RGE

Réseaux Grand Est

Réunion du 14 février 2013 à Belfort

Résumés des présentations

Filtres médian sur GPU

Gilles Perrot, FEMTO-ST Belfort

Accélération d'une simulation de sol sur GPU

Guillaume Laville, FEMTO-ST Besançon

Victime de leur succès, les modèles et les simulateurs agents sont utilisés pour décrire des systèmes complexes de plus en plus massifs. Dans un même temps, les exigences requises dans les projets de recherche (en biologies, en géographie, ou en science du sol) sont de plus en plus importantes. Ainsi, le domaine de la simulation orientée agent connait aujourd'hui un tournant : l'évolution des environnements d'exécution tend vers des architectures parallèles (CPU multi-cœurs, GPU, cluster, grille de calcul) auxquelles les techniques agents actuelles doivent s'adapter. Cependant, tirer véritablement profit de ces nouveaux moyens n'est pas trivial et nécessite l'usage d'approches issues du monde du calcul haute performance, notamment une bonne connaissance des modèles de programmation en lien avec l'infrastructure matérielle. Sur la base du modèle MIOR (modèle agent pour la décomposition de la matière organique dans les sols), nous présentons les adaptations réalisées pour son exécution sur GPU. De cette expérience, nous suggérons des procédés pour tirer parti de ressources CPU et GPU. Sous le couvert d'un exemple concret, le modèle SWORM (modèle de bioturbation des sols), nous illustrons nos propos en élaborant une architecture hybride de simulateur (CPU-GPU).

Recherche et retour d'expérience sur le Xeon-phi

Damien Dubuc, Expert HPC, société ANEO (exposé invité)

Solveurs parallèles non linéaires creux des problèmes de l'obstacle sur des grappes GPU

Lilia Ziane Khodja, FEMTO-ST Belfort

Dans cette présentation, nous nous intéressons aux solutions des problèmes de l'obstacle de très grandes tailles définis dans un domaine tridimensionnel. Les problèmes de l'obstacle interviennent dans de nombreuses applications scientifiques, par exemple: la m´mécanique des fluides, la bio-mathématique ou les mathématiques des finances. Leurs résolutions consistent en la résolution de systèmes d'équations non linéaires issus de leurs discrétisations spatiales. Pour cela, nous avons utilisé des algorithmes itératifs parallèles pour des grappes de GPUs. Nous avons mis en œuvre les algorithmes parallèles, synchrone et asynchrone, de la méthode itérative Richardson projetée. Nous avons utilisé une programmation parallèle hétérogène MPI et CUDA. Les expérimentations sur une grappe de 12 nœuds GPU montrent que l'utilisation de GPUs permet de réduire le rapport calcul/communication. Dans ce contexte, l'algorithme asynchrone supporte mieux le passage à l'échelle que l'algorithme synchrone.

Exposé sur les technologies FPGA

Nicolas Colombain, Société Armadeus (exposé invité)

Générateur de coprocesseur pour le traitement de données en flux (vidéo ou similaire) sur FPGA

Gwenhaël Goavec-Merou, FEMTO-ST Besancon

Le FPGA devient de plus en plus incontournable lorsqu'il est nécessaire de garantir des traitements en temps-réel avec des débits très importants. Toutefois ce composant, de par sa structure matérielle mais également du fait des langages utilisés pour décrire les applications, présente une difficulté non négligeable pour l'utilisateur n'ayant pas une bonne connaissance de ce type de matériel. Notre objectif a été de créer un outil permettant à l'utilisateur de constituer des chaînes de traitements par assemblage d'algorithmes disponibles dans une bibliothèque dédiée tout en lui permettant de s'affranchir de la plupart des vérifications nécessaires à la garantie de la bonne génération du binaire (vérification de la compatibilité des blocs et garantie de respect des ressources disponibles) et au bon fonctionnement de l'ensemble (validation des capacités temporelles de chaque bloc à recevoir et traiter le flux).

Nous nous intéresserons au cas particulier du traitement d'images en temps réel, application particulièrement gourmande en bande passante et ressources de traitement. La mise en œuvre des divers algorithmes classiques de traitement imposent de vastes gammes de contraintes, avec traitement de chaque pixel indépendamment de ses voisins, ligne par ligne, par fenêtre glissante ou n´nécessitant le stockage de l'image complète (cas de la rotation par exemple). Chacun de ces cas doit être analysé en terme de temps de traitement, de vitesse de flux maximum, et en terme de latence de remplissage du pipeline de traitement. Trois cas se présentent : la source de données, la séquence de traitement, et le puits de données. L'outil d'assemblage de blocs vise à sélectionner, parmi un ensemble d'implémentations d'un même algorithme, la solution la plus en adéquation avec les ressources matérielles disponibles ainsi que le flux entrant, et ce en insérant un nombre minimum de latences afin de maximiser la bande passante de traitement.

Après une présentation globale d'une part des difficultés inhérentes, pour un utilisateur n'ayant pas ou peu de compétences en développement dans ce type d'environnement, d'autre part des possibilités d'un FPGA en terme de solutions d'implémentation, et pour finir des caractéristiques de quelques algorithmes ayant guidés certains de nos choix, nous présenterons le fonctionnement de l'outil avant de rentrer dans les détails de sa structuration et de la description des algorithmes en terme de typage, de ressources mais également de comportement temporel. Ensuite nous verrons les analyses réalisées afin de déterminer les combinaisons d'implémentations compatibles avec le matériel mais également avec le flux propagé dans la chaîne. Cette présentation se finira par la description d'une application réalisée afin de valider l'ensemble des choix autant concernant la description que les analyses.