

SUJET DE THESE : Parallélisme et équilibrage dynamique de charge en dynamique rapide

Doctorant : Marwa Sridi	Financement : CFR
Responsable CEA : Vincent Faucher	Université d'inscription : Université de Joseph Fourier-Grenoble
Directeur universitaire : Bruno Raffin	Ecole doctorale et DEA : Ecole doctorale Mathématique, Sciences et Technologies de l'Information, Informatique
Laboratoire d'accueil : CEA	Date de début de thèse : 28/01/2013

1 Contexte

La mise en œuvre d'une stratégie de parallélisme permet d'exécuter simultanément un groupe d'instructions indépendantes sur plusieurs processeurs. Cette technique de partage de travail est devenue incontournable dans le cas des traitements complexes afin d'aboutir aux résultats souhaités dans un temps raisonnable.

Pour tirer profit du parallélisme, il est impératif de bien exploiter les nœuds de calcul mis à notre disposition. Pour ce faire, il faut gérer d'une façon optimale l'utilisation de la mémoire et adopter une stratégie efficace pour la distribution des tâches sur les nœuds de calcul.

L'enjeu consiste dans le cadre de ce projet de thèse à aborder le problème du parallélisme dans le code EUROPLEXUS, développé par le CEA, la commission européenne, EDF et l'ONERA pour la simulation en dynamique rapide des transitoires accidentels brutaux.

2 Objectifs

Dans la première phase de ce travail, nous avons mis en place une nouvelle organisation de la structure de données d'EUROPLEXUS. Cette nouvelle structure est construite de telle sorte que les données consultées par le processeur travaillant sur une partie du calcul pendant un pas de temps donné soient le plus contiguës possible afin qu'elles tiennent dans le cache de ce dernier.

Dans cette répartition, les blocs de données sont compacts et indépendants. Ces deux caractéristiques ont préparé le terrain pour une accélération parallèle optimale de la boucle de calcul élémentaire via la librairie de calcul parallèle KAAPI (INRIA/LIG).

Cette librairie implémente un ordonnancement dynamique par vol de travail et offre la possibilité d'effectuer un parallélisme hiérarchique (HWS) qui répond aux spécificités de l'architecture des nœuds de calcul. Nous avons implémenté deux niveaux de parallélisme dans EUROPLEXUS en distribuant dynamiquement les groupes d'éléments sur les macro-threads du premier niveau hiérarchique du nœud de calcul. Les threads de la boucle interne prennent en charge les calculs élémentaires à l'intérieur de chaque groupe.

3 Approches

- Réorganisation de la structure de données par passage d'une structure de type Structure of Array (SOA) à une structure de type Array of structure (AOS).

- Parallélisme à un seul niveau en utilisant la bibliothèque de programmation parallèle OpenMP.
- Parallélisme à un seul niveau en utilisant la bibliothèque de programmation parallèle KAAPI.
- Parallélisme Hiérarchique basé sur le vol de travail pour plusieurs niveaux de boucle.

4 Résultats

Les mesures de performances sur différents jeux de données montrent que notre approche est efficace pour l'utilisation de la mémoire cache. Cette approche nous a permis de réduire le temps d'exécution d'environ 50% pour une taille de groupe de données bien défini selon les caractéristiques matérielles de la plateforme de calcul (taille de la mémoire cache L3) et selon le cas test simulé.

Les performances du parallélisme de la boucle externe du code EUROPLEXUS en utilisant la bibliothèque OpenMP sont comparables à celles obtenues par la bibliothèque expérimentale KAAPI. Dans ces deux cas, grâce à l'approche de réorganisation de la structure de données en groupes contigus, nous avons réussi à accélérer la boucle élémentaire en réduisant le temps écoulé dans cette partie du code d'environ 70%.

L'implémentation d'une approche de parallélisme hiérarchique basée sur un équilibrage dynamique de charge entre les différents niveaux est devenue faisable grâce à l'ordonnanceur offert par KAAPI. Ce nouveau type d'ordonnement que nous proposons, assure l'équilibrage de charge entre les différents niveaux imbriqués tout en respectant la hiérarchie de la plate-forme parallèle. Les gains de performances obtenus par plusieurs cas test, prouvent que, pour obtenir un meilleur équilibrage de charge avec cet ordonnanceur, nous devrions opter pour un ordonnancement « contraint » pour les différents niveaux imbriqués.

5 Conclusions

Nous avons étudié l'organisation de la structure de données du code de simulation en dynamique rapide EUROPLEXUS. Afin d'optimiser l'utilisation du cache pour ce code, nous avons proposé une nouvelle approche d'organisation de sa structure de données. Notre méthode est basée sur la réorganisation des données des tableaux élémentaires sous forme de groupes. Le but de cette approche est d'augmenter la localité des données dans le cache. Nous avons vérifié expérimentalement l'apport de cette approche, dans le cas des problèmes de grande taille, pour diminuer le taux de défauts de cache dans la boucle des calculs élémentaires.

Nous avons commencé par implémenter un premier niveau de parallélisme qui tire profil de l'approche de réorganisation par groupe ainsi proposée. Ensuite, nous avons proposé un parallélisme à deux niveaux imbriqués basé sur un équilibrage dynamique de charge entre les threads d'un même niveau et aussi entre les threads des différents niveaux hiérarchiques.