Rechnerstrukturen

Vorlesung im Sommersemester 2005

Prof. Dr. Wolfgang Karl

Universität Karlsruhe (TH)

Fakultät für Informatik

Institut für Technische Informatik

Kapitel 1: Grundlagen

1.1 Einführung: Bewertung der Leistungsfähigkeit

Bewertung der Leistungsfähigkeit

- Ziele
  - Auswahl der Rechenanlage
  - Veränderung der Konfiguration einer bestehenden Anlage
  - Entwurf von Anlagen

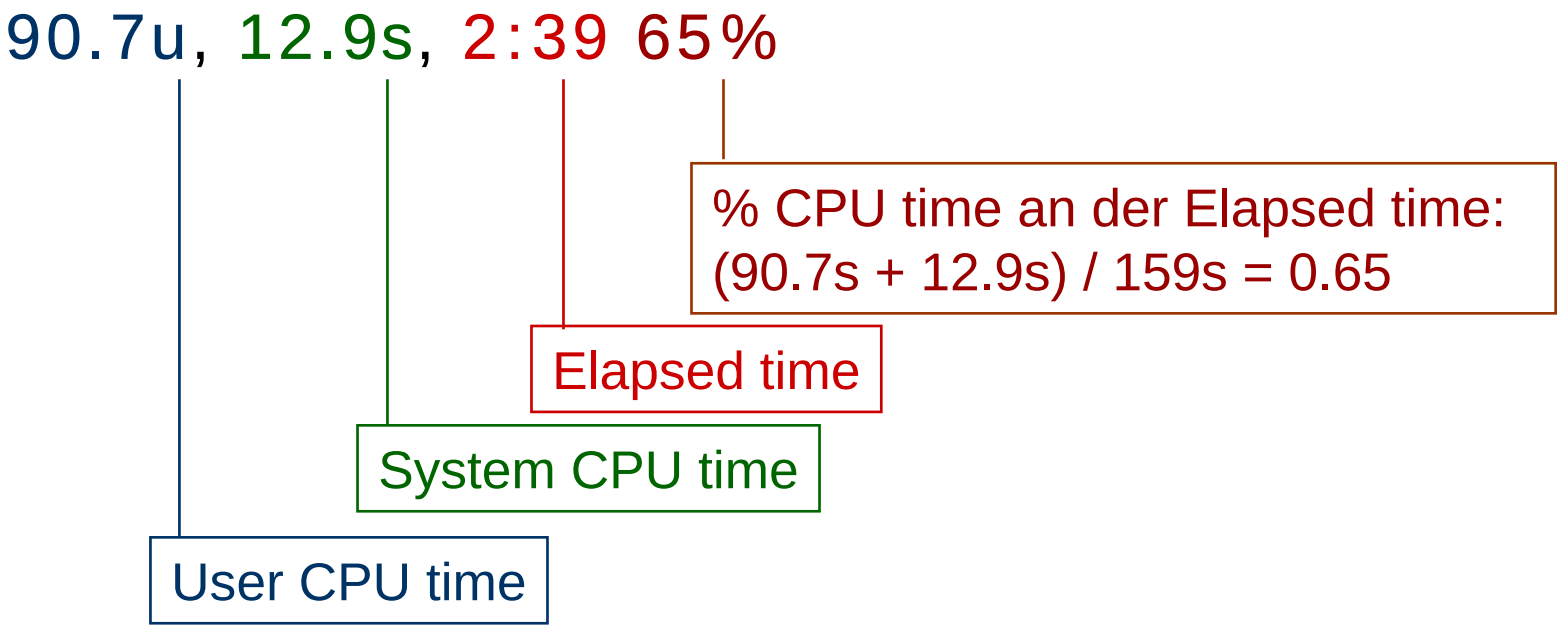
- Was heißt es: Ein Rechner ist schneller als ein anderer Rechner?
  - Der Benutzer eines Arbeitsplatzrechners:
    - „Ein Rechner A ist schneller als ein Rechner B, wenn ein Programm auf A weniger Zeit benötigt.“
    - Reduzierung der Antwortzeit (response time) oder Ausführungszeit (execution time)
      - Zeit zwischen dem Beginn und dem Ende eines Ereignisses, einer Aufgabe
    - A ist *n*-mal schneller als B:

