Colloque « Penser Petaflops »

Atelier « *Architecture, Algorithmique, Applications : l'intégration* »

Jean ROMAN (INRIA Bordeaux – Sud Ouest / LaBRI UMR 5800),
Francis DAUMAS (CINES - Montpellier)

Introduction

L'utilisation des futurs supercalculateurs pétaflopiques et exaflopiques, constitués d'un grand nombre de CPUs et certainement hybrides, nécessitera une nouvelle approche, qui devra étroitement combiner des compétences applicatives, de programmation, d'algorithmique et la connaissance des architectures sous jacentes.

L'atelier « Architecture, Algorithmique, Applications : l'intégration » du colloque « Penser Petaflops » s'articule ainsi autour de deux axes de réflexion principaux : quelles stratégies et quelles méthodologies doivent être mises en place pour relever le défi de l'utilisation maximale de la puissance de calcul ?

Mode opératoire

Le mode opératoire adopté pour cet atelier a consisté dans un premier temps en l'ouverture d'un wiki (http://penser-petaflops.cines.fr), login générique « petaflop » et mot de passe « atelier3A », sur lequel les participants ont pu déposer leurs contributions.
Une réunion a été programmée et s'est tenue le 26 septembre 2008 au CNRS, ce qui a permis de dégager un certain nombre d'axes de réflexion à l'intérieur des thèmes qui avaient été proposés pour cet atelier.
Les participants à cette réunion (chercheurs, enseignants-chercheurs, constructeurs, et utilisateurs du monde industriel) ont été alors invités à rédiger le contenu de leurs interventions pour alimenter le wiki.

Thèmes abordés

- **Aspects stratégiques**

Les thèmes sur lesquels ont porté la réflexion dans ce domaine et qui ont fait l'objet d'une contribution dans le wiki sont les suivants :

Adaptation et pérennité des codes existants

– Pluridisciplinarité : Les contributions ont mis en avant l'intérêt de démarches pluridisciplinaires même si la complexité de leur mise en œuvre a été soulevée. Celles-ci concernent les collaborations : métiers – Mathématiques/Informatique mais aussi dans le cas des couplages de code : multidisciplinaires « métier ».
– Massivement parallèle : Un second point évoqué concerne le passage des codes sur un grand nombre de cœurs (au-delà de 10 000) versus l'intégration dans des

chaînes complexes de codes moins massivement parallèle (5 000 cœurs). Le calcul pétaflopique conduirait à piloter N simulations par un superviseur.

- Pérennité des langages : Fortran et l'ère pétaflopique semblent compatibles, une bonne nouvelle pour les nombreux codes existants, mais le passage à du MPI non bloquant exige la réécriture de noyaux de codes qui peut s'avérer pénible.
- Tirer partie des accélérateurs : Le côté hybride des machines soulève également des réflexions, le gain important obtenu par l'utilisation des accélérateurs nécessite un investissement non négligeable et la combinaison avec MPI se heurte au rapport temps de calcul – temps de communication qui ne permet plus de recouvrir facilement des communications non bloquantes par du calcul.
- Qualité des algorithmes : Enfin certains algorithmes n'étant pas adaptés au passage à l'échelle, il sera nécessaire de les remplacer par des algorithmes adéquats si on veut la performance. Il serait utile d'organiser des séminaires techniques réguliers sur les algorithmes et leur utilisation optimale.
- Deux paramètres doivent être examinés au regard du passage à l'échelle : le problème lui-même et sa décomposition d'une part et la qualité de son implémentation d'autre part.

Développement de nouveaux codes

- Algorithmes et performances : l'importance des progrès des algorithmes dans la performance est primordiale, mais un algorithme parallèle n'est pas suffisant, il faut également qu'il soit numériquement efficace. Efficacité et parallélisme ne vont pas nécessairement facilement de pair. Un accent doit donc être mis sur la recherche et l'utilisation de « bons » algorithmes.
- Mutualisation de noyaux de codes : la constitution de bibliothèques de noyaux de codes optimisés et multi plateformes devrait être favorisée (exemple les librairies numériques parallèles PETSc, Hypre). Cela permettrait d'assurer une certaine portabilité en préservant la performance.

Gestion des données et des I/O hautes performances

La prise en compte des grands volumes de données liés aux nouvelles puissances doit s'accompagner d'une progression de la partie I/O des architectures et d'une utilisation des I/O parallèles. La gestion des meta-données prendra une importance cruciale.

Pluridisciplinarité

- La mise en place de véritables équipes pluridisciplinaires est une réponse fortement préconisée pour dominer la complexité des applications et des architectures. Il faut alors valoriser cette activité dans les carrières.
- La création de plateaux (peut-être limités dans le temps) regroupant plusieurs disciplines pourrait être expérimentée.
- Par contre il faut éviter que les applicatifs soit les seuls donneurs d'ordre pour préserver une réflexion à plus long terme sur les méthodes mathématiques et informatiques.

– La pluridisciplinarité doit également être respectée à l'intérieur même des mathématiques et de l'informatique (spécialistes des systèmes, de la programmation, …).

- **Aspects méthodologiques et applicatifs**

Les thèmes sur lesquels ont porté la réflexion dans ce domaine et qui ont fait l'objet d'une contribution dans le wiki sont donnés ci-dessous. Les 3 premiers sont naturellement très liés et ont fait l'objet de contributions croisées. Le dernier point a fait l'objet de contributions dans les domaines applicatifs du climat, de la sismologie et plus globalement des sciences de la Terre ; des documents détaillés sur ces aspects applicatifs sont accessibles via le wiki.

