

Palabos, ordinateur parallèle, Paraview

Nguyen Phuc-Thien Thomas

15 mai 2013

Résumé

Ce document a été rédigé dans le cadre de mon stage à l'Université de Genève. Plus précisément au centre universitaire d'informatique, où on travaille notamment sur la simulation numérique de divers phénomènes physiques ou biologiques. C'est la suite du chapitre sur les automates cellulaires et lattice Boltzmann.

Jusque là, j'ai eu à coder des automates cellulaires : le jeu de la vie, la règle de parité, le modèle HPP et lattice Boltzmann, afin de bien en comprendre le principe. J'ai pu aussi lire des la documentation sur la mécanique des fluides, mais sans entrer dans les détails. Et puis, j'ai appris à utiliser l'ordinateur parallèle qui fera les simulations et les logiciels nécessaires. En outre, j'ai oublié de dire que tout ce qui a été et est fait se passe sous Linux (Ubuntu), système auquel je me suis familiarisé, puisqu'étant un (maintenant ancien) Windowsien.

Ici sera présenté le logiciel Palabos, sur lequel est basé les simulations numériques sur l'anévrisme dans le projet *Thrombus* auquel je contribue. Il a été écrit en grande partie par deux de mes assistants, y compris pour les exemples (divers écoulements et simulation de l'anévrisme), les docteurs Lätt Jonas et Malaspinas Orestis. Il y aura également une description du superordinateur qui fera les calculs, et de Paraview, un visualiseur de données.

1 Introduction

Palabos est un logiciel open-source, qui permet de simuler l'écoulement de fluides dans tout types de cas, comme l'écoulement dans une cavité, l'écoulement avec différentes températures dans le fluide, l'écoulement multiphase (par exemple liquide et gaz), l'écoulement avec particules, etc. Il est basé sur la méthode de lattice Boltzmann.

Nous allons ici expliquer pas à pas l'utilisation de ce logiciel, et aussi l'exécution sur un ordinateur parallèle, Scylla. Et nous allons également utiliser un logiciel d'analyse de données appelé Paraview.

2 Prise en main

2.1 Téléchargement

Les sources sont disponibles sur le site officiel [1], dans la section *Software*. Télécharger, puis décompresser quelque part la version la plus récente (v 1.3.0 à l'heure actuelle).

2.2 Quelques commandes Linux à connaître

Ces commandes de base sont fournies à titre indicatif pour ceux qui feraient leur débuts sur Linux. Note : ne pas mettre les chevrons `<>`. Cela indique qu'il faut mettre un paramètre.

- Lister le contenu d'un dossier : `ls`
- Choisir un répertoire : `cd <répertoire>`
 - Exemples : `cd Dossier`, `cd Desktop`
- Aller dans le dossier parent : `cd ..`
- Exécuter un programme : `./<programme>`
 - Exemple : `./prog`
- Compiler des sources dans un dossier : `make`
- Exécuter de manière parallèle : `mpirun -np <nombre processus> <programme>`
 - Exemple : `mpirun -np 4 make`

2.3 Compilation et exécution d'exemples

Ceci se fait simplement en utilisant les commandes ci-dessus. Par exemple, exécutons l'écoulement dans une cavité 2D :

- Ouvrir un terminal et aller dans le dossier Palabos à l'aide de la commande `cd` (et `ls` si besoin)
- Cet exemple se trouve dans le répertoire suivant : `examples/showCases/cavity2d`. Utiliser `cd` pour y accéder :
 - `cd examples/showCases/cavity2d`
 - ou bien faire successivement `cd examples`; `cd showCases`; `cd cavity2d`
- Compiler le code à l'aide de `make`. Il est possible de le faire en parallèle avec la commande `mpirun` si les ressources sont disponibles, pour plus de rapidité.
- Après quelques minutes, le programme est créé dans le même répertoire, sous le nom de `cavity2d`. Il peut être exécuté en faisant `./cavity2d` (fonctionne aussi avec `mpirun`).
- Le programme s'exécute et fait les simulations. Des images représentant l'état du système sont périodiquement créées dans le dossier `tmp` du répertoire. On peut soit attendre que la simulation s'arrête (certains programmes ne s'arrêteront jamais, mais celui-ci s'arrêtera au temps $t = 10$), soit arrêter le programme en faisant `Ctrl+C`.