$$\frac{\text{Ausführungszeit(B)}}{\text{Ausführungszeit(A)}} = n$$

- Was heißt es: Ein Rechner ist schneller als ein anderer Rechner?
  - Der Rechenzentrumsleiter:
    - „Ein Rechner A ist schneller als ein Rechner B, wenn A in einer Stunde mehr Aufträge (Jobs) erledigt.“
    - Erhöhung des Durchsatzes (throughput)
      - Anzahl der ausgeführten Aufgaben in einem gegebenen Zeitintervall
    - Durchsatz von A ist *m*-mal höher als der von B:
      - Die Anzahl der erledigten Aufgaben auf A ist *m*-mal die Anzahl der erledigten Aufgaben auf B.

- Ausführungszeit (execution time)
  - Wall-clock time, response time, elapsed time
    - Latenzzeit für die Ausführung einer Aufgabe
      - Schließt den Speicher- und Plattenzugriff, Ein-/Ausgabe etc. mit ein.
  - CPU Time
    - Zeit, in der die CPU arbeitet
      - User CPU Time: Zeit, in der die CPU ein Programm ausführt
      - System CPU Time: Zeit, in der die CPU Betriebssystemaufgaben ausführt, die von einem Programm angefordert werden

- Ausführungszeit (execution time)
  - CPU Time
    - Beispiel: UNIX `time` Kommando:



Bewertung der Leistungsfähigkeit

- Verfahren
  - Auswertung von Hardwaremaßen und Parametern
  - Laufzeitmessungen bestehender Programme
  - Messungen während des Betriebs von Anlagen
  - Modelltheoretische Verfahren

- Gleichung für die Leistung der CPU
  - Der Rechner läuft mit fester Taktrate, angegeben durch
    - Dauer eines Taktzyklus (z. B. 1ns)
    - Taktfrequenz (z. B. 1 GHz)
  - Die CPU-Zeit einer Programmausführung kann dargestellt werden mit

$$\text{CPU-Zeit} = \text{Anzahl Taktzyklen für das Programm} * \text{Taktzyklusdauer} = \frac{\text{Anzahl Taktzyklen für das Programm}}{\text{Taktfrequenz}}$$

- Gleichung für die Leistung der CPU
  - Einführung der Maßzahl CPI (clock cycles per instruction)
    - Mittlere Anzahl der Taktzyklen pro Befehl
    - mit IC (instruction count), der Anzahl der ausgeführten Befehle eines Programms

$$\text{CPI} = \text{Anzahl Taktzyklen für das Programm} / \text{IC}$$

    - Damit

$$\begin{aligned}\text{CPU-Zeit} &= \text{IC} \times \text{CPI} \times \text{Taktdauer} \\ &= \text{IC} \times \text{CPI} \ / \ \text{Taktfrequenz}\end{aligned}$$

- Gleichung für die Leistung der CPU
  - Konsequenz
    - CPU-Leistung ist bestimmt durch
      - Taktfrequenz
        - » Abhängig von der Halbleitertechnologie
      - CPI
        - » Abhängig von der Implementierung, ISA (Befehlssatzarchitektur), Programm
      - IC
        - » Abhängig von ISA, Compilertechnologie, Programm

- Maßzahlen für die Operationsgeschwindigkeit
  - Analog lässt sich für eine Programmausführung definieren:
    - MIPS: Millions of Instructions Per Second

$$\text{MIPS} = \frac{\text{Anzahl der ausgeführten Instruktionen}}{10^6 \times \text{Ausführungszeit}}$$

