

Portage de YML/XMP dans Docker et expérimentations ;

Benchmarking avec des codes du calcul à grande échelle d'algèbre linéaire

L'exploitation optimale des architectures many-cœurs avec des structurations de mémoire complexe est une tâche difficile. Celle-ci passe nécessairement le plus souvent à l'utilisation d'un environnement d'aide à la programmation et à l'exécution de ce type d'architectures multi-niveaux avec beaucoup de composants hétérogènes. La portabilité de ce type d'environnement sur différentes architectures ou solutions techniques est un véritable challenge pour son utilisateur.

YML est un environnement de programmation à très grande échelle incluant un langage d'expression de graphe de composantes/tâches d'une application. Chacune des composantes pouvant s'écrire avec un langage de programmation parallèle (comme XMP, HPF, etc.) ou séquentiel (comme C, C++, Fortran, etc.). Le langage XMP qui est un langage parallèle permettant la programmation selon le modèle PGAS (permettant la combinaison des deux modèles mémoire partagée et mémoire distribuée) est intégré dans une version récente de YML appelée YML-XMP. Beaucoup de travaux de recherche et développement dans le cadre de plusieurs projets internationaux ont été effectués. Ceux-ci incluent la programmation des applications numériques avec YML-XMP sur des systèmes distribués (comme la plateforme GRID'5000) ou sur des superordinateurs (comme ceux K computer, Oakforest-PACS au Japon, supercalculateur hybride au centre ROMEO à l'URCA France, Hooper à Berkeley aux USA) de la liste TOP500.

YML répond à ces exigences des outils d'aide à la programmation et à l'exécution des architectures citées. Il repose sur un paradigme de graphe de dépendances qui permet de distribuer les tâches de calcul sur les nœuds d'un système distribué tout en permettant l'exploitation du parallélisme de tâche au sein de chaque nœud.

Pour atteindre ces buts, YML-XMP réalise une analyse de dépendance (global et local) et génère un code source qui sera compilé de manière adaptée à chaque nœud. Le code de chacun des nœuds peut être en un langage parallèle comme XMP (langage à pragma avec C et Fortran) en C, C++ ou tout autre langage.

Les codes de la solution YML-XMP ont été écrits avec des versions des compilateurs GCC5.4 et antérieures associés à MPI3.1. Ces compilateurs évoluent et mettent en place des règles de compilation qui ne sont pas nécessairement compatibles avec l'outil YML-XMP tel qu'il a été pensé il y a plusieurs années.

Docker est une solution de para-virtualisation Linux qui permet la distribution d'images « prêtes à l'emploi » et qui est devenu un standard dans le domaine d'informatique.

L'objectif du stage est de produire une version de YML-XMP compatible avec la distribution Debian Bullseye (Debian 11). Cela, par l'adaptation des scripts et des paramètres de compilation. Ce travail pourra être réalisé dans une image Docker. Une attention particulière sera portée au niveau de privilège pour l'exécution de l'image (non-root) qui est un paramètre important pour les systèmes distribués à grande échelle.

Le processus de développement se fera selon le déroulé suivant :

- 1) Identification des verrous à la compilation sur Debian Bullseye
- 2) Implémentation des solutions et correctifs.
- 3) Livraison d'une maquette Docker multi nœuds (dockerisés) – benchmarking de l'implémentation en tenant compte des mécanismes de communication des nœuds effectuée par le middleware de YML.
- 4) Proposition d'un ensemble de programmes tests, basés sur des algorithmes d'algèbre linéaire, pour la mise à jour de YML-XMP ainsi réalisé,
- 5) Proposition de métriques de performances permettant la mise en évidence de l'efficacité de la solution proposée.

Responsables du stage :

Nahid EMAD, (Nahid.Emad@uvsq.fr)
Jean-Michel Batto (jean-michel.batto@uvsq.fr)

Lieu de stage :

Maison de la Simulation
Digitéo Labs – Bat. 565, CEA/Saclay, 91191 Gif-sur-Yvette Cedex