TD/TP 4 : diffusion de la chaleur

Version du 20 février 2020

1 Introduction

Comme vous le savez, les microprocesseurs chauffent. En 2020, les plus puissants absorbent plus de 200 Watts et les libèrent sous forme de chaleur. Évacuer cette chaleur est un problème critique, faute de quoi l'élévation de la température détruirait le processeur. Pour simplifier, si le processeur dépasse une certaine température T_{max} (par exemple 100°C), alors il est détruit.

La *densité de flux thermique* (c.a.d. le nombre de Watts transférés par mètre-carré) des processeurs est très élevée. Un CPU haut-de-gamme moderne peut développer $280\,\mathrm{W}$ sur $10\,\mathrm{cm}^2$, ce qui fait $280\,\mathrm{kW/m}^2$. C'est dix fois plus qu'une plaque de cuisson, et c'est à peine deux fois moins qu'une barre de combustible dans un réacteur nucléaire.

Pour cette raison, des *dissipateurs thermiques* en métal sont plaqués contre les processeurs. Ils permettent le transfert efficace de la chaleur depuis le processeur vers un milieu externe. Dans les machines « grand public », un ventilateur permet d'évacuer la chaleur du dissipateur métallique par convection forcée [1]. Dans les centres de calcul, le dissipateur thermique peut être refroidi par un fluide (*water-cooling*). Enfin, dans le cas de processeurs peu puissants, le phénomène de convection naturelle [2] peut suffire à évacuer la chaleur du dissipateur.

On admet ici que le dissipateur est un parallélépipède rectangle (un « cuboïde »). Sa surface, son épaisseur et le choix de matière dans laquelle il est réalisé affectent ses caractéristiques thermiques.

Ce TP consiste à paralléliser une simulation numérique qui détermine la température atteinte par un CPU récent (un AMD EPYC « rome ») sur lequel on pose une plaque d'aluminium de $15\,\mathrm{cm} \times 12\,\mathrm{cm} \times 0.8\,\mathrm{cm}$. La face $z = 0$ du dissipateur est maintenue de manière forcée à $T = 20\,^\circ\mathrm{C}$, et on suppose que l'ensemble du dissipateur est initialement à cette température. On suppose que l'air ambiant est aussi à $20\,^\circ\mathrm{C}$. L'idée générale consiste à allumer le processeur au temps $t = 0$ et à simuler la diffusion de la chaleur dans le dissipateur jusqu'à ce qu'un régime à peu près stationnaire soit atteint.

Le processeur va-t-il surchauffer ? Et si on réduisait/augmentait l'épaisseur/la surface du dissipateur ? Et si on remplaçait l'aluminium par du cuivre, du fer ou... de l'or ?

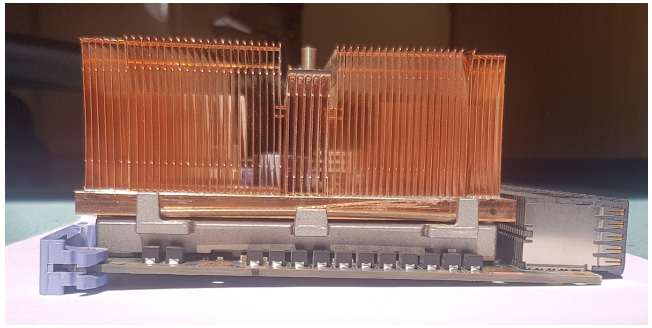

FIGURE 1 – Un dissipateur thermique en cuivre (avec des ailettes) posé sur une carte contenant un CPU.

1. Pour augmenter la surface de contact avec le flux d'air et maximiser le transfert de chaleur, les dissipateurs thermiques ont alors généralement des ailettes
2. L'air en contact avec le dissipateur absorbe de la chaleur, chauffe, monte et fait de la place à de l'air frais.

