Analyse der Einsatzmöglichkeiten von Graphikprozessoren in der Optimierung und in Simulationen

Bachelorarbeit

zur Erlangung des akademischen Grades „Bachelor of Science (B. Sc.)“ im Studiengang Wirtschaftswissenschaft der Wirtschaftswissenschaftlichen Fakultät der Leibniz Universität Hannover

vorgelegt von:

Name: Heller Vorname: Tim

[redacted] [redacted] [redacted] [redacted]

Prüfer: Prof. Dr. M. H. Breitner

Hannover, den 07.02.2016

Inhaltsverzeichnis

1. Einleitung

1.1 Motivation

Ursprünglich für die Berechnung der Bildausgabe entwickelt, aber in den letzten Jahren aufgrund ihrer immensen Rechenkraft auch zunehmend für allgemeinere Anwendungsfälle genutzt: die graphics processing unit (GPU). Mittlerweile beschleunigen Grafikprozessoren Anwendungen in vielerlei Plattformen, angefangen bei Autos über Mobiltelefone und Tablets bis hin zu Drohnen und Robotern.[1,2]

Die Motivation für die Nutzung der GPU für wissenschaftliche Anwendungen ist, dass viele rechenintensive Operationen parallelisiert oder zumindest teilweise parallelisiert werden können. Die berechnungsintensiven Teile der Anwendung werden auf die GPU ausgelagert, während der übrige Code auf dem Hauptprozessor, der central processing unit (CPU) läuft. Die GPU repräsentiert mit ihren hochspezialisierten, massiv-parallelen Architekturen einen perfekten Kandidaten für die Ausführung. Hierdurch wurde das sogenannte general-purpose computing on graphics processing unit (GPGPU) eingeführt, also der Einsatz der Grafikkarte als universelle Rechenschnittstelle.[3,4]

Der Trend zu GPU-Computing ist angeregt durch eine große Anzahl an Multiprozessoren und der hohen Speicherbandbreite in heutigen GPUs. Die Entwicklung der Grafikkarte ist insbesondere angetrieben von der unersättlichen Nachfrage nach einer hochauflösenden Grafikdarstellung in Echtzeit der Videospieleindustrie. Eine herkömmliche CPU besitzt nicht mehr als eine Hand voll Kerne, während eine typische GPU mehrere hundert (kleinerer) Kerne besitzt.[5,6]

Die traditionellen Einzelkern-Mikroprozessoren haben Schwierigkeiten, höhere Taktfrequenzen zu erzielen. Eine Lösung für eine Steigerung der Performance sind die Multikern-Architekturen, die mehrere Kerne auf einen Chipsatz integrieren. Beispiele hierfür sind die Produkte Duo Core und Quad Core von der Firma Intel.[7]

Die Kerne einer Grafikkarte sind ideal geeignet, um Aufgaben auszuführen, die eine hohe arithmetische Intensität besitzen. Berechnungen, bei denen das Verhältnis der arithmetischen Operationen zu den Speicheroperationen hoch ist, zählen hierzu.[8] Die geringe Taktfrequenz und der Zugriff auf weniger Speicher sind kennzeichnend für den

[1] Vgl. unn GmbH 2010.
[2] Vgl. Nvidia GmbH 2015a.
[3] Vgl. Nvidia GmbH 2015a.
[4] Vgl. Wu et al. 2010, S. 1.
[5] Vgl. Reese und Zaranek 2012, S. 1.
[6] Vgl. Nvidia Coproration 2015, S. 1.
[7] Vgl. Wu et al. 2010, S. 1.
[8] Vgl. Lee et al. 2010a, S. 7.

Kern einer GPU. Kompensiert wird dieses durch die enorm hohe Anzahl an verfügbaren Kernen. Die GPU kann so programmiert werden, dass dieselben numerischen Vorgänge simultan ausgeführt werden.[9] Das parallele Rechnen hat insbesondere durch die relativ geringen Kosten einer GPU viel Aufmerksamkeit erlangt.