Des fichiers avec l'extension `.vti` sont également créés. Ils contiennent des données du fluides, par exemple les vitesses en tout point donné à un moment donné, et peuvent être ouverts par un logiciel appelé Paraview que nous verrons plus tard.

Il est maintenant possible de lancer d'autres exemples proposés dans le dossier `showCases`, la démarche étant sensiblement la même.

3 Benchmark

Palabos propose un benchmark ¹ disponible dans le dossier `examples/benchmarks/cavity3d`. Il permet d'estimer la performance d'un ou des processeurs en affichant le nombre de

1. Test de performance.

millions de sites mises à jour par seconde², autrement dit le nombre de valeurs du domaine de la situation calculées pour chaque seconde.

Ce benchmark se compile et s'exécute de la même manière que les exemples. Il faut juste en plus indiquer lors de l'exécution la résolution du programme (le nombre de cases sur une arête de la cavité simulée).

Il peut être utile pour tester si le processeur parallélise bien. Nous avons lancé ce benchmark sur un ordinateur de bureau (avec Intel Core i7 860), avec un, deux, trois et quatre cœurs :

Cœurs	1	2	3	4
MSUPS	6.47	11.4	13.5	17.1
Ratio	1	1.76	2.08	2.65

Ratio est le rapport entre le résultat pour un nombre donné de cœurs et celui pour un seul. Ainsi, trois cœurs ne fournissent qu'une performance doublée par rapport à un seul. Un graphique se trouve à la 3.

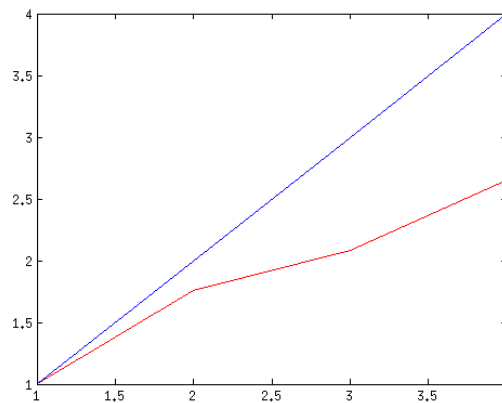


FIGURE 1 – En rouge les performances obtenues, en bleu celles théoriques, si 4 cœurs étaient effectivement 4 fois plus performants qu'un seul.

Nous pouvons constater que la parallélisation est loin d'être parfaite ici, l'amélioration de la performance n'est pas du tout linéaire. Ceci est soit dû au processeur qui ne répartit pas parfaitement les tâches, les cœurs ne communiquent pas de manière optimale, soit au programme qui n'a pas un grand pouvoir de parallélisation.

4 Scylla

4.1 Description

Scylla est un ordinateur parallèle [4], doté de 960 Gio³ de mémoire vive et de 40 processeurs Intel Xeon 5500, qui contiennent eux même chacun six cœurs (donc 240 au total). Ces ressources sont décomposées en 20 nœuds de 2 processeurs, donc de 12 cœurs.

2. Ou MSUPS, mega sites updates per second.

3. 1 Go = 10⁹ o ; 1 Gio = 2³⁰ o $\approx 1.07 \cdot 10^9$ o. Go : gigaoctet ; Gio : gibioctet. Trop de gens font cette confusion et pensent qu'il faut utiliser les puissances de deux, qu'un ko vaut 1024 octets [3] !

4.2 Accès

Il faut au préalable posséder un login et son mot de passe, et faire partie du centre universitaire d'informatique. Dans une session à l'université, l'accès à Scylla se fait à l'aide de la commande `ssh scylla0`. Il faut ensuite entrer son mot de passe, et on a maintenant ouvert une session sous Scylla dans le terminal. Puis, il faut demander un nœud de calcul en tapant `qsub -I`.