Prise en compte de l'hétérogénéité des ressources physiques des machines -- Virtualisation du hardware, modèles d'exécution unifiés, évaluation et portabilité des performances -- Approche unifiée de programmation et instrumentation des codes

Il se dégage de manière claire des contributions que la taille en terme de nombre de cœurs de calcul ainsi que l'hétérogénéité de la structure des machines pétaflopiques seront des obstacles majeurs à l'obtention de performances et que les modes de programmation classiquement utilisés aujourd'hui ne conviendront plus.

Il apparaît donc qu'une virtualisation forte de l'architecture globale est incontournable puisqu'il s'agira de comprendre et surtout d'utiliser des mécanismes (très) différents intervenant à différents niveaux (calcul pipeliné/vectoriel et superscalaire; multi threading et multi cœurs massifs; exploitation de caches multi niveaux et gestion hiérarchique optimisée de la mémoire ; synchronisation d'un grand nombre d'entités concurrentes ; asynchronisme, communications et recouvrement).

L'idée d'avoir une modélisation hiérarchique (plus ou moins générique) de l'architecture correspondant aux divers niveaux de parallélisme exploitables et à partir de laquelle des supports d'exécution pourraient garantir de manière raisonnable une certaine portabilité des performances semble une bonne voie.

Cette virtualisation et un modèle d'exécution quasi-unifié pourraient ainsi conduire à des générations de code spécifiques en fonction du hardware, à une optimisation de code et à une exploitation plus massive du parallélisme à grain fin, et aussi à des politiques génériques d'ordonnancement d'un grand nombre de « threads » tout en favorisant la localité des accès mémoire. À noter qu'une instrumentation des codes bien intégrée au niveau des langages de programmation (avec une approche incrémentale d'optimisation des codes) est alors indispensable pour aider autant que faire ce peut les compilateurs et les supports d'exécution pour tirer de la performance.

Des modèles de programmation hiérarchiques, en cohérence avec la virtualisation de l'architecture et les supports d'exécution sous-jacents, pourraient conduire à une approche unifiée (probablement hybride) de programmation, cela pouvant déboucher même alors sur une conception plus unifiée et hiérarchique des algorithmes et des

applications de grande taille ; il va s'en dire que cela serait une démarche pour les nouveaux codes devant utiliser ces nouvelles machines mais le problème est entier pour les (très) anciens codes que l'on souhaiterait porter à moindre coût de réécriture… Les langages de programmation devront donc permettre d'exprimer du parallélisme à grain moyen et un maximum de parallélisme à grain fin, l'usage systématique d'appels à des noyaux de calcul et à des bibliothèques (parallèles) spécifiques étant incontournable pour avoir des performances. Il s'agira donc de trouver le bon niveau intermédiaire entre une généricité importante qui conduirait à virtualiser au maximum la complexité du hardware de ces nouvelles machines et la capacité laissée à l'utilisateur d'exprimer les propriétés fines de ces algorithmes pour que le compilateur et le support d'exécution puissent mettre en œuvre des optimisations pertinentes.

Passage à l'échelle pour l'utilisation efficace du multicore massif

En relation avec ce qui a été dit ci-dessus, concevoir une algorithmique (de fait hiérarchique) exploitant différents types de parallélisme (sur des cœurs de calcul généraliste et sur des cœurs de calcul spécialisés) pour des problèmes irréguliers et sur machine pétaflopique (donc qui passe à l'échelle avec une scalabilité très importante) est un vrai challenge de recherche sur lequel la communauté scientifique des algorithmiciens du HPC doit se mobiliser.

Il conviendra aussi de travailler sur la conception de bibliothèques numériques encore plus scalables et plus performantes car elles seront les briques de base incontournables pour les applications en vraie grandeur.

Enfin, l'ordonnancement des calculs et des communications sera de fait un problème à reconsidérer compte tenu du très grand nombre de threads à gérer et en prenant en compte de manière centrale les contraintes d'affinité mémoire.

Aspects tolérance aux pannes

Le problème est considéré comme crucial pour une exploitation effective des calculateurs pétaflopiques.

La première voie pour laquelle il commence à y avoir quelques solutions (spécifiques) est la mise en œuvre systématique d'un système de checkpoint/restart mais au prix d'I/O très performantes car les masses de données à sauvegarder/restaurer sont sans commune mesure avec les calculs parallèles traditionnels. La poursuite de travaux concernant un MPI tolérant aux pannes et adapté à ce nouveau contexte a été aussi proposée comme prospective.

Le diagnostic d'une défaillance du hardware (la probabilité de panne sera naturellement très forte sur une machine ayant des centaines de milliers voir des millions de cœurs) sera aussi un élément à intégrer dans cette problématique.

Impact sur les grands challenges applicatifs

Les contributions montrent bien les attendus et l'impact énorme que les calculateurs pétaflopiques (et les suivants..) pourront avoir sur la compréhension de phénomènes physiques complexes multi-physiques et multi-échelles. Nous renvoyons aux textes présents sur le wiki qui sont particulièrement intéressants en ce qui concerne plus particulièrement la climatologie, la sismologie et plus globalement les sciences de la Terre.