1.2 Zielsetzung

In dieser Arbeit wird der Frage nachgegangen, inwiefern sich Optimierungsprobleme effizienter durch den Einsatz von Grafikprozessoren berechnen lassen. Insbesondere die Firma Nvidia ist auf das grafikprozessorbeschleunigte Rechnen spezialisiert. Über die Nvidia Homepage ist eine Vielzahl an technischen und wissenschaftlichen Forschungsanwendungen aufgelistet, die für das GPU-Computing geeignet sind. Hier wird mit signifikanten Beschleunigungen geworben, für Matlab ist beispielsweise von 2 – 20-fachen Speedups die Rede.[10] Zu beachten ist jedoch, dass die erzielten Beschleunigungen stark von der verwendeten Systemkonfiguration abhängen. Eine Vielzahl der Tests führt Nvidia hausintern durch. Von dem Einsatz einer Spitzentechnik kann ausgegangen werden. Aber auch in der Literatur wird von sogar 10 – 1000-fachen Beschleunigungen berichtet.[11] In dieser Arbeit wird insbesondere untersucht, ob die Beschleunigung eines Programmes durch den Einsatz des Grafikprozessors in den angegebenen Größenordnungen tatsächlich als realistisch einzustufen ist. Der Fokus liegt nicht auf Forschungsergebnissen, die sich auf das high performance computing beziehen. Schlichtweg, weil es der Mehrzahl an Wissenschaftlern nicht zur Verfügung steht. Zuletzt wird der Frage nachgegangen, ob der Einsatz einer GPU in puncto Energieeffizienz vorteilhaft ist. Mit diesen Zielen gehen folgende Forschungsfragen einher:

- Welche Optimierungsprobleme lassen sich mit dem Grafikprozessor schneller lösen?
- Wie hoch sind die Beschleunigungen für Optimierungsprobleme durch den Einsatz des Grafikprozessors tatsächlich?
- Wie energieeffizient ist das grafikprozessorbeschleunigte Rechnen?

1.3 Literaturüberblick und Forschungsmethodik

Nachfolgend ist die in dieser Arbeit vorgestellte Literatur aufgelistet. Bei der Auswahl wurde darauf geachtet, auf Disziplinen zuzugreifen, die in einem Bezug zur Wirtschaftswissenschaft stehen.

[9] Vgl. Suchard et al. 2010, S. 420.
[10] Vgl. Nvidia GmbH 2015b.
[11] Vgl. Lee et al. 2010b, S. 451.

Kapitel	Autoren	Titel	Jahr der Veröffentlichung
4.1	Garcia, Debreuve, Barlaud	Fast k Nearest Neighbor Search using GPU	2008
4.1	Cayton	A Nearest Neighbor Data Structure for Graphics Hardware	2010
4.2	Lee, Yau, Giles, Doucet, Holmes	On the Utility of Graphics Cards to Perform Massively Parallel Simulation of Advanced Monte Carlo Methods	2010
4.2	Lotze, Sutton, Lahlou	Many-core Accelerated LIBOR Swaption Portfolio Pricing	2012
4.2	Tian, Zhu, Klebaner, Hamza	Option Pricing with the SABR Model on the GPU	2010
4.3	Catanzaro, Sundaram, Keutzer	Fast Support Vector Machine Training and Classification on Graphics Processors	2008
4.3	Lin, Chien	Support Vector Machines on GPU with Sparse Matrix Format	2010
4.3	Sun, Tong	CUDA based Fast Implementation of Very Large Matrix Computation	2010
4.3	Couturier, Domas	Sparse systems solving on GPUs with GMRES	2011
4.3	Michels	Sparse-Matrix-CG-Solver in CUDA	2011
4.4	Aldrich, Fernández-Villaverde, Gallant, Rubio-Ramírez	Tapping the supercomputer under your desk: Solving dynamic equilibrium models with graphics-processors	2010
4.4	Aldrich	GPU Computing in Economics	2013
5.2	Timm, Gelenberg, Marwedel, Weichert	Reducing the Energy Consumption of Embedded Systems by Integrating General Purpose GPUs	2010

