Übungen zur Vorlesung

Paralleles Höchstleistungsrechnen

Dr. S. Lang, D. Popović

Abgabe: 26. Januar 2010 in der Übung

Übung 29 Gravitations-N-Körper-Problem mit Tiling und OpenMP (5 Punkte)

Schreiben Sie eine Version des N-Körper-Problems, dass ein Tiling mit der Verwendung von *OpenMP* kombiniert. Als Gerüst wird die Datei `nbody_tiled_openmp.c` bereitgestellt, die gleichzeitig eine Lösung der Kachelungs-Aufgabe ist. Es muss also nur noch die Parallelisierung mit *OpenMP* eingebaut werden. Führen Sie nun wiederum Simulationen mit den gleichen Parametern wie auf den beiden vorherigen Übungsblättern aus und messen Sie den Speed-Up und MFLOP-Rate. Falls möglich, vergleichen Sie die Ergebnisse mit jenen der vorherigen Übungszettel.

Übung 30 N-Körper-Problem mit Cuda (5 Punkte)

Die Datei `nbody_cuda.cu` enthält eine Version des Gravitation-N-Körper-Problems, das auf der Grafikkarte gerechnet wird. Im Pool sind Nvidia GPUs eingebaut. Der Code läuft teilweise auf der GPU, teils auf der CPU. Machen Sie sich mit dem Code vertraut und führen Sie mehrere Simulationen mit unterschiedlichen Parametern durch.

Die Datei `nbody_cuda.cu` enthält zwei Varianten der Funktion `acceleration()`: `acceleration_gpu()` benutzt die GPU, `acceleration_cpu()` hingegen die CPU. Mit `make` werden zwei Binaries erstellt, `nbody_cuda` und `nbody_cuda_gpu`, die die entsprechenden Funktionen zur Berechnung der Beschleunigungen verwenden. Die Ergebnisse der Simulationen werden in `.vtk`-Dateien abgelegt.

1. Warum wurde der Parameter $\epsilon 2$ im Plummer-Potential eingeführt? Warum kann nicht der ursprüngliche Wert von $1e-14$ verwendet werden (siehe die Vanilla-Variante), falls auf der GPU gerechnet wird?
2. Vergleichen Sie die Ergebnisse von mehreren Simulationen mit identischen Parametern auf je der GPU und CPU. Wie unterscheiden sich die Ergebnisse hinsichtlich der Genauigkeit?

Sie müssen unter Umständen den Pfad zum Cuda-Compiler explizit setzen, er liegt unter `/usr/local/cuda/bin/nvcc`, und den Pfad zur Bibliothek `libcudart.so.2` setzen: `export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/usr/local/cuda/lib/`.