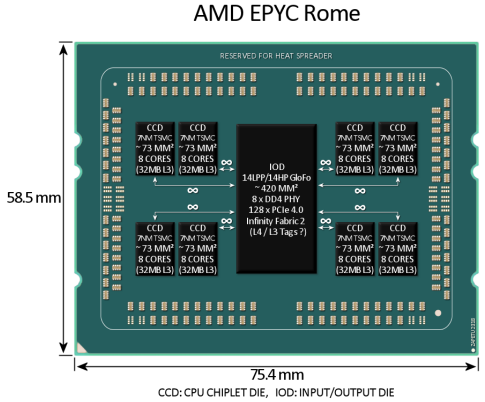

FIGURE 2 – Le CPU sous le dissipateur simulé. Taille réelle. Puissance dissipée : 280W. Seule les parties noires sont en contact avec le dissipateur

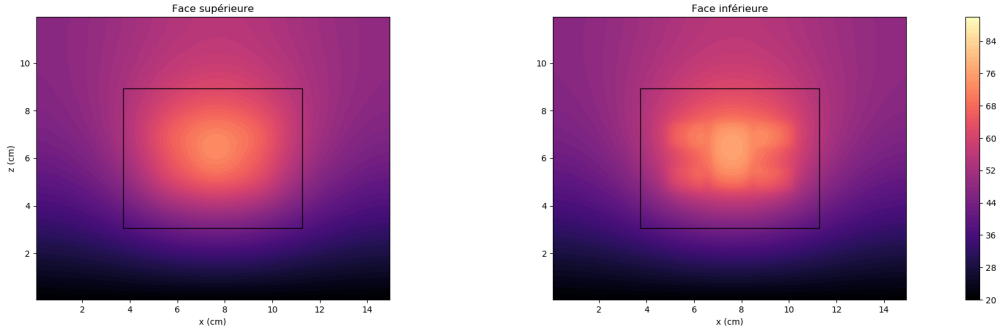

FIGURE 3 – Résultat de la simulation numérique en régime stationnaire.

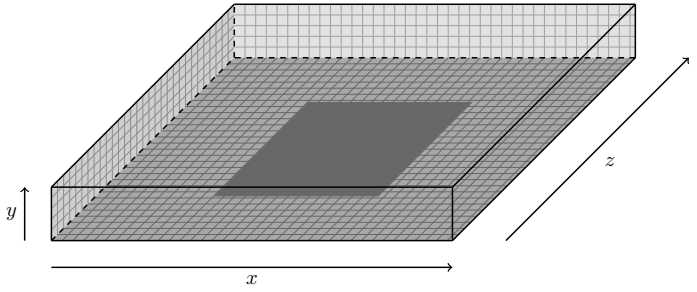

FIGURE 4 – Représentation schématique du dissipateur thermique.

2 Principe physique de la simulation numérique

2.1 Généralités

Il ne faut pas confondre *chaleur* et *température*. La chaleur est un transfert d'énergie thermique. Lorsque deux corps sont mis en contact, un échange d'énergie thermique a lieu, jusqu'à ce que les deux corps se trouvent à la même température. La température se mesure en degrés Celsius (°C) ou en degrés Kelvin (K, avec $0\,\mathrm{K} = -273.1\,^\circ\mathrm{C}$), tandis que les quantités d'énergie (thermique ou autre) se mesurent en joules (J).

Lorsqu'il y a des échanges d'énergie thermique, il est commode d'en mesurer le « débit ». Le *flux thermique* est la quantité d'énergie transférée par unité de temps, et il s'exprime en Watts (W) — c'est l'équivalent d'une puissance. Par exemple, si un flux thermique constant Φ est transféré pendant t secondes, la quantité d'énergie thermique transférée est $Q = \varphi t$. Dans le cas de notre CPU, le flux thermique du processeur vers le dissipateur est de $\Phi = 280\,\mathrm{W}$.

Lorsqu'un transfert thermique a lieu à travers une certaine surface, on peut aussi examiner comment le « débit » du flux thermique varie le long de cette surface. La *densité de flux thermique* φ indique quel flux thermique passe à travers un point donné de la surface. S'il est constant, alors le flux thermique total est $\Phi = \varphi S$. Toujours dans le cas de notre processeur, sa surface de contact avec le dissipateur thermique est d'environ $10\,\mathrm{cm}^2$, ce qui donne $\varphi = 280\,\mathrm{kW\,m^{-2}}$.

Lorsqu'un matériaux reçoit de l'énergie thermique, sa température augmente. Sa *capacité thermique massique*, généralement notée c, est la quantité d'énergie (en J) qu'il faut lui apporter pour que la température de 1kg de matière augmente de 1 degré. Par exemple, on a $c = 897\,\mathrm{J\,kg^{-1}\,K^{-1}}$ pour l'aluminium, donc si on pose un bloc d'alu d'un kilo sur notre processeur, alors on sait qu'il va chauffer de $\approx 0.31\,^\circ\mathrm{C}$ par seconde (c'est $280/897$). Sans refroidissement externe, il passera de $20\,^\circ\mathrm{C}$ à $100\,^\circ\mathrm{C}$ en 256.3 secondes.