Tabelle 1: Literaturübersicht zur vorliegenden Arbeit
Quelle: Eigene Darstellung

Die Literaturrecherche wurde mit bekannten Datenbanken begonnen. Springer Link, JSTOR, ScienceDirect, IEEE Xplore, SSRN und Google Scholar sind hier zu nennen. Weitere Literatur konnte anschließend über die Backward-Suche generiert werden. In der weiteren Recherche wurde insbesondere mit der Datenbank der Website gpucomputing.net gearbeitet, auf der explizit Forschungsbeiträge zum Thema GPU-Computing zu finden sind.[12]

1.4 Aufbau der Arbeit

Kapitel 2 umfasst eine Einführung in die Thematik, um ein solides Grundverständnis zu erlangen. In Kapitel 3 wird die ausgewählte Literatur vorgestellt. Von Bedeutung sind hier insbesondere die Systemkonfiguration und die erzielten Beschleunigungen. Der ökonomische Aspekt des Einsatzes der Grafikkarte wird in Kapitel 4 behandelt, anschließend folgt eine Diskussion der Ergebnisse samt Limitation. Zuletzt erfolgt ein Fazit, bevor diese Arbeit mit einem Ausblick abgeschlossen wird.

[12] Vgl. Anhang 1, S. 58.

6. Fazit und Ausblick

Im Rahmen der vorliegenden Arbeit wurde die Sinnhaftigkeit der parallelen Datenverarbeitung für wissenschaftliche Optimierungsprobleme, unter dem Einsatz der GPU, erörtert. In unterschiedlichen Disziplinen wurde untersucht, ob Grafikkarten signifikante Beschleunigungen der Berechnungszeiten erzielen können und überdies hinaus wurde der Energieverbrauch analysiert.
Die Auswertungen der vorgestellten Literatur zeigen, dass der Einsatz der GPU unter gewissen Umständen vorteilhaft ist. Im Folgenden werden die in der Einleitung formulierten Forschungsfragen beantwortet.
Zunächst wird auf die Frage eingegangen, welche Optimierungsprobleme sich überhaupt für den Einsatz des Grafikprozessors eignen. Grafikkarten sind für die Berechnung rechenintensiver Probleme optimal, wenn eine Parallelisierung der Operationen möglich ist. Folglich können nicht alle Probleme von dem Einsatz der GPU profitieren. Sofern sich ein sequentieller Algorithmus nicht in parallele Teile zerlegen lässt, profitiert er nicht von dem Einsatz der Grafikkarte. Auch bei parallelen Algorithmen ist ein Nutzen nicht garantiert, wenn die GPU auf den ganzen Datensatz zugreift. Aufgrund des beschränkten Speichers sind GPUs darauf ausgerichtet, kleinere Datenströme zu bearbeiten. Der größte Speedup wird erzielt, wenn dieselben arithmetischen Operationen zeitgleich ausgeführt werden. Ein effektives Arbeiten mit der GPU wird begünstigt durch Separation der Daten und Parameter.[159]
Die sich daran anschließende Forschungsfrage zielt auf die tatsächliche Höhe der Beschleunigung ab. Gegenüber konventionellen Hauptprozessoren lassen sich die aufgetragenen Aufgaben um ein vielfaches schneller erledigen. Beschleunigungen konnten bis zu dem Faktor 208 aufgezeigt werden. Es wurde festgestellt, dass die Beschleunigung höher ist, je komplexer das Problem ist. Anderenfalls fallen die Beschleunigungen deutlich geringer aus, aber die Rechenzeit ist in einer Vielzahl der Fälle auch erhöht, sofern sehr geringe Parameterwerte gewählt werden. Weiterhin wird die Beschleunigung durch eine Vielzahl von Faktoren beeinflusst, wobei in dieser Arbeit nicht alle Faktoren berücksichtigt werden konnten. Die Speedups sind von Anwendungsfall zu Anwendungsfall unterschiedlich ausgefallen. Die Ergebnisse haben gezeigt, wenn sich Optimierungsprobleme massiv-parallel ausführen lassen, dass der Einsatz eines Grafikprozessors zu einer signifikanten Verringerung der Berechnungszeit führt. Die abschließende Forschungsfrage ist auf die Energieeffizienz von Grafikkarten gerichtet. Im Falle einer reduzierten Rechenzeit, arbeitet die Grafikkarte energieeffizienter. Je größer der Speedup ist, desto größer ist auch die Verringerung der verbrauchten Energie.

