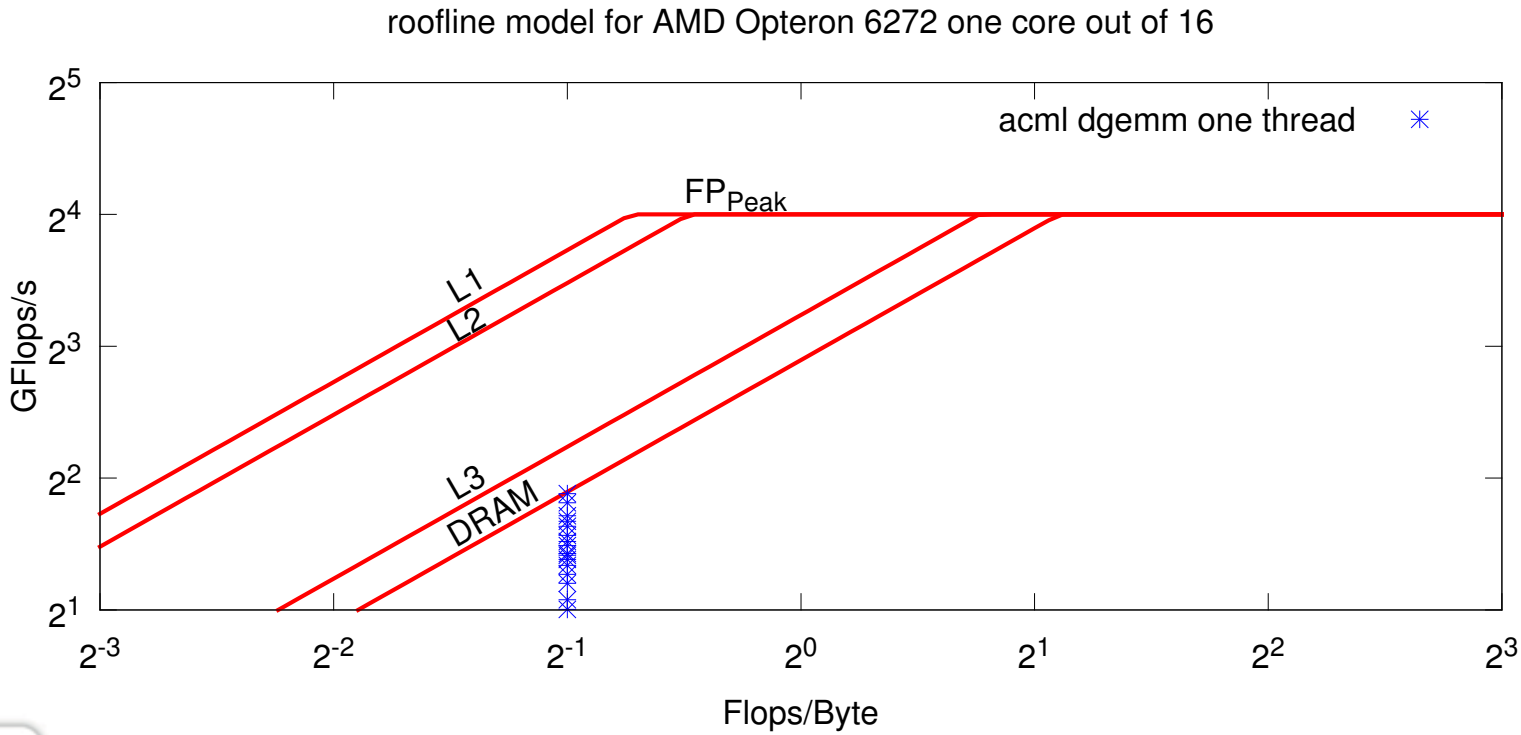
for $i \leftarrow 0 \ldots n-1$ **do**
  $result \leftarrow result + array_A[i] * array_B[i]$
end



