

Parallel Computing I

Einführung in das
Hochleistungsrechnen

Einführung

24. April 2012
Paralleles Rechnen
SS 2012
Thorsten Grahs

Aufbau

Termine & Ansprechpartner

- VL: Mo. 11:30 - 13:00 RZ 124
Thorsten Grahs (t.grahs@tu-bs.de)
RZ 120
- Üb. Do. 8:00 - 9:30 G40
Marcell Kehmstedt (m.kehmsedt@tu-bs.de)

Scheinkriterien:

- min. 50 d. Punkte aus den Übungen
Voraussetzung zur
- Mündliche Prüfung

Schwerpunkt

Verteilte Systeme

MPI (Message Passing Interface)

- Hochleistungsrechnen
- Beschleunigung von numerischen Simulationen
- Einzelprozessorperformance
- Speicher- und Rechnerarchitekturen
- Performancebetrachtungen
- Programmiermodelle
- Message Passing Interface (MPI)
- Iterative Löser/Bibliotheken

Parallel Computing

Wissenschaftliche Disziplin im Spannungsfeld von

- Informatik
- Mathematik
- Ingenieurwesen

Problemstellungen meist aus

- Ingenieurs-/Geowissenschaften
 - Klimamodellierung
 - Wetter/Sturmflutvorhersage
 - Simulation komplexer Systeme (Flugzeug, Fahrzeug)
 - Sternbildung (Astronomie)
 - Molekülsimulation (Chemie)
(Vorhersage Stoffeigenschaften)

Parallel Computing

Wissenschaftliche Disziplin im Spannungsfeld von

- Informatik
- Mathematik
- Ingenieurwesen

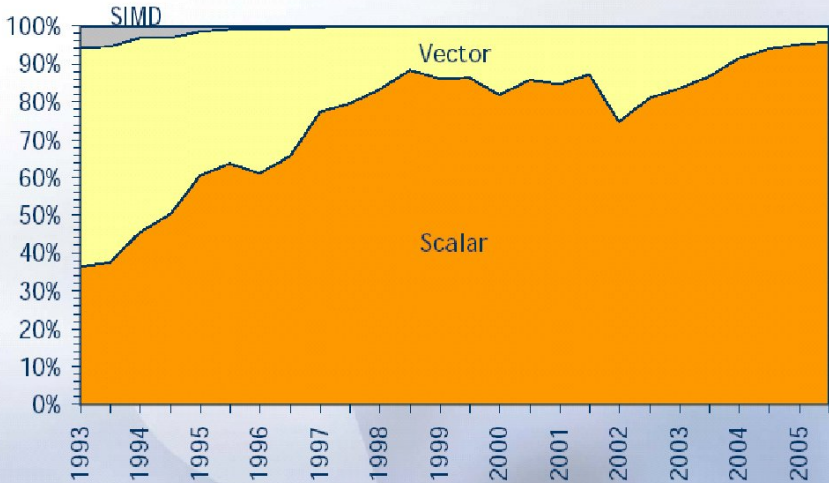
Lösungen durch:

- Mathematik
 - Numerischen Algorithmen
 - Diskretisierung partieller Differentialgleichungen
 - Iterative Gleichungssystemlöser
 - Domain Decomposition
 - Matrix/Vektor-Manipulationen
- Informatik
 - effiziente Algorithmen
 - Kommunikation

Umbrüche im Parallelen Rechnen

- „Ablösung“ der Dominanz der Vektorrechner durch Cluster Computing (Beowulf-Projekt)
 - Einsatz und Zusammenschluss von vielen Einzelrechner aus Standardkomponenten
 - kostengünstig
 - große Probleme
- GPUs (Graphical Processor Units)
 - getrieben durch Spielkonsolen/-industrie
 - Anwendung im wissenschaftlichen Rechnen
 - notwendig sind entsprechende Datenstrukturen

Umbrüche im Parallelen Rechnen



Cluster Computing

Alter Wein in neuen Schläuchen

Paralleles Rechnen auf Systemen mit verteiltem Speicher

- Jahrzehntlang Randgebiet der Computerwissenschaften
- Paradigmenwechsel:
 - Probleme werden größer
 - Gap zwischen Vektorrechner und Standard-PCs kleiner
 - Standardkomponenten
 - Freie Betriebssysteme (Linux)

Beowulf-Projekt

Donald Becker & Thomas Sterling
1994, NASA

- Unterschied zu einem COW (Cluster of Workstations)
 - Ansprechbar als ein Rechner
 - 16 Motherboards mit 486DX4 Prozessoren
 - 16MB RAM pro Board,
 - Festplatte mit je 500 MB pro Board,
- Open Source Software
 - Unix/Linux
 - PVM/MPI

Beispiel: Wettervorhersage

Numerische Simulation der Atmosphäre

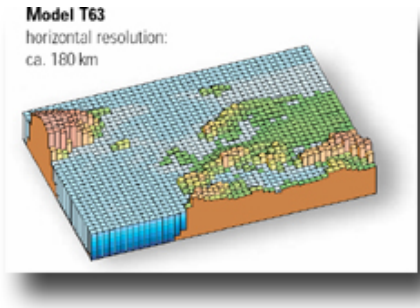
- Diskretisierung der Lufthülle
 - Repräsentation durch 3-dimensionales Gitter
 - Berechnung an jedem Gitterpunkt
-
- 3-Dimensionale Navier-Stokes-Gleichungen
 - Temperatur
 - Luftdruck
 - (Wind-)Geschwindigkeit

Beispiel: Wettervorhersage

Nichtlinearitäten:

Das Wetter in Deutschland hängt auch ab vom

- Azorenhoch
- Islandtief



⇒ Modell umfasst große Skalen

Andererseits müssen lokale Strukturen aufgelöst werden.