4.3 Transfert de données

Pour transférer des fichiers de l'ordinateur de bureau vers Scylla, il faut exécuter la commande suivante : `scp <fichier> scylla0:.`

Dans l'autre sens, il faut utiliser `rsync :rsync -avz scylla0:<source> <destination>`.

Notez que ces opérations se font dans un terminal dont la session est sous l'ordinateur de bureau et non sous Scylla.

4.4 Exécution d'exemples

Maintenant que nous disposons de ressources plus puissantes, il est possible d'exécuter les simulations plus rapidement. Il suffit alors de procéder comme nous l'avons fait sur l'ordinateur de bureau, la démarche reste la même. Il faut cependant veiller d'avoir bien installé un compilateur. Quelques exemples en images :

4.4.1 Écoulement avec température

La situation est tridimensionnelle, mais les images sont en deux dimensions. C'est une situation où un liquide est chauffé en continu par le bas.

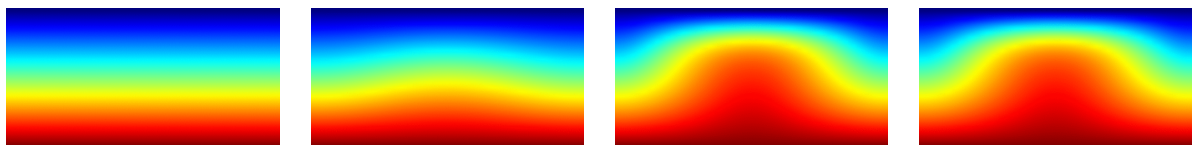


FIGURE 2 – Températures au temps $t = 0$

FIGURE 3 – Au temps $t = 25000$

FIGURE 4 – Au temps $t = 50000$

FIGURE 5 – Au temps $t = 100000$

On peut voir comment la température évolue au cours du temps. Le bleu représente le *froid*, le rouge représente le *chaud*. Au début, la température semble diminuer de manière linéaire par rapport à l'altitude. Puis, la distribution change et converge vers une sorte de bosse.

4.4.2 Écoulement multiphase

Il s'agit d'une situation où un bloc de liquide se situe dans une boîte, avec de l'air et un petit bloc solide (toujours en 3D mais affiché en 2D).

Naturellement, ce liquide s'effondre sous l'effet de la gravité et fera des mouvements de va et vient pendant un certain temps, avant de tendre au repos.

FIGURE 6 – Densités au temps $t = 0$ FIGURE 7 – Au temps $t = 100$ FIGURE 8 – Au temps $t = 1000$ FIGURE 9 – Au temps $t = 1200$ FIGURE 10 – Au temps $t = 10000$

5 Paraview

5.1 Description

Paraview est un logiciel open-source permettant d'analyser des bases de données [2]. Comme évoqué lors d'un chapitre antérieur, des fichiers avec l'extension `.vti` ou `.vtk` peuvent être créés lors des simulations. Ils contiennent des données du fluide, par exemple les températures en tout point donné à un moment donné.

Prise en main, exemple

Ouvrons donc ce programme en exécutant simplement la commande `paraview` dans un terminal. Puis, nous pouvons par exemple ouvrir les fichiers `.vti` de la simulation de l'écoulement avec température en cliquant sur l'icône du dossier tout en haut à gauche. Ensuite, dans la section à gauche, dans l'onglet Properties, appuyez sur Apply. Les arêtes d'un parallélépipède apparaissent alors. Pour régler ce problème, remplacez Outline par Volume dans le menu déroulant en haut. Un volume apparaît alors, qui est désormais coloré en fonction de la température du liquide.

En haut à droite, Paraview propose la possibilité de jouer l'animation de la situation.

