

123 4 5 6 7 8 9 10

Wissenschaftliches Programmieren mit CUDA

Daniel Klimeck und Christian Renneke

Abstract— Das Projekt "Wissenschaftliches Programmieren mit CUDA" beschäftigt sich mit der Berechnung von Simulationen aus dem Bereich der theoretischen Elektrotechnik. Das Verarbeiten der Algorithmen sollte mit Hilfe der Rechenleistung einer Grafikkarte beschleunigt werden, um so die Simulationszeiten zu reduzieren. Die Implementierung erfolgte mittels CUDA (Compute Unified Device Architecture) von dem Unternehmen NVIDIA. Dabei handelt es sich um ein SDK (Software Development Kit) was die Programmierung der Grafikprozessoren von NVIDIA ermöglicht.

I. EINFÜHRUNG

Da bei der Verbesserung von Prozessoren die Erhöhung der Taktfrequenz an ihre wirtschaftlichen Grenzen stößt, setzen die führenden Unternehmen auf parallele Verarbeitung. So werden aktuell vier Kerne in einer CPU genutzt, um mehr Rechenleistung zu erhalten.

Dieser Trend ist bei Grafikkarten bereits deutlich länger zu beobachten. Deren Prozessoren sind darauf ausgelegt immer ähnliche Operationen auf große Datenmengen anzuwenden. Dies lässt sich sehr gut parallelisieren, so dass in Grafikkarten meist mehrere Hundert parallele Prozessoren arbeiten und dies resultiert in einer deutlich höheren Rechenleistung. Um diese hohe Rechenleistung nun auch für andere Zwecke als zur Bildverarbeitung nutzen zu können, hat NVIDIA ein SDK herausgebracht, das es ermöglicht C- oder Fortran-Code auf der GPU auszuführen. Diese Technologie ist für alle von Interesse, die große Datenmengen auf ähnliche Weise verarbeiten und berechnen müssen, wie es zum Beispiel bei vielen Simulationen der Fall ist.

Daher sollte in diese Projektarbeit erarbeitet werden, wie gut sich die Rechenleistung der GPUs nutzen lässt, um typische Simulationsaufgaben der theoretischen Elektrotechnik schneller auszuführen. Dabei sollte ebenfalls untersucht werden, ob die Nutzung der Grafikprozessoren Genauigkeitsverluste mit sich bringt und die Programmierung von CUDA erlernt werden.

GPUs und parallele Berechnungen - Vorteile - Nachteile -
CUDA als Programmierungsumgebung - erwartete Ergebnisse

II. CUDA

Die Compute Unified Device Architecture des Grafikkartenherstellers NVIDIA ist eine Schnittstelle, die es ermöglicht die Rechenleistung der Grafikprozessoren für allgemeine Zwecke zu gebrauchen. So wird von dieser API eine Programmierungsumgebung zur Verfügung gestellt, mit der man C-Code auf Grafikprozessoren ausführen kann. Diese haben Programmierung unter C

Aufruf in Matlab über MEX als Schnittstelle
Aufbau der Grafikkarten

III. SIMULATIONSBEISPIEL

A. Leapfrog-Algorithmus

B. Umsetzung in CUDA

- Beispielcode

IV. ERGEBNISSE DANIEL

Vorstellung der Beispiele, Bragg Reflex Diagramm

V. ZUSAMMENFASSUNG DANIEL

-Lohnt es sich für große Beispiele -Kostenreduzierend

REFERENCES

- [1] J. Driesen, R. Belmans, K. Hameyer, "Adaptive relaxation algorithms for thermo-electromagnetic FEM problems", IEEE Transactions on Magnetics, Vol. 35, No. 3, 1999, pp. 1622-1625.
- [2] L. Jänicke, A. Kost, "Convergence properties of the Newton-Raphson Method for nonlinear problems", in Proc. XIth IEEE COMPUMAG, Rio de Janeiro, Brazil, PF3-6, 1997, pp. 585-586.

Eingereicht am 05.02.2009

Daniel Klimeck (Matr. 6345768),
Christian Renneke (Matr. 6257603)

Universität Paderborn

FG Theoretische Elektrotechnik

Warburger Str. 100

33098 Paderborn

Germany

Email: Klimeck@tet.upb.de / Renneke@tet.upb.de