Der wichtigste Beitrag dieser Dissertation ist es aufzuzeigen, dass Grafikprozessoren (GPUs) als Repräsentanten der Entwicklung hin zu Vielkern-Architekturen sehr gut geeignet sind zur schnellen und genauen Lösung großer, dünn besetzter linearer Gleichungssysteme, insbesondere mit parallelen Mehrgittermethoden auf heterogenen Rechenclustern. Solche Systeme treten bspw. bei der Diskretisierung (elliptischer) partieller Differentialgleichungen mittels finiter Elemente auf. Wir demonstrieren Beschleunigungsfaktoren von mindestens einer Größenordnung gegenüber konventionellen, hochoptimierten CPU-Implementierungen, ohne Verlust von Genauigkeit und Funktionsumfang. Im Detail liefert diese Dissertation die folgenden Beiträge:

Berechnungen in einfach genauer Fließkommadarstellung können für die hier betrachteten Problemklassen nicht ausreichen. Wir greifen die Methode gemischt genauer iterativer Verfeinerung (Nachiteration) wieder auf, um nicht nur die Genauigkeit von berechneten Lösungen zu verbessern, sondern vielmehr die Effizienz des Lösungsprozesses als ganzes zu steigern. Sowohl auf CPUs als auch auf GPUs demonstrieren wir eine deutliche Leistungssteigerung ohne Genauigkeitsverlust im Vergleich zur Berechnung in höherer Fliesskomma-Genauigkeit.

Wir präsentieren effiziente Parallelisierungstechniken für Mehrgitter-Löser auf Grafik-Hardware, insbesondere für numerisch starke Glätter und Vorkonditionierer, die für stark anisotrope Gitter und Operatoren geeignet sind. Ein Beispiel ist die Entwicklung einer effizienten Reformulierung des Verfahrens der zyklischen Reduktion für die Lösung tridiagonaler Gleichungssysteme. Im Hinblick auf Hardware-orientierte Numerik analysieren wir sorgfältig den Kompromiss zwischen numerischer und Laufzeit-Effizienz für inexakte Parallelisierungstechniken, die einige der inhärent sequentiellen Charakteristiken solcher starker Glätter zugunsten besserer Parallelisierungseigenschaften entkoppeln.

Die Reimplementierung großer, etablierter Softwarepakete zur Anpassung auf neue Hardwareplattformen ist oft inakzeptabel teuer. Wir entwickeln einen "minimalinvasiven" Zugang zur Integration von Co-Prozessoren wie GPUs in FEAST, einem exemplarischen finite Elemente Diskretisierungs- und Löserpaket. Der Hauptvorteil unserer Technik ist, dass Applikationen, die auf FEAST aufsetzen, nicht geändert werden müssen um von der Beschleunigung durch solche Co-Prozessoren zu profitieren. Wir evaluieren unseren Zugang auf großen GPU-beschleunigten Rechenclustern für klassische Benchmarkprobleme aus der linearisierten Elastizität und der Simulation stationärer laminarer Strömungsvorgänge, und beobachten gute Beschleunigungsfaktoren und gute schwache Skalierbarkeit. Die maximal erreichbare Beschleunigung wird zudem analysiert und theoretisch modelliert, um bspw. Vorhersagen treffen zu können.

Weiterhin fassen wir die historische Entwicklung des Forschungsgebiets "wissenschaftliches Rechnen auf Grafikhardware" seit 2001/2002 zusammen, d.h. die Entwicklung von GPGPU als obskures Nischenthema hin zum fachübergreifenden Einsatz heute. Die Darstellung umfasst gleichermaßen die Hardware und das Programmiermodell und beinhaltet eine ausgiebige Bibliografie von Veröffentlichungen im Bereich der Simulation von PDE-Problemen auf GPUs.

Stichworte:
Wissenschaftliches Rechnen
Hardware-orientierte Numerik
Finite Elemente
Mehrgitter-Verfahren
Krylov-Unterraum-Verfahren
Iterative Lösungsverfahren
Tridiagonale Löser
Zyklische Reduktion
Gemischt genaue iterative Verfeinerung
Nachiteration
Double-Single Emulation
Schlecht konditionierte lineare Gleichungssysteme
Dünn besetzte lineare Gleichungssysteme
Lokal strukturierte Systeme

Poisson-Problem
Stationäre laminare Strömungssimulation
Computational Fluid Dynamics
Linearisierte Elastizität
Computational Solid Mechanics

Hochleistungsrechnen
Heterogenes Rechnen
Hybrides Rechnen
Grafikprozessoren (GPUs)
GPGPU
GPU Computing
NVIDIA CUDA

Gebietszwelegung
Paralleles Rechnen
Verteiltes Rechnen
Feingranulare Parallelität
Mehrfarben-Parallelisierung