Modèle d'application pour le placement de tâches

ddot (BLAS1) memory-bound

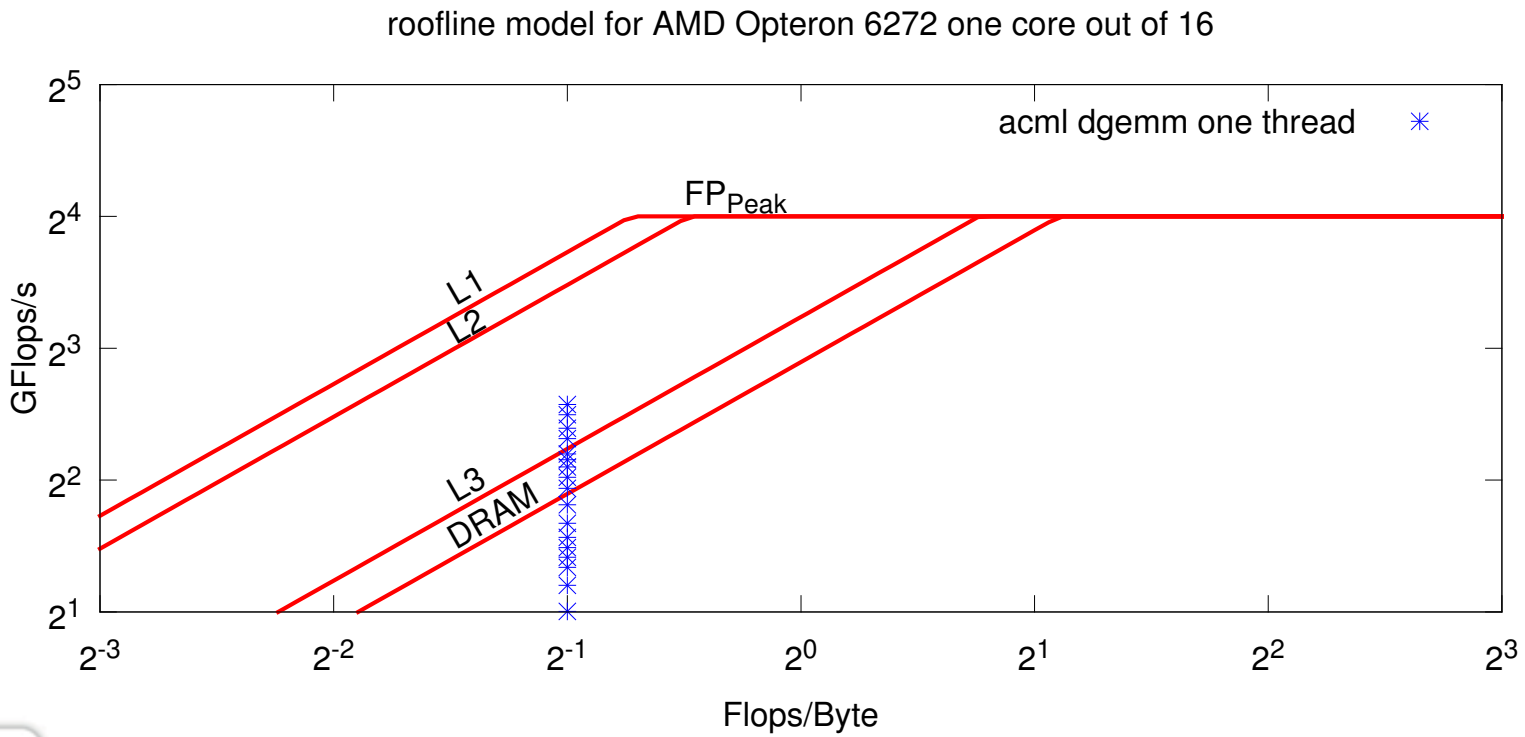
for $i \leftarrow 0 \ldots n-1$ **do**
  $result \leftarrow result + array_A[i] * array_B[i]$
end