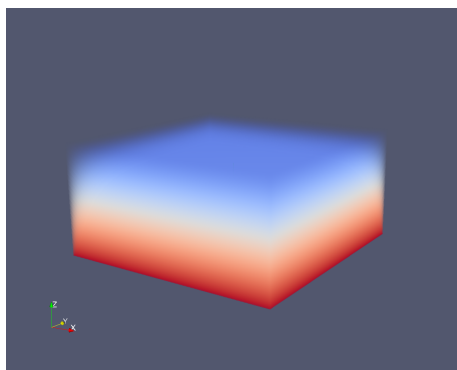


FIGURE 11 – Températures initiales

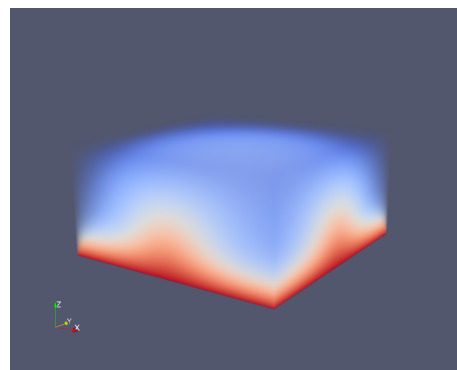


FIGURE 12 – Après 50 frames

Il est aussi possible d'afficher les vitesses en changeant temperature en velocityNorm dans le bon menu déroulant. Mieux, on peut afficher les vecteurs vitesses en cliquant sur l'icône du globe avec des points (Glyph), puis en appuyant sur Apply dans l'onglet Properties.

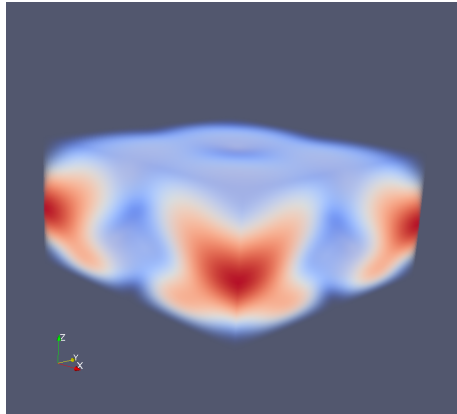


FIGURE 13 – Intensité des vitesses à la frame 50

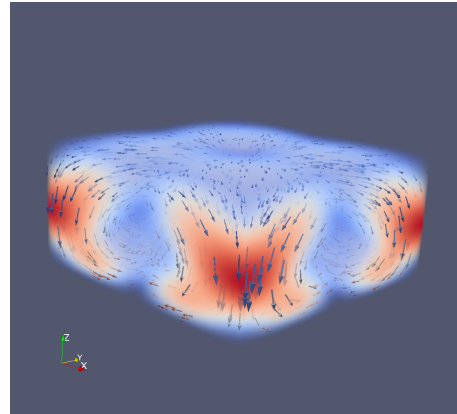


FIGURE 14 – Idem mais avec des vecteurs

Cela nous montre que cet écoulement est tourbillonnaire. Cela est dû à un paramètre du fluide, le nombre de Rayleigh Ra [5] qui caractérise le transfert de chaleur dans un fluide. Il n'a pas de dimension et possède une valeur critique : si ce nombre est inférieure à cette valeur, le transfert se fait par conduction et il n'y a pas ce tourbillonnement, sinon, ces perturbations se produisent et la chaleur se transfère par convection. Pour information, la simulation a été lancée avec $Ra = 3000$.

Si nous mettons les vecteurs vitesses, sur les couleurs de températures, nous pouvons constater que le froid descend, et le chaud monte, car un liquide froid est plus dense qu'un même liquide chaud :

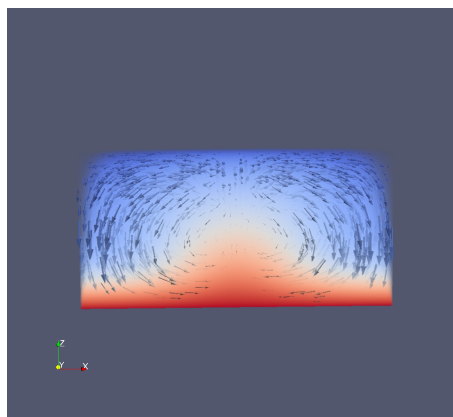


FIGURE 15 – Le froid descend, et le chaud monte

Nous avons alors survolé quelques une des nombreuses fonctions disponibles sur Paraview.