2.2 Transferts thermiques

On suppose que quatre types de transfert thermiques distincts sont à l'oeuvre. Dans chacun de ces cas, le flux thermique (le « débit » du transfert) est proportionnel à la surface sur laquelle le transfert se réalise. On se contente donc de donner la densité de flux thermique.

1. La *conduction* thermique du CPU vers le dissipateur. Elle a lieu uniquement aux endroit où le CPU est en contact (qui sont tous dans le plan $z = 0$). On a déjà calculé que $\varphi = 2.8 \times 10^5\,\mathrm{W\,m^{-2}}$.
2. La *conduction* thermique au sein du dissipateur : la chaleur se propage dans le métal, des parties chaudes vers les parties froides. Ce transfert obéit à la loi de Fourier : $\vec{\varphi} = -\lambda \cdot \overrightarrow{\mathrm{grad}}\ T$, où λ (la *conductivité thermique*) caractérise la capacité du milieu à conduire la chaleur : plus λ est grand, plus le transfert thermique est rapide. La conductivité s'exprime en Watts par mètre par Kelvin et représente le flux thermique qui circule entre deux points lorsque le gradient de température est de 1 degré par mètre.

   Plus concrètement, si on a une paroi d'épaisseur e dont les deux côté sont de température homogène T_1 et T_2, alors la densité de flux thermique en tout point de la paroi (de 1 vers 2) est $\varphi = \lambda(T_1 - T_2)/e$.
3. La *convection* sur les bords du dissipateur : le métal chauffe l'air ambiant qui entre en mouvement et évacue la chaleur. Si une paroi solide de surface S et de température T_1 est en contact avec un fluide (l'air) de température T_2, la densité de flux thermique est $\varphi_{1\to 2} = h(T_1 - T_2)$, où h est le *coefficient de convection thermique* du matériau. On observe expérimentalement que $h \approx 10\,\mathrm{W\,m^{-2}\,K^{-1}}$.
4. Le *rayonnement* sur les bords du dissipateur : le métal émet un rayonnement électromagnétique dont la fréquence dépend de la température (il peut appartenir au spectre de la lumière visible : métal « chauffé à blanc »). Sous des hypothèses raisonnables (rayonnement du corps noir), la densité de flux thermique émise vers l'extérieur est $\varphi = \sigma T^4$, où σ est la constante de Stefan-Boltzmann (elle vaut $5.6703 \times 10^{-8}\,\mathrm{W\,m^{-2}\,K^{-4}}$; attention la température doit être en degré Kelvin).

La simulation numérique proposée prétend à un certain réalisme, ceci dit elle n'est pas parfaite : la température du milieu est supposée fixe, l'hypothèse du corps noir est une simplification, et il est supposé que le dissipateur ne reçoit pas d'énergie thermique du milieu ambiant par rayonnement, ce qui est impossible.

2.3 Schéma de la simulation numérique

Pour simuler tous ces transferts thermiques, on discrétise le temps et l'espace. On découpe le dissipateur en *cellules* (« petits cuboïdes ») de taille $\Delta_x \times \Delta_y \times \Delta_z$, et on suppose que pendant un pas de temps suffisamment court, la température

1: **procedure** SIMULATION(...)
2: Allouer deux tableaux R et T de taille nmo
3: $T[:] \leftarrow 293.15$ ▷ toutes les cellules sont initialement à 20 °C
4: $t \leftarrow 0$
5: `convergence` $\leftarrow 0$
6: **while** `convergence` $= 0$ **do**
7: **for** $0 \le k < o$ **do** ▷ Traite le k-ème plan xy
8: **for** $0 \le j < m$ **do**
9: **for** $0 \le i < n$ **do**
10: $R[knm + jn + i] \leftarrow$ UPDATETEMP(i, j, k) ▷ Accède potentiellement aux 6 cellules voisines
11: **end for**
12: **end for**
13: **end for**
14: **if** t est entier **then**
15: $T_{max} = \max R[:]$
16: $\varepsilon = \sum_{u=0}^{omn} (R[u] - T[u])^2$
17: **if** $\sqrt{\varepsilon}/\Delta_t < 0.1$ **then**
18: `convergence` $\leftarrow 1$
19: **end if**
20: Afficher t et T_{max} pour faire patienter l'utilisateur...
21: **end if**
22: $t \leftarrow t + \Delta_t$
23: $T \leftarrow R$
24: **end while**
25: Sauvegarder T dans un fichier pour le rendu graphique ultérieur.
26: **end procedure**

FIGURE 5 – Principe de la simulation

est homogène au sein de chaque cellule. On effectue des pas de temps suffisamment petits pour que la propagation de l'énergie thermique soit relativement lente, et en particulier pour qu'elle ne puisse pas « franchir » plusieurs cellules à la fois (donc plus le métal est un bon conducteur thermique plus il faut faire de petits pas de temps).

