

Leistungsbewertung:

meist Rechengeschwindigkeit, aber auch Stromverbrauch, Zuverlässigkeit, ...

Ziele der Leistungsbewertung

- Auswahl eines Rechnersystems
- Veränderung der Konfiguration eines bestehenden Systems
- Optimierung des Systems
- Entwurf von Rechnern

Wann ist ein Rechner schneller?

"... wenn Programm weniger Zeit benötigt" - PC-User

Reduzierung der Antwortzeit oder Ausführungszeit

A ist n -mal so schnell wie B: $\frac{\frac{1}{T_A}}{\frac{1}{T_B}} = n$

"... mehr Aufträge in einer Stunde erledigt" - Rechenzentrum

Erhöhung des Durchsatzes

Ausführungszeit:

Wall-clock time, response time, elapsed time

Latenzzeit für Ausführung einer Aufgabe (+ Speicherzugriff, Ein/Ausgabe, ...)

CPU-Zeit

User CPU Time: Zeit, in der die CPU ein Programm ausführt

System CPU Time: Zeit, in der die CPU Betriebssystemaufgaben ausführt

Programmunabhängige Leistungsmodelle

Taktfrequenz: ungeeignete Metrik, wird in Werbung verwendet

MIPS: Million Instructions per second

$$\text{MIPS} = \frac{N(\text{Instr})}{10^6 \text{ pro sec}}$$

CPI: Cycles per Instruction

IPC: Instructions per Cycle

native MIPS, peak MIPS, relative MIPS

MIPS als Leistungsindikator wenig gut geeignet

Programmabhängige Leistungsmodelle

$$\text{Performance: } P = \frac{1}{T(\text{Programm}X)}$$

$$\text{Ausführungszeit: } T(\text{Programm}X) = N_{\text{Instr}}(\text{Programm}X) \cdot \text{CPI}(\text{Programm}X) \cdot t_{cy}$$

CPI:

- ohne Super Skalartät: theoretisch 1 (1,2 - 1,5)
- n-Fach Issue-Architekturen (z.B.: 3-fach: PIII): $\text{CPI}_{\text{min}} = \frac{1}{3}$ (0,6 ... 0,9)
- Benchmarks so, dass Code und Daten nicht in Cache passen

Relative Performance:

$$\frac{P(RA)}{P(RB)} = \frac{1}{T(\text{Prog}X, RA)} : \frac{1}{T(\text{Prog}X, RB)}$$

Teiloptimierung:

$$T_{\text{NEU}} = \frac{T_{\text{OPTIMISIERBAR}}}{\text{Verbesserungsfaktor}} + T_{\text{NICHT OPT.}} \quad (= \text{Amdahlsches Gesetz})$$

Parallelverarbeitung:

$$T_{\text{NEU}} = \frac{T_{\text{parallelisierbar}}}{p} + T_{\text{Nicht Parall.}} + OH$$

Speedup: Verbesserung durch Parallelisierung:

$$SU = \frac{T_{\text{act}}}{T_{\text{NEU}}} = \frac{P_{\text{NEU}}}{P_{\text{act}}}$$

Effizienz:

$$E = \frac{SU}{p}$$