    - MFLOPS: Millions of Floating Point Operations Per Second

$$\text{MFLOPS} = \frac{\text{Anzahl der ausgeführten Gleitkommaoperationen}}{10^6 \times \text{Ausführungszeit}}$$

- Probleme mit diesen Maßzahlen
  - Abhängigkeit von ISA und ausgeführter Befehlssequenz
  - Problem:
    - Vergleich von Rechnern mit unterschiedlicher ISA
    - Unterschiedliche MIPS/MFLOPS-Zahlen bei verschiedenen Programmen
    - MIPS kann umgekehrt zur Leistung variieren
      - Beispiel: Gleitkommarechnung in Hardware bzw. mit Software-Routinen
    - MIPS/MFLOPS Angaben von Herstellern oft nur best-case-Annahme: theoretische Maximalleistung

- Zusammenfassung
  - Vergleich von Rechnern bezüglich ihrer Leistung ohne großen Aufwand
  - Maßzahlen bewerten nur spezielle Aspekte
  - Kritische Betrachtung der Leistungsangabe unbedingt notwendig!
  - Angabe einer hypothetische Maximalleistung!

Laufzeitmessung bestehender Programme

- Benchmarks
  - Bewertung der Leistungsfähigkeit aufgrund von Messungen
    - Programm oder Programmsammlung im Quellcode
    - Übersetzung notwendig
    - Messung der Ausführungszeiten
    - In die Bewertung fließt „Güte“ des Compiler und Betriebssoftware ein
    - Zugriff auf die Maschinen notwendig

- „Spiele“-Benchmarks
  - 10- 100 Zeilen Code
  - Beispiele:
    - Sieb des Erathostenes
    - Puzzle
    - Queens
    - Quicksort
- „Echte“ Benutzerprogramme
  - Verwendung typischer Benutzer- oder Systemprogramme
  - Oft bei Kauf von Großrechnern

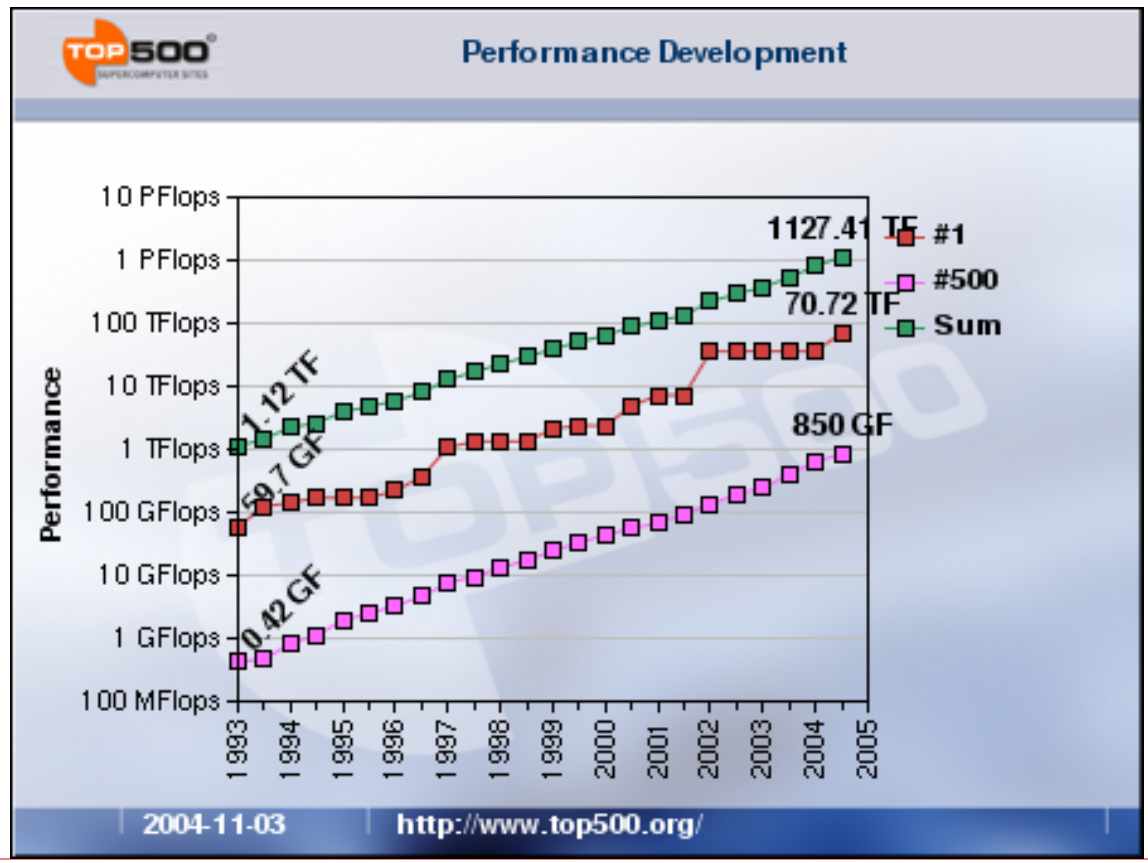
- Synthetische Benchmarks
  - Charakteristisch für bestimmte Anwendungsklasse
  - Abbilden der Häufigkeit von Befehlen und Operanden aus einer Menge von Programmen
  - „Simuliert“ typisches Anwenderprogramm