Utilisation des BLAS pour notre modèle

dgemm (BLAS3) compute-bound

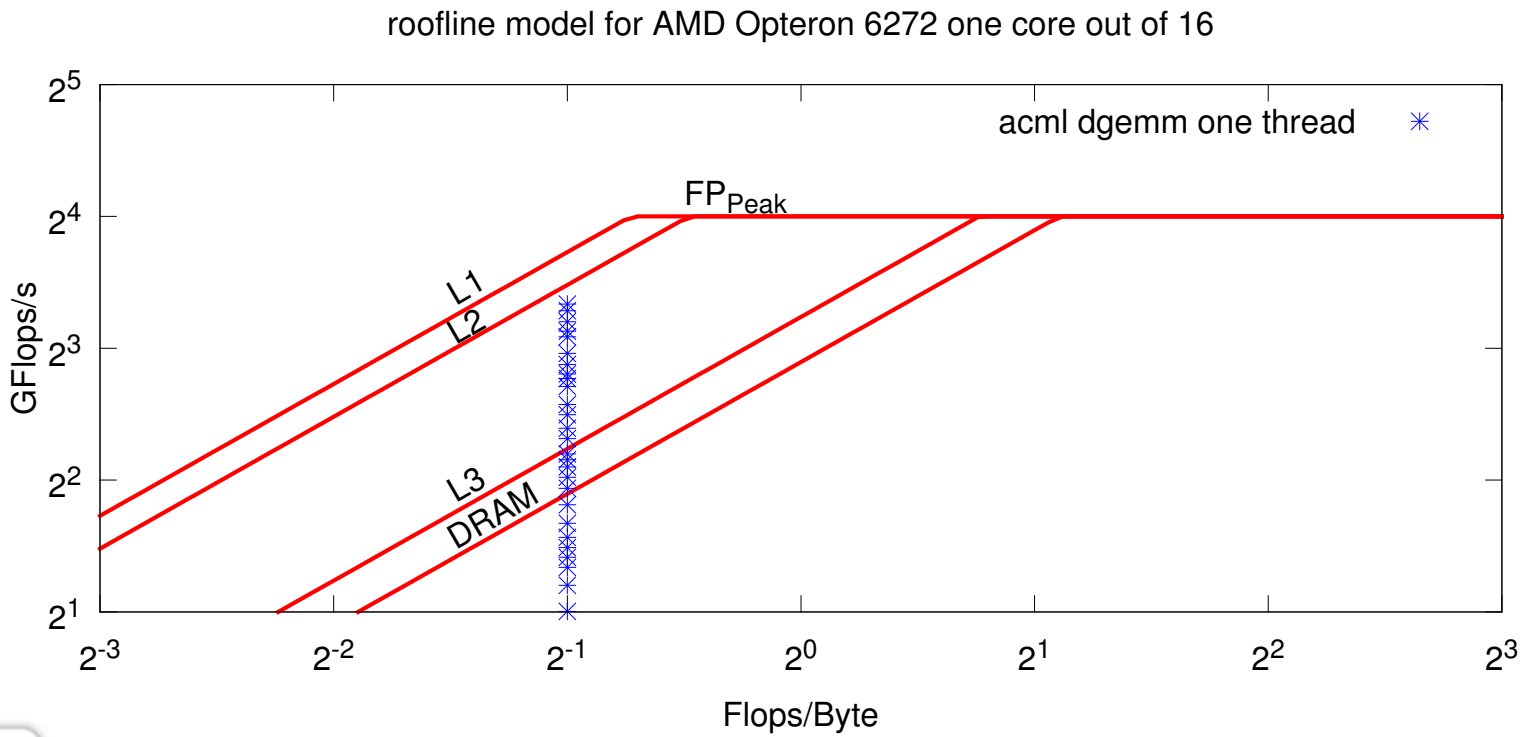
for $i \leftarrow 1 \ldots N, j \leftarrow 1 \ldots M, k \leftarrow 1 \ldots K$ **do**
$\quad C[i][j] \leftarrow C[i][j] + A[j][k] * B[k][i]$
end