[159] Vgl. Zhou et al. 2010, S. 2.

Ziel dieser Arbeit war es, den Einsatz der Grafikkarte als universales Rechenwerkzeug zu beurteilen, sowohl in puncto Effizienz als auch in puncto Wirtschaftlichkeit. Insgesamt gesehen, kann für die Grafikkarte bei massiv-paralleler Datenverarbeitung eine klare Handlungsempfehlung ausgesprochen werden. Sofern eine Parallelisierung von Operationen nicht möglich ist, ist der Einsatz der GPU nicht gerechtfertigt. Für eine bestmögliche Leistung muss nicht nur die Architektur, sondern auch der Algorithmus verbessert werden.[160] Die GPU ist nicht nur aufgrund der Zeitersparnis, sondern auch aufgrund der wirtschaftlicheren Arbeitsweise eine Integration wert.
Zuletzt ist anzumerken, dass die vollständige Ausschöpfung der GPU ein Akt ist, dessen Bewältigung Monate oder gar Jahre benötigen kann.[161] Bei engagiertem Lernen schätzen Lee et al., dass CUDA für einen kompetenten C-Programmierer binnen weniger Wochen zu erlernen ist.[162] Abschließend wird ein Ausblick in die zukünftigen Entwicklungen in der Forschung gegeben.

Der Entwicklung von Prozessoren mit multiplen Kernen wird weiter voranschreiten. Das liegt insbesondere daran, dass die Industrie enorme Summen in die Forschung und Entwicklung investiert. Der milliardenschwere GPU-Spezialist Nvidia erzielte im abgelaufenen Geschäftsjahr einen Umsatz von knapp fünf Milliarden Dollar und steckt fast ein Drittel seines Umsatzes in die Forschung und Entwicklung. Im Grafikgeschäft konnte zudem ein Wachstum von 11% erzielt werden. Die Tesla Sparte zeigte ebenfalls einen Zuwachs, daher sind die Entwicklungsabsichten keinesfalls nur der Spieleindustrie zuzuschreiben. Die tatkräftige Unterstützung der Industrie ist folglich aufgrund wirtschaftlicher Interessen definitiv gesichert.[163,164]
Einen aufschlussreichen Blick in die Zukunft des GPU-Computing gewährt Jack Dongarra, einer der international führenden Experten im Bereich der parallelen Datenverarbeitung.

> *„GPUs have evolved to the point where many real-world applications are easily implemented on them and run significantly faster than on multi-core systems. Future computing architectures will be hybrid systems with parallel-core GPUs working in tandem with multi-core CPUs.“*[165]

Die Aussagen von Dongarra zielen darauf ab, dass der Erfolg der Grafikkarte auch in kommender Zeit bestehen wird. Mit innovativen Computing-Architekturen wird eine weitere Steigerung der Systemleistung erfolgen. Aber auch für Anwendungen ist es von Bedeutung, eine Leistungssteigerung durch eine Programmierung für multiple

[160] Vgl. Lin und Chien 2010, S. 313.
[161] Vgl. Brodtkorb et al. 2013, S. 1.
[162] Vgl. Lee et al. 2010a, S. 3.
[163] Vgl. Rißka 2015.
[164] Vgl. Mantel 2015.
[165] Vgl. Shi et al. 2013, S. 222.

Kerne zu erzielen. Für Nutzer, die mit geringem Aufwand einen moderaten Speedup erzielen wollen, gilt es Programmbibliotheken zu entwickeln, die aus dem existierenden Code automatisch einen Code generieren, der auf der GPU läuft.[166]

[166] Vgl. Lee et al. 2010a, S. 19.