- Synthetische Benchmarks
  - Whetstone (~1970)
    - In FORTRAN
    - Überwiegend Gleitkommaoperationen
      - *„This program is the result of extensive research to determine the instruction mix of a typical Fortran program. The results of this program on different machines should give a good indication of which the machine performs better under a typical load of Fortran programs. The statement a purposely arranged to defeat optimizations by the compiler.“* (H.J. Curnow and B.A. Wichmann, Comments on the Whetstone Benchmark, 1976)

- Synthetische Benchmarks
  - Dhrystone (~1984)
    - In C geschrieben
    - Auswahl der Anweisungen aufgrund statistischer Analysen über die Verwendung von Sprachkonstrukten in Programmen
    - Keine Bedeutung mehr im Bereich Arbeitsplatzrechner, aber noch Einsatz im Bereich eingebetteter Systeme: problematisch!
    - *„Dhrystone does not use floating point. Typical programs don‘t...“* (Rick Richardson, Clarification of Dhrystone, 1988)

- Kernels
  - Rechenintensive Teile realer Programme
  - Vorwiegend numerische Algorithmen
  - Beispiele:
    - Lawrence Livermore Loops:
      - Zur Bewertung vektorisierender Compiler
    - LINPACK Softwarepaket:
      - Lösung eines Systems linearer Gleichungen
      - Verwendet für die TOP500 Liste (http://www.top500.org)
    - BLAS (Basic Linear Algebra Subprograms)
  - Wenig Aufwand, aber nur bedingt aussagekräftig

- Kernels
  - LINPACK Softwarepaket:
    - Verwendet für die TOP500 Liste



- Standardisierte Benchmarks
  - Ziel: Vergleichbarkeit von Rechnern (inkl. Betriebssystem und Compiler)
  - Anforderungen:
    - Gute Portierbarkeit
    - Repräsentativ für typische Nutzung der Rechner
  - Sammlung von Benchmark-Programmen (Benchmark Suites)
    - Ausgeglichene Bewertung durch die unterschiedlichen Eigenschaften der Programme

- Standardisierte Benchmarks
  - Standardisierungsorganisationen
    - TPC (Transaction Processing Performance Council)
      - Mitte der 80‘er Jahre, http://www.tpc.org
      - Zusammenschluss von Datenbank- und Rechnerherstellern
      - Ziel: Bewertung von Datenbanksystemen
    - EEMBC (Embedded Microprocessor Benchmark Consortium)
      - Anwendungen aus dem Bereich Eingebettete Systeme