Utilisation des BLAS pour notre modèle

dgemm (BLAS3) compute-bound

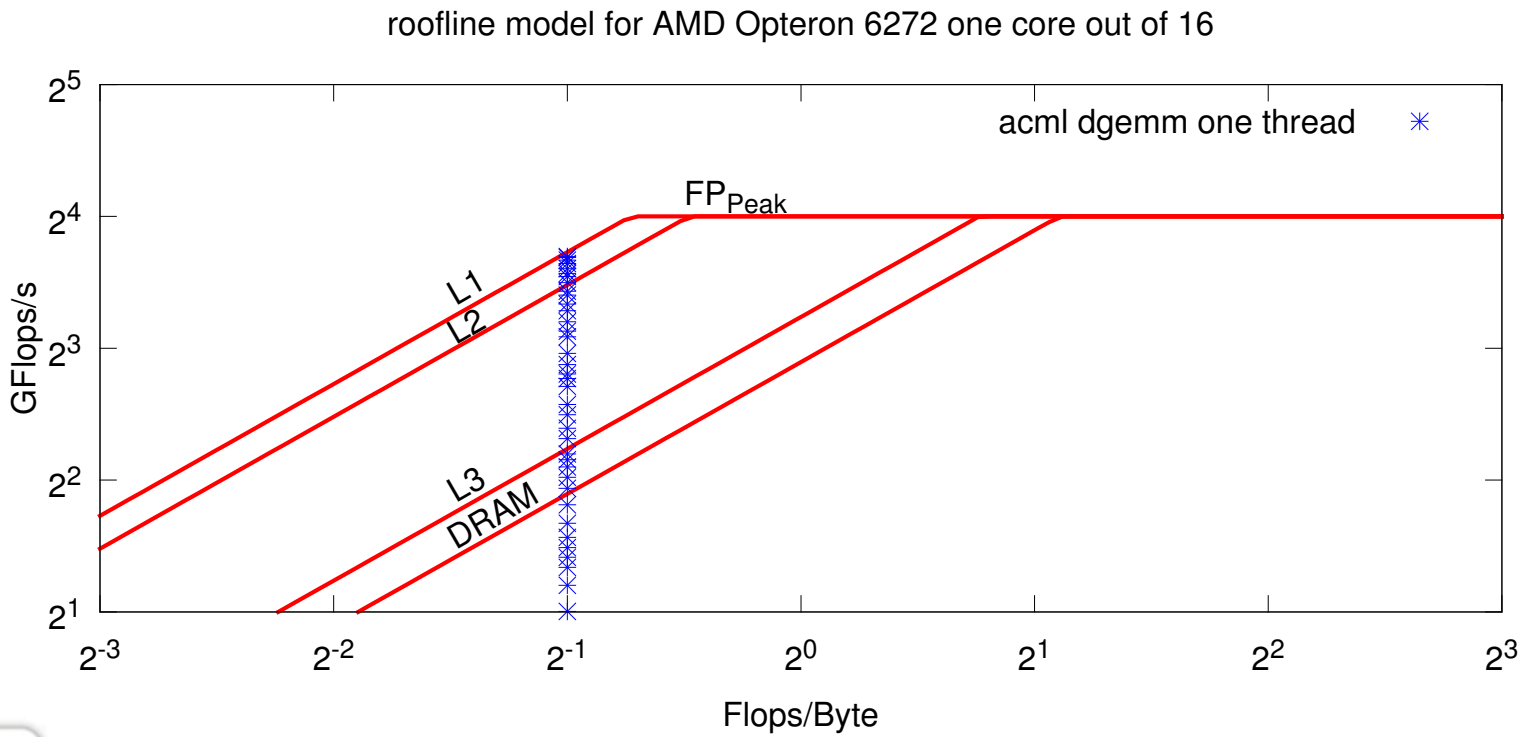
for $i \leftarrow 1 \ldots N, j \leftarrow 1 \ldots M, k \leftarrow 1 \ldots K$ **do**
$\quad C[i][j] \leftarrow C[i][j] + A[j][k] * B[k][i]$
end



Utilisation des BLAS pour notre modèle

dgemm (BLAS3) compute-bound

for $i \leftarrow 1 \ldots N, j \leftarrow 1 \ldots M, k \leftarrow 1 \ldots K$ **do**
$\quad C[i][j] \leftarrow C[i][j] + A[j][k] * B[k][i]$
end



Hypothèses

Sur les conditions de l'expérience

- Les tâches sont indépendantes.

Hypothèses

Sur les conditions de l'expérience

- Les tâches sont indépendantes.
- Pour des tâches dgemm ou ddot l'intensité opérationnelle est constante et égale à 0,5

Hypothèses

Sur les conditions de l'expérience

- Les tâches sont indépendantes.
- Pour des tâches dgemm ou ddot l'intensité opérationnelle est constante et égale à 0,5

Hypothèses

Sur les conditions de l'expérience

- Les tâches sont indépendantes.
- Pour des tâches dgemm ou ddot l'intensité opérationnelle est constante et égale à 0,5

