
Protokoll

DezSys08 - GPGPU

Systemtechnik-DezSys
5BHIT 2015/16

Erik Brändli/Michael Weinberger

Note:
Betreuer: Borko/Micheler

Version 0.1
Begonnen am 15. Januar 2016
Beendet am ???

Inhaltsverzeichnis

1	Einführung	1
1.1	Ziele	1
1.2	Aufgabenstellung	1
1.3	Links	1
2	Möglichkeiten GPU Desktop-Anwendungen, Vorteil GPU bei rechenintensiven Implementierungen	2
3	Entwicklungsumgebungen, Programmiersprachen	3
4	Bestehende Programme (C/C++ und Java) auf GPUs nutzen und Grundvoraussetzungen	4
5	Transcompiler und deren Einsatz	5
6	Praktisches Beispiel: Benchmark	6
6.1	Auswahl und Argumentation der Algorithmen	6
6.2	Gegenüberstellung CPU/GPU	6
6.3	Anzahl der Durchläufe	6
6.4	Informationen bei Benchmark	6
6.5	Beschreibung, Bereitstellung des Beispiels	6

1 Einführung

1.1 Ziele

Die Aufgabe beinhaltet eine Recherche über grundsätzliche Einsatzmöglichkeiten für GPGPU. Dabei soll die Sinnhaftigkeit der Technologie unterstrichen werden. Die Fragestellungen sollen entsprechend mit Argumenten untermauert werden. Im zweiten Teil der Arbeit soll der praktische Einsatz von OpenCL trainiert werden. Diese können anhand von bestehenden Codeexamples durchgeführt werden. Dabei wird auf eine sprechende Gegenüberstellung (Benchmark) Wert gelegt. Die Aufgabenstellung soll in einer Zweiergruppe bearbeitet werden.

1.2 Aufgabenstellung

Informieren Sie sich über die Möglichkeiten der Nutzung von GPUs in normalen Desktop-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/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 aussagekräftigen Benchmark welche Vorteile der Einsatz der vorhandenen GPU Hardware gegenüber dem Ausführen auf einer CPU bringt (OpenCL).

Punkteschlüssel:

Auswahl und Argumentation der zwei rechenintensiven Algorithmen (Speicher, Zugriff, Rechenoperationen) [0..4Pkt]

Sinnvolle Gegenüberstellung von CPU und GPU im Benchmark [0..2Pkt]

Anzahl der Durchläufe [0..2Pkt]

Informationen bei Benchmark [0..2Pkt]

Beschreibung und Bereitstellung des Beispiels (Ausführbarkeit) [0..2Pkt]

1.3 Links

OpenCL-Examples von René Hollander & Paul Kalauner [1]

2 Möglichkeiten GPU Desktop-Anwendungen, Vorteil GPU bei rechenintensiven Implementierungen

asdf

3 Entwicklungsumgebungen, Programmiersprachen

asda

4 Bestehende Programme (C/C++ und Java) auf GPUs nutzen und Grundvoraussetzungen

adf

5 Transcompiler und deren Einsatz

asdf

6 Praktisches Beispiel: Benchmark

asdf

6.1 Auswahl und Argumentation der Algorithmen

asdf

6.2 Gegenüberstellung CPU/GPU

sadf

6.3 Anzahl der Durchläufe

sadf

6.4 Informationen bei Benchmark

asdf

6.5 Beschreibung, Bereitstellung des Beispiels

sadf

Literatur

- [1] Kalauner Hollander. Opencl example. <https://github.com/ReneHollander/opencl-example.git>. zuletzt besucht: 15.01.2016.

Listings

Abbildungsverzeichnis