CommMaMa :
Asynchronisation transparente de communications MPI

Contexte

La simulation numérique permet de modéliser des phénomènes physiques qu'il n'est pas envisageable d'expérimenter réellement car prédictifs (météorologie, sismologie), onéreux (conception d'un avion), irréel (effets spéciaux) ou encore trop dangereux (essai nucléaire). Parce qu'exigeant le traitement de beaucoup de données pour atteindre une précision suffisante en des temps de calcul contraints, les modèles de simulation numérique sont exécutés sur des machines de calcul considérable qui sont en réalité des agglomérats de machines reliées par des réseaux dédiés. Afin d'échanger les données de travail dans cet environnement d'exécution distribué, l'interface de communication MPI (pour *Message Passing Interface*) est historiquement utilisée. Il en existe différentes implémentations proposées par des laboratoires de recherche comme des industriels, chacune avec leurs spécificités. Cependant, le grand défi commun de tels supports d'exécution est de faire en sorte que les ressources de calculs soient au maximum utilisées à réaliser du calcul, et non des opérations (même si indispensables) comme des communications ou pire encore, à ne rien faire car en attente de réception de données, par exemple.

MPI propose des primitives de communication non bloquantes permettant de recouvrir une communication par du calcul. Si ce mecanisme permet de cacher le coût d'une communication, il nécessite que l'implementation MPI fasse progresser les communications pendant le calcul. Pour cela, la plupart des implementations ont besoin que l'application fasse explicitement appel à des primitives de test de terminaison de communication (MPI_Test) régulièrement. Au delà du manque d'efficacité de ce procédé, le code se voit grandement complexifié et le développeur solicité sur des aspects qu'il ne devrait pas avoir à gérer.

Objectifs du stage

Dans ce stage, nous proposons de nous intéresser à la prise en charge transparente et efficace de la progression des communications en parallèle du calcul. Le stage cible le développement de CommMaMa , un support d'exécution s'intercalant entre l'application *legacy* de HPC distribuée et une instance de MPI.

Dans un premier temps, il s'agira de mettre en place le moteur de progression de communication et d'intercepter les communications non bloquantes et les primitives de test de terminaison de communication. Il faudra mettre en place une méthodologie d'évaluation des performances obtenues sur des simulations de petite et moyenne taille afin d'être en mesure de se comparer à des implantations MPI assurant un support de progression.

Dans un second temps, il s'agira de prendre en compte les communications bloquantes. Après les avoir identifiées, vous mettrez en oeuvre des stratégies consistant à sauvegarder les messages à émettre en vue de stipuler à l'application la terminaison de sa communication à son dépot. L'envoi pourra ainsi être mené totalement en parallèle de l'application par CommMaMa . Différentes stratégies pourront être envisagées comme de la copie simple et brute, l'interception d'accès aux pages mémoire, l'utilisation de mémoire rapide intermédiaire,maintien de journalisation de modification, le checkpointing de la mémoire de l'application, etc. Aucune n'étant universelle, une évaluation fine devra être faite afin d'identifier les seuils et paramètres à prendre en compte par CommMaMa pour en sélectionner une à l'exécution.

Enfin, il s'agira d'utiliser et d'évaluer CommMaMa pour une application réelle de plus grande ampleur.

Mots clés : Supercomputing, MPI, recouvrement de communications, support d'exécution, mémoire, système

Contact : Elisabeth Brunet - elisabeth.brunet@telecom-sudparis.eu