Sur le résultat

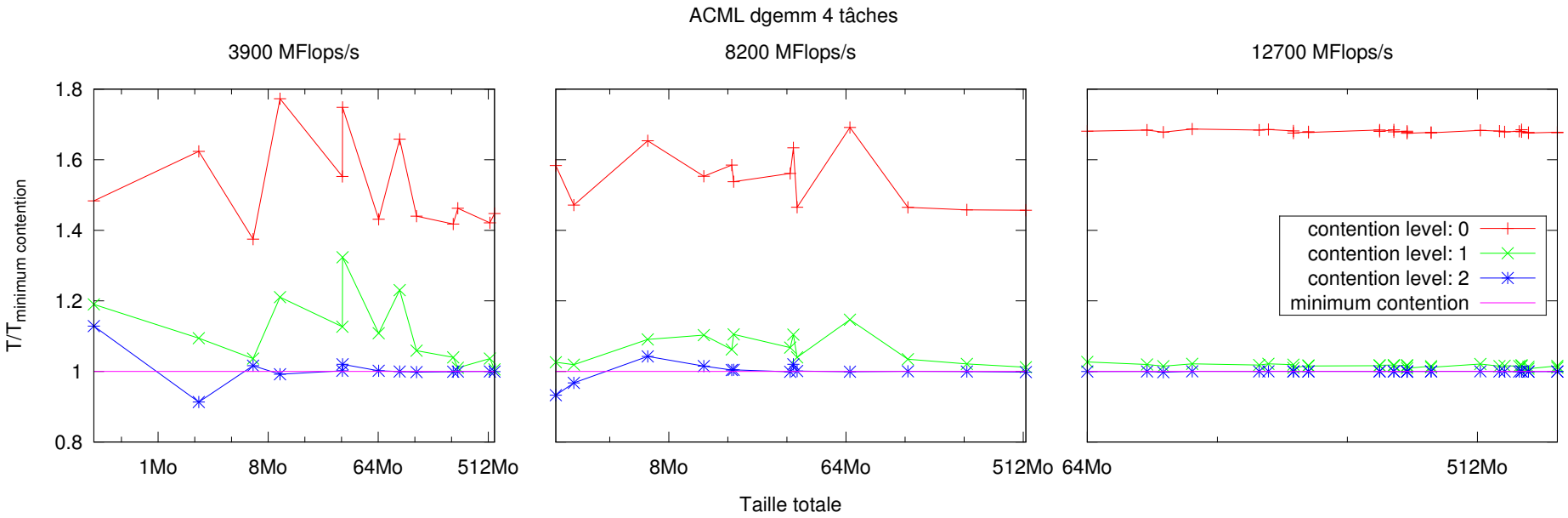
- Intensité opérationnelle=constante & performance=constante $\Longrightarrow$ perte=constante

Méthodologie

Pour chaque dgemm dont on a mesuré la performance précédemment :