Beispiel: Wettervorhersage II

Theor. Wettermodell

- Globales Modell 1 km Grid-Spacing
- Höhe: 20 km
 $\Rightarrow \approx 10^{10}$ Gitterpunkte
- Zeitliche Auflösung abhängig von räumlicher (CFL-Kriterium)
 $\Delta t \approx 10$ Sekunden
 \Rightarrow Simulation für 3 Tage im Voraus
ca 26.000 Zeitschritte
- Berechnung aller phys. Größen
(5 partielle DGLen)
- Annahme: 100 Operationen pro Zeitschritt

2.6×10^{16} Operationen für die Simulation

Beispiel: Wettervorhersage III

$2,6 \times 10^{16}$ Operationen

- PC 10 GigaFLOP
(10×10^9 Floating Point Op./Sek.)
Simulationsdauer: 30 Tage
- Großrechner 1 TerraFLOP
Simulationsdauer: 8 Stunden

Beispiel: Wettervorhersage III

Seymour Cray (1925 – 1996)

If you were plowing a field , what would you rather use?

Two strong oxen for 1024 chickens?

Allerdings: Problem Speicherdichte

- Daten können nicht schnell genug zur CPU gelangen
- in 10^{-12} Sek. legt das Licht 0,3mm zurück
Speicher muss im Radius v. 0,3 mm um CPU angeordnet werden.
- Daten der Simulation
- 20 Zahlen p.Gitterpunkt (T, V, P, k, ...)
- pro Zahle 32 Bit für 10^{10} Gitterpunkte
- $6,4 \times 10^{12}$ Bit

⇒ Speicherdichte von 1 Bit pro Atom.

Beispiel: Wettervorhersage III

W. Groppe, E. Lusk, A. Skjellum *Using MPI*

To pull a bigger wagon, it is easier to add more oxen than to grow a gigantic oxen

Idee:

Nutze z.B. 1000 Standard-Computer mit 10^9 -Operationen p. Sek.

- Domain Decomposition
- Jeder Prozessor behandelt 10^7 Gitterpunkte

Aufgrund der geringeren Geschwindigkeit kann der Abstand Speicher-CPU ca 300 mm betragen

⇒ Benötigter Speicher ca. 800 MByte p. CPU

Rechenleistung

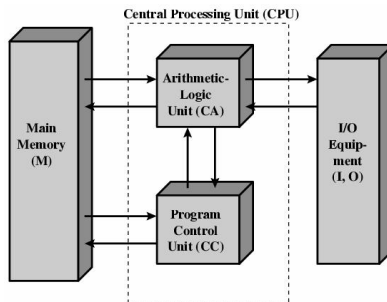
FLOP

Durchführung von elementaren arithmetischen Operationen

Vektoraddition

```
#define N 1000000
unsigned long i;
double a[N], sum=0.0;
...
for (i=0; i<N; ++i)
    sum=sum+a[i];
```

Sequentieller Rechner
(Prozessor P, Speicher M)
von-Neumann-Modell



Befelsliste (Compiler)

- Hole nächsten Befehl aus dem Speicher in das Befehlsregister
- Interpretiere diesen Befehl
- Lade erstes Argument (`sum`) aus dem Speicher in ein Register
- Lade zweites Argument (`a[i]`) aus dem Speicher in ein anderes Register
- Führe den Befehl aus und schreibe Ergebnis in ein drittes Register
- Schreibe das Ergebnis (`sum`) zurück in den Speicher

GFlops = GHZ?

Für vektorielle oder superskalare Prozessoren, allerdings nur unter optimalen Bedingungen (peak performance). Für praktischen Anwendungen ist dies jedoch meist nicht zu erreichen.

Von-Neumann Flaschenhals

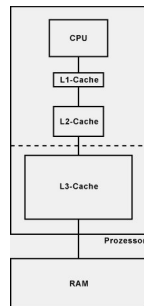
Haupthindernis Speicher (DRAM)

Dynamic Random Access Memory, zb. 1066 MHz

Nur ein Bruchteil der Taktfrequenz moderner Rechner. Hauptspeicher bremst den Rechenprozess des von-Neumann-Modell aus.

Abhilfe

L1,L2,L3-Cache



Performance-Vergleich

Linpack-Benchmark

BLAS (Basic Linear Algebra Subroutines)

Steigerung der Anzahl der Variablen (Matrixgröße)