On note $T(i, j, k)$ la température de la cellule de coordonnées (i, j, k) au temps t. Le problème consiste à calculer la température de toutes les cellules à l'instant $t + \Delta_t$.

Pour cela, il faut déterminer le flux thermique Φ reçu par chaque cellule, et alors on pourra calculer la variation de sa température grâce à la capacité thermique massique du métal et à la masse des cellules — heureusement on connaît la densité ρ du métal et le volume des cellules. On aura donc $\Delta T = \Delta Q/(c\Delta_x\Delta_y\Delta_z\rho)$.

Les cellules qui sont sur le bord cèdent de l'énergie au milieu extérieur par rayonnement. Par exemple, sur la face $z = 1$, une cellule de température T transfère vers l'extérieur un flux thermique $\Delta_x\Delta_y\sigma T^4$ Watts (en effet, la surface d'échange est $\Delta_x\Delta_y$).

Les cellules qui sont sur le bord cèdent aussi de l'énergie au milieu extérieur par convection. Par exemple, sur la face $x = 0$, une cellule de température T transfère $\Delta_y\Delta_z h(T - 293.1)$ Watts.

Les cellules en contact avec le CPU en reçoivent chacune un flux thermique de $\Delta_x\Delta_y 2.8 \times 10^5$.

Enfin, une cellule échange de l'énergie thermique avec chacun de ses voisines. La cellule a reçoit de la cellule b (sa voisine de droite sur l'axe x, disons) un flux thermique $\lambda\Delta_y\Delta_z(T_b - T_a)/\Delta_x$.

2.4 Implémentation

Pour simplifier un peu les choses, on suppose que les cellules sont cubiques ($\Delta_x = \Delta_y = \Delta_z$). Dans le programme, on note `dl` la taille du côté des cellules.

Représentation en mémoire Une des petites difficulté est qu'il faut représenter le tableau tridimensionnel T en mémoire. Supposons qu'il soit de dimension $n \times m \times o$ (selon les axes x, y, z). On l'aplatit sur une seule dimension en utilisant la convention suivante (*row-major*, habituelle dans le langage C) : les plans xy sont représentés de manière contiguë en mémoire, et au sein de ces plans les lignes x sont représentées de manière contiguë. Autrement dit, si la température de la cellule (i, j, k) est dans $T[u]$, alors :

— $T[u + 1]$ contient la température de la cellule $(i + 1, j, k)$;
— $T[u + n]$ contient la température de la cellule $(i, j + 1, k)$;
— $T[u + nm]$ contient la température de la cellule $(i, j, k + 1)$.

Enfin, la température de la cellule (i, j, k) est donc dans $T[i + nj + knm]$. La simulation fonctionne selon le pseudo-code de la figure 5.

3 Questions

1. Pourquoi doit-on écrire les nouvelles températures dans un tableau intermédiaire R au lieu de faire le calcul directement dans T ?
2. Quelles sont les séquences parallélisables de l'algorithme ?
3. Quel type d'équilibrage de charge doit-on prévoir entre les processeurs ?
4. Quel(s) découpage(s) (répartition des données entre processeur) est naturel dans ce contexte ? Quel est le plus simple à mettre en oeuvre ?
5. Quelle(s) opération(s) collective(s) utiliser pour gérer le calcul de T_{max} et de ε ? Et la sauvegarde de T à la fin ?
6. Implémenter un algorithme parallèle avec des envois bloquants de messages, en utilisant `MPI_Send` et `MPI_Recv` pour l'échange des « halos ». On peut supposer (pour se simplifier la vie) que le nombre de processus divise o.
7. Même question en utilisant `MPI_Sendrecv`. C'est celui-là qu'il faudra implanter en TP.
8. Question subsidiaire : implémenter l'algorithme avec cette fois des primitives non-bloquantes et de façon à ce que les temps de calcul recouvrent les temps de communication. Analyser les performances obtenues.
9. Question super-subsidiaire : implémenter un autre découpage des données.