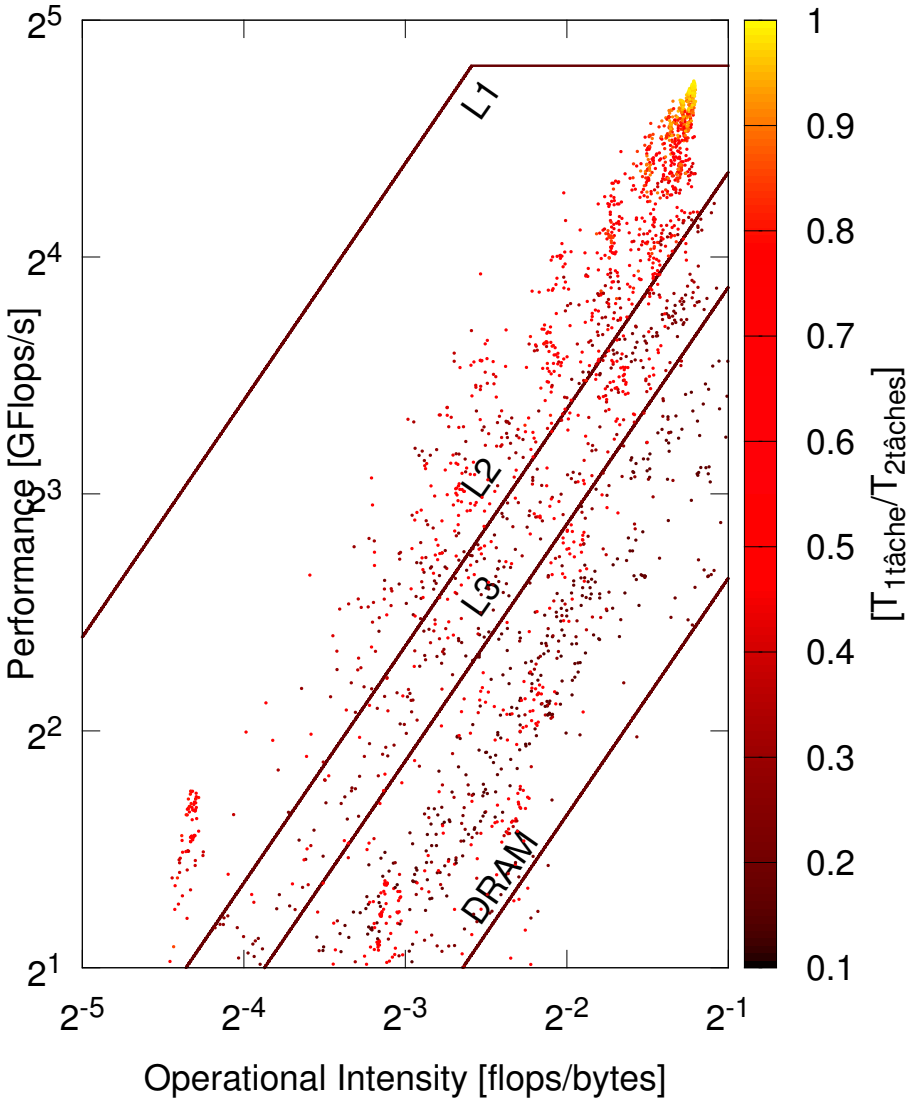
- On exécute 4 tâches de la même application,
- plusieurs fois en écartants les tâches un peu plus à chaque fois.
- On note le temps d'exécution divisé par le temps atteint avec un minimum de contention.

Résultats



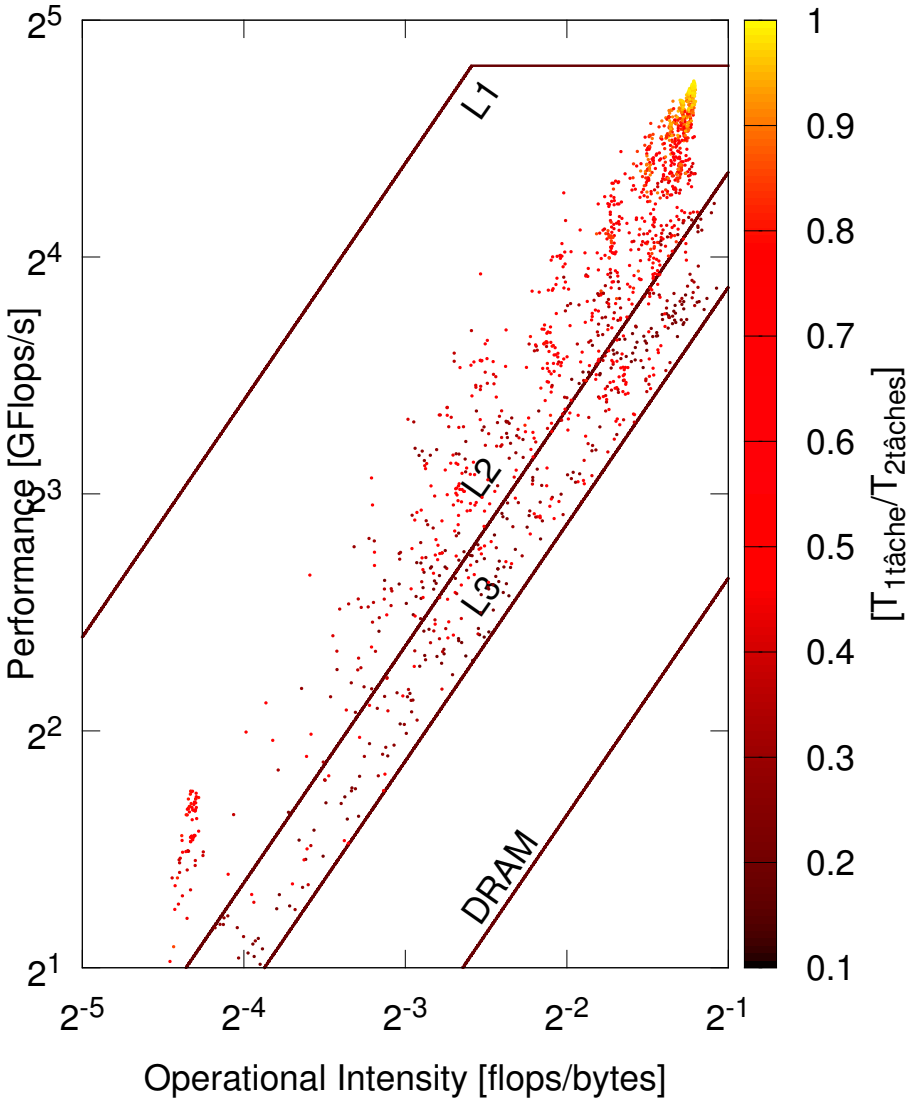
- Intensité opérationnelle=constante & performance=constante
  $\Longrightarrow$ perte=constante ✗

