

Preview

- Überschriften
 - Abstract
 - Overview
 - Concepts and Terminology
 - Parallel Computer Memory Architecture
 - Parallel Programming Models
 - Designing Parallel Programs
 - Parallel Examples
 - References and more Information
- Abbildungen
 - Kuchendiagramm
 - Vergleich sequentielle / parallele Verarbeitung
 - Codeschnipsel
 - Veranschaulichung Komplexität
 - Formeln
 - diverse andere Abbildungen, die ohne begleitende Erklärung nur bedingt verständlich sind
- Tabellen
 - Vergleich zwischen gemeinsam genutzten und verteilten Speicherarchitekturen

Question

- Welche Problemstellungen werden durch paralleles Programmieren besser gelöst?
- Welche Nachteile hat paralleles Programmieren im Gegensatz zum sequentiellen Programmieren?
- Welche Implementierungen existieren bereits?
- Welche Geschwindigkeitsvorteile sind zu erwarten?
- Was sind parallele Algorithmen bzw. was gibt es bei denen zu beachten?

Read

- http://www.llnl.gov/computing/tutorials/parallel_comp/

Reflect

Der Text behandelt viele der Fragen, die vor dem eigentlichen Lesen gestellt wurden. Bei den Dateisystemen, die Parallelität unterstützen, wird XFS von SGI nicht genannt.

Recite

Der Text behandelt kurz die Grundlagen der heute üblichen Computerarchitektur basierend auf dem Modell von von Neumann. Es wird eine Klassifizierung von parallel arbeitenden Computern vorgenommen (einfach / mehrfache Instruktionen / Daten, 4 Kombinationen). Die Unterscheidung zwischen geteilten Speicher, bei dem alle CPUs auf einen gemeinsamen Speicher zugreifen, und verteilten Speicher, bei dem jede CPU ihren eigenen Speicher besitzt und auf den der anderen nur per Nachrichten zugreift, wird eingeführt. Der Text behandelt verschiedene Ansätze zur Parallelisierung von Programmen, wobei angemerkt wird, dass oftmals mehrere dieser Ansätze in Kombination zum Einsatz kommen. Sequentielle Programme können entweder manuell oder automatisch parallelisiert werden, wobei beim letzteren Einsatz Compiler mit Hilfe von Heuristiken zur Parallelisierung geeignete Stellen suchen, aber auch immer die Gefahr

besteht, dass die Heuristiken falsch angewendet werden und ggf. zu langsameren Code führen. Manuelle Parallelisierung ist meistens schwierig und fehleranfällig und muss häufig iterativ angegangen werden. Der Text behandelt in einem langen Abschnitt andere Aspekte, die bei der Parallelisierung von Programmen beachtet werden müssen. Nicht alle Probleme sind parallelisierbar. Bei den parallelisierbaren Problemen gibt es zwei grundlegende Ansätze: Das Problem anhand der Daten oder der Instruktionen in diskrete Einheiten aufbrechen. Zusätzlich müssen noch unter anderem I/O Belastung, Kommunikationsoverhead und möglichst gleichmäßige Auslastung der parallel arbeitenden Einheiten beachtet werden. Zum Schluss wird Parallelisierung an mehreren Beispielen demonstriert.

Review

Ein einführender Text, der einen breiten und anscheinend umfassenden Überblick über verschiedene Aspekte der parallelen Programmierung gibt ohne jedoch oberflächlich zu wirken. Der Text ist sachlich verfasst. Die Materie wird einem Stück für Stück näher gebracht, obwohl Grundkenntnisse der Informatik sicherlich an vielen Stellen behilflich sind. Zahlreiche Abbildungen ergänzen den Text. Viele Aspekte werden detailliert behandelt und es ergeben sich zahlreiche Punkte, an denen man mit noch detaillierterer Literatur über Einzelaspekte anknüpfen kann. Mehr hinweise auf weiterführende Literatur wäre jedoch wünschenswert gewesen, eine Empfehlung im WWW zu suchen erscheint unzureichend.