

VSDB

Protokoll

GPGPU

Dominik Backhausen Alexander Rieppel

17. Januar 2014

5AHITT

Inhaltsverzeichnis

1	Aufgabenstellung	3
2	Recherche	4
2.1	Nutzung von GPUs und Vorteil gegenüber CPU	4
2.2	Entwicklungsumgebungen	4
2.3	Transcompiler	4
3	Arbeitsaufteilung	5
4	Arbeitsdurchführung	6
5	Testbericht	7
6	Quellen	8

1 Aufgabenstellung

GPU Computing oder GPGPU(= General Purpose Computing on GPUs) bezeichnet die Verwendung eines Grafikprozessors (engl. Graphics Processing Unit oder GPU) für allgemeine Berechnungen im wissenschaftlich-technischen Bereich. Übersetzt bedeutet GPGPU in etwa Allgemeine Berechnung auf Grafikprozessoren.

Informieren Sie sich über die Möglichkeiten der Nutzung von GPUs in normalen Anwendungen. Zeigen Sie dazu im Gegensatz den Vorteil der GPUs in rechenintensiven Implementierungen auf [1Pkt]. Gibt es Entwicklungsumgebungen und in welchen Programmiersprachen kann man diese nutzen [1Pkt]? Können bestehende Programme (C und Java) auf GPUs genutzt werden und was sind dabei die Grundvoraussetzungen dafür [1Pkt]? Gibt es transcompiler und wie kommen diese zum Einsatz [1Pkt]?

Präsentieren Sie an einem praktischen Beispiel den Nutzen dieser Technologie. Wählen Sie zwei rechenintensive Algorithmen (z.B. Faktorisierung) und zeigen Sie in einem Benchmark welche Vorteile der Einsatz der vorhandenen GPU Hardware bringt [12Pkt]! Um auch einen Vergleich auf verschiedenen Plattformen zu gewährleisten, bietet sich die Verwendung von OpenCL an.

Diese Aufgabe ist als Gruppenarbeit (2) zu lösen. Es ist zu beachten, dass diese Aufgabe mit der Aufgabe05 gekoppelt ist, d.h. nur eine der beiden Aufgaben wird verpflichtend bewertet! Zusätzliche Abgaben erhöhen die Gesamtpunkte und können somit zur Notenverbesserung dienen.

2 Recherche

2.1 Nutzung von GPUs und Vorteil gegenüber CPU

GPUs können vor allem sehr gut mit parallelisierbaren Algorithmen Arbeiten, da sie weit mehr Kerne besitzen, als eine CPU. Die GPU ist daher besonders für wiederholende und gleichzeitig rechenintensive Aufgaben geeignet und bei diesen auch deutlich schneller. Die GPU zieht erst bei oft wechselnden und ausschließlich sequentiell abzuarbeitenden Aufgaben den Kürzeren.

2.2 Entwicklungsumgebungen

Vor kurzem wurde von AMD die Ati Stream SDK 2.0 veröffentlicht, diese ist nicht nur auf GPUs sondern auch auf CPUs lauffähig (auch von Intel). Neben dem Arbeiten in C, ist es allerdings auch möglich mit Eclipse, in Java zu arbeiten. Man benötigt dazu lediglich, die Bibliothek javaCL.

2.3 Transcompiler

Ja es gibt transcompiler die beispielsweise von OpenCL auf CUDA kompilieren können.

3 Arbeitsaufteilung

Name	Arbeitssegment	Time Estimated	Time Spent
Alexander Rieppel	Dokumentation	1h	1h
Alexander Rieppel	Recherche	1h	1h
Alexander Rieppel	Algorithmensuche	1h	1h
Alexander Rieppel	Kernel	2h	2h
Dominik Backhausen	Kernel	2h	3h
Dominik Backhausen	Benchmarkprogramm	2h	3h
Gesamt		7h	11h

4 Arbeitsdurchführung

5 Testbericht

6 Quellen

[http : //ht4u.net/news/21412_erste_finale_opengl-entwicklungsumgebung_von_android_erfuegbar/](http://ht4u.net/news/21412_erste_finale_opengl-entwicklungsumgebung_von_android_erfuegbar/)

<https://code.google.com/p/javac/>