Résultats



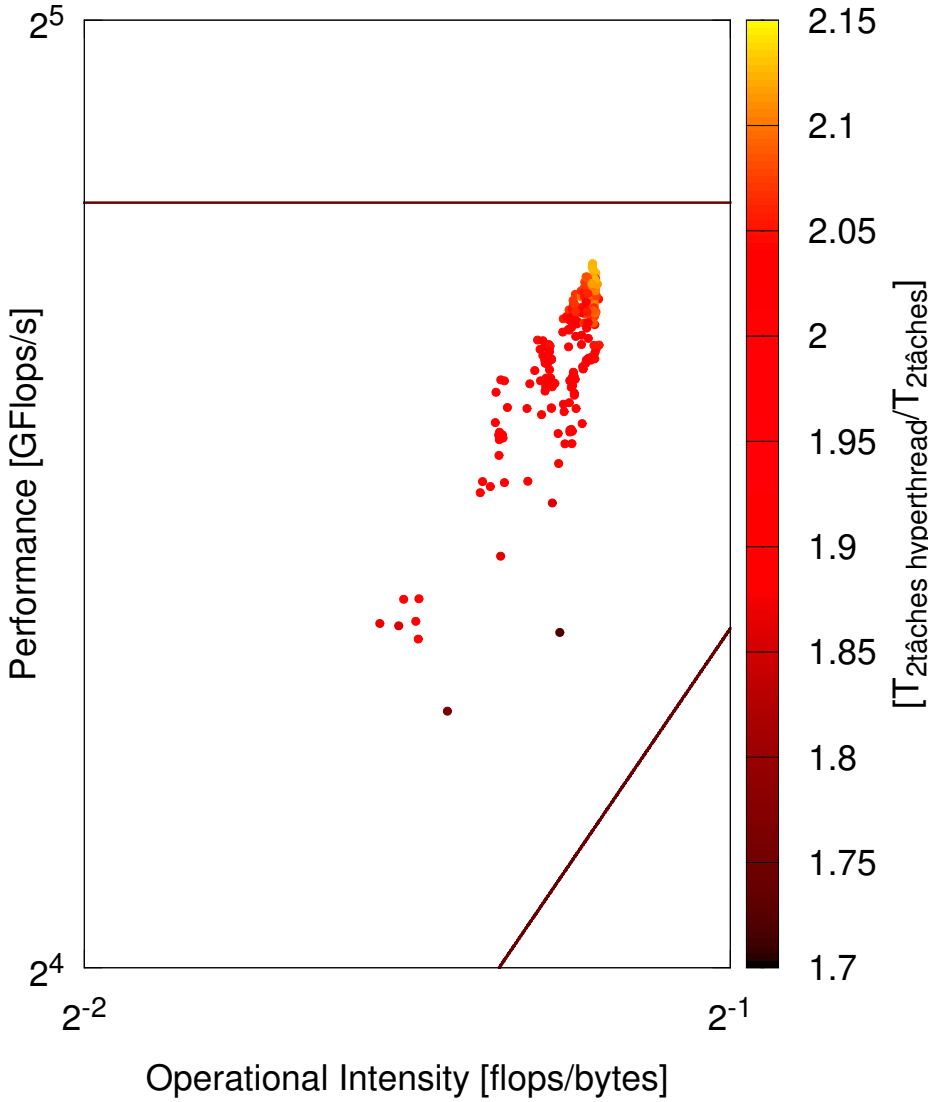
L'intensité opérationnelle est constante ✗.