6 Modification d'un programme Palabos, résultat

Dans le domaine de la recherche, il est souvent nécessaire d'adapter des logiciels dans le but de répondre à des besoins spécifiques. Cela a été fait sur une des simulations disponibles, l'anévrisme.

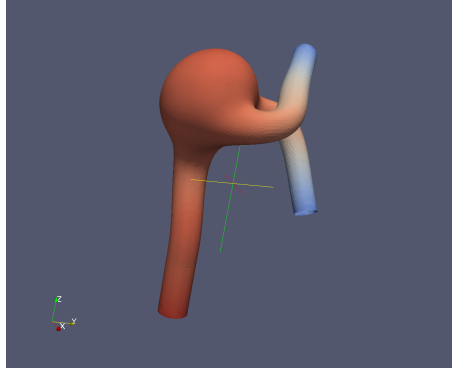


FIGURE 16 – Visualisation sous Paraview. Les couleurs sont fonction de la pression.

Dans ce programme, les valeurs telles que la vitesse sont données à la fin de l'exécution pour quelques points donnés. Nous avons modifié le programme afin qu'il affiche les valeurs pour d'autres points. Cela a été fait à l'aide de l'éditeur Code : :Blocks.

Les points sont en fait les vingt de distances égales entre eux qui sont situés entre le point $\vec{x}_0 = (0.0342; 0.00582; 0.0918)$ et le point $\vec{x}_1 = (0.0369; 0.0205; 0.1000)$. Leurs positions \vec{x} sont fonctions d'un nombre t , variant entre 0 et 1 et découpé en vingt bouts $(0, 1/19, 2/19, \dots, 19/19)$. Elles sont déterminées par l'équation $\vec{x}(t) = \vec{x}_0 + t(\vec{x}_1 - \vec{x}_0)$.

Nous avons ensuite mis en pratique cela en mesurant les normes des vitesses $\|\vec{u}(t)\|$ sur ces vingt points, et ce, en exécutant le programme avec une résolution de 100, puis 200, pour voir s'il y a convergence à 100. Résultat : une résolution de 100 est insuffisante, car les valeurs obtenues avec une résolution de 200 diffèrent de manière conséquente. Vitesses (normes) en fonction de t :

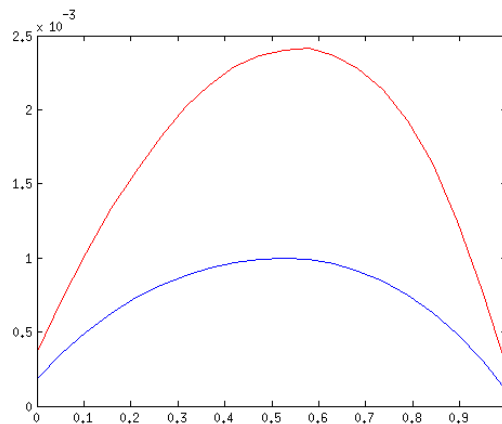


FIGURE 17 – En rouge avec une résolution de 100, en bleu avec une de 200

7 Conclusion

Dans ce rapport, nous avons pu avoir un survol des logiciels Palabos et Paraview, ainsi qu'une introduction à l'exécution parallèle et l'utilisation d'un ordinateur parallèle. Nous avons également vu un exemple d'adaptation d'un code source en fonction des besoins.

Références

- [1] Page web de Palabos. <http://www.palabos.org/>, consulté le 13 mai 2013.
- [2] Page web de Paraview. <http://www.paraview.org/>.
- [3] Sébastien Sauvage. C'est quoi ko, Mo, Go ? <http://sebsauvage.net/comprendre/unites/index.html>, consulté le 13 mai 2013.
- [4] UNIGE. Ressources de calcul. http://spc.unige.ch/doku.php?id=teaching:parallelisme:tutoriels:computing_resources, consulté le 13 mai 2013.
- [5] Wikipédia. Nombre de Rayleigh. http://fr.wikipedia.org/wiki/Nombre_de_Rayleigh, consulté le 13 mai 2013.