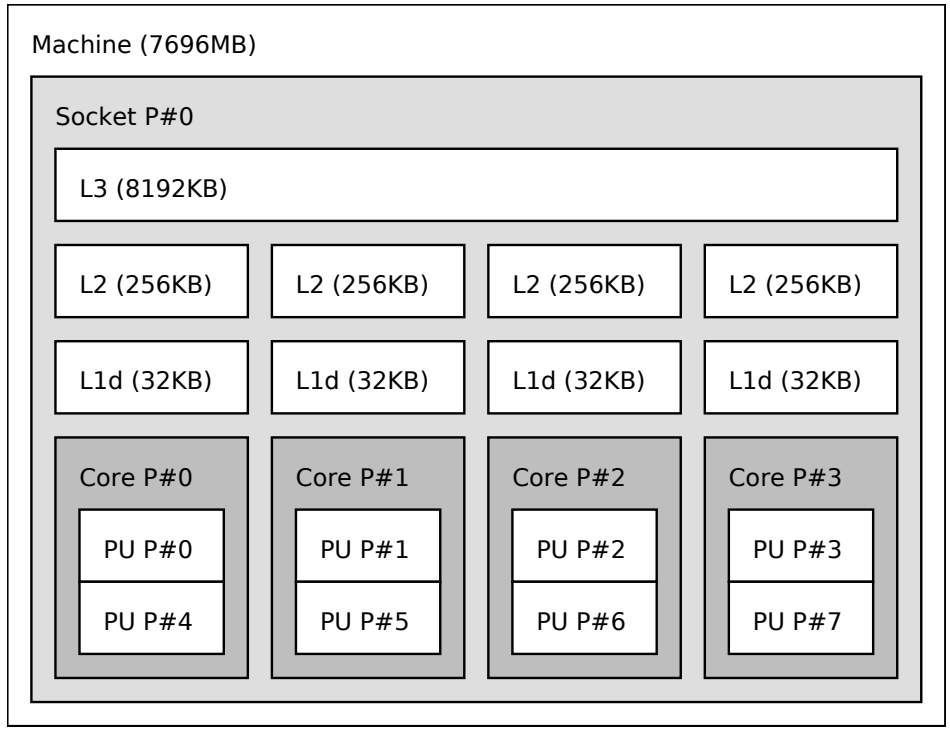
Résultats



- Pas de concurrence sur la bande passante
- Pas de concurrence sur le cache
  $\Longrightarrow \quad T_{1tâche} / T_{2tâches} \simeq 1$

Résultats


2^5
2^4
2^{-2}
2^{-1}
Performance [GFlops/s]
Operational Intensity [flops/bytes]
2.15
2.1
2.05
2
1.95
1.9
1.85
1.8
1.75
1.7
$[T_{2\text{tâches hyperthread}}/T_{2\text{tâches}}]$


Machine (7696MB)
Socket P#0
L3 (8192KB)
L2 (256KB)
L2 (256KB)
L2 (256KB)
L2 (256KB)
L1d (32KB)
L1d (32KB)
L1d (32KB)
L1d (32KB)
Core P#0
PU P#0
PU P#4
Core P#1
PU P#1
PU P#5
Core P#2
PU P#2
PU P#6
Core P#3
PU P#3
PU P#7

5 Conclusion

et perspectives

Conclusion

- Réduire les conflits de caches permet d'améliorer significativement les performances
- Ce critère semble plus important que l'affinité.
- Notre modèle est à perfectionner :
  - Mesure de l'intensité opérationnelle sur une machine hierarchique plus simple.
  - Partage de la bande passante vs partage des caches.
  - Couvrir un panel d'applications plus large.

Perspectives

Il faut établir un modèle quantitatif pour prévoir le gain en optimisant le placement par rapport aux affinités, en prenant en compte le gains sur la bande passante, la synchronisation de tâches et le partage de données.

Inria développe un outil de placement de processus par affinités. On pourrait l'améliorer en mettant des poids négatifs aux affinités entre les tâches qui provoquent de la contention pour les éloigner au lieu de les rapprocher.

MERCI


Bordeaux INP
ENSEIRB
MATMECA
informatiques mathématiques
Inria