

Vorlesung Rechnerstrukturen

□ Kapitel 1: Grundlagen

◆ 1.5 Bewertung der Leistungsfähigkeit

Ziele bei der Bewertung der Leistungsfähigkeit

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

Vorlesung Rechnerstrukturen

1.5 Bewertung der Leistungsfähigkeit

Verfahren zur Bewertung der Leistungsfähigkeit

- ♦ Auswertung von Hardwaremaßen und –parametern
- ♦ Laufzeitmessungen bestehender Programme
- ♦ Messungen während des Betriebs von Anlagen
- ♦ Modelltheoretische Verfahren

Auswertung von Hardwaremaßen

□ Auswertung von Hardwaremaßen und Parametern

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

Auswertung von Hardwaremaßen

□ Mixe

- Theoretische Berechnung
- Berechnung der **mittleren Operationszeit T** aus den Operationszeiten von n Befehlen:

$$T = \sum_{i=1}^n p_i * t_i$$

mit

T : mittlere Operationszeit

n : Anzahl der betrachteten Befehle

t_i : Operationszeit des Befehls i

p_i : relative Häufigkeit des Auftretens (Gewicht) des Befehls i , wobei gelten muß:

$$\sum_{i=1}^n p_i = 1, \quad p_i \leq 1$$

Auswertung von Hardwaremaßen

□ Kernprogramme

- Typische Anwendungsprogramme, die für einen zu bewertenden Rechner geschrieben werden
- Berechnung der Ausführungszeit anhand der Operationszeiten der Befehle
- Aufwendig

⇒ Keine Bedeutung mehr!

Laufzeitmessungen bestehender Programme

□ Maßzahlen für die Operationsgeschwindigkeit

- ◆ MIPS: Millions of Instructions Per Second
 - ◆ $\text{MIPS} = \text{Anzahl der Befehle} / \text{Ausführungszeit} \times 10^6$
 - ◆ $\text{Ausführungszeit} = \text{Anzahl der Befehle} \times 10^6$
 - ◆ MIPS ist abhängig vom Befehlssatz
 - ◆ Schwierig: Rechner mit unterschiedlichen Befehlssätzen zu vergleichen
 - ◆ Unterschiedliche MIPS-Zahl auf einem Rechner bei verschiedenen Programmen
 - ◆ MIPS kann umgekehrt zur Leistung variieren
- ◆ MFLOPS: Millions of Floating Point Instructions Per Second

Laufzeitmessungen 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

Benchmarks

□ „Spiele“-Benchmarks

- 10- 100 Zeilen Code
- Sieb des Erathostenes
- Puzzle
- Queens
- Quicksort

▪ „Echte“ Benutzerprogramme

- Verwendung typischer Benutzer- oder Systemprogramme
- Oft bei Kauf von Großrechnern

Benchmarks

□ Synthetische Benchmarks

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

- Whetstone (~1970)
 - Meist in FORTRAN
 - Überwiegend Gleitkommaoperationen
- Dhrystone (~1984)
 - In C geschrieben
 - Auswahl der Anweisungen aufgrund statistischer Analysen über die Verwendung von Sprachkonstrukten in Programmen
 - Keine Bedeutung mehr!

Benchmarks

□ Kernels

- ◆ Teile von realen Programmen
- ◆ Vorwiegend numerische Programme
- ◆ Lawrence Livermore Loops (vektorisierende Compiler)
- ◆ LINPACK Softwarepaket, Lösung eines Systems von linearen Gleichungen
 - ◆ Vergleich von Supercomputern
 - ◆ TOP500 Liste (<http://www.top500.org>)
- ◆ BLAS (Basic Linear Algebra Subprograms)

Benchmarks

□ Standardisierte Benchmarks

- ◆ Portierbarkeit
- ◆ Vergleichbarkeit von Rechnern
- ◆ Standardisierungsorganisationen
 - ◆ SPEC (Standard Performance Evaluation Corporation)
 - ◆ Gegründet 1988, <http://www.spec.org>
 - ◆ Zusammenschluss von mehr als 40 Firmen (Rechnerhersteller)
 - ◆ Festlegung von Richtlinien für eine gemeinsame Rechnerbewertung
 - ◆ TPC (Transaction Processing Performance Council)
 - ◆ Mitte der 80'er Jahre, <http://www.tpc.org>
 - ◆ Zusammenschluss von Herstellern
 - ◆ Datenbankbereich
 - ◆ EEMBC Benchmarks
 - ◆ Anwendungen aus dem Bereich Eingebettete Systeme
 - ◆ PARKBENCH
 - ◆ Parallelverarbeitung, <http://www.netlib.org/parkbench>

Benchmarks

□ SPEC

◆ Philosophie von SPEC

„The goal of SPEC is to ensure that the marketplace has a fair and useful set of metrics to differentiate candidate systems. The path chosen is an attempt to balance between requiring strict compliance and allowing vendors to demonstrate their advantages. The belief is that a good test that is reasonable to utilize will lead to a greater availability of results in the marketplace.

The basic SPEC methodology is to provide the benchmarker with a standardized suite of source code based upon existing applications that has already been ported to a wide variety of platforms by its membership. The benchmarker then takes this source code, compiles it for the system in question and then can tune the system for the best results. The use of already accepted and ported source code greatly reduces the problem of making apples-to-oranges comparisons.“

Benchmarks

□ Verschiedene Untergruppen in SPEC

- ◆ Open Systems Group (OSG)
 - ◆ **SPEC CPU 2000**: Prozessorleistung
 - ◆ **SPEC JBB2000**: JAVA Business Benchmark
 - ◆ **SPEC JVM98**: Vergleich von Java Virtual Machine Client Plattformen
 - ◆ **SPEC MAIL2001**: Mail Server Benchmark
 - ◆ **SPEC SFS 2.0**: System: File Server
 - ◆ **SPEC WEB99**: Test für WWW Server
- ◆ High Performance Group (HPG)
 - ◆ **SPECchpc96**: Industrielle große Anwendungen
- ◆ Graphics Performance Characterization Group (GPC)
 - ◆ Zusammenschluss mit SPECopc: OpenGL Leistungsbewertung

Beispiel SPEC CPU2000

□ SPEC CINT2000

Benchmark	Referenzzeit	Bemerkung
164.gzip	1400	Datenkompression
175.vpr	1400	FPGA Layout und Platzierung
176.gcc	1100	C Compiler
181.mcf	1800	Lineare Optimierung
186.crafty	1000	Schachprogramm
197.parser	1800	Sprachverarbeitung
252.eon	1300	Ray Tracing
253.perlbnk	1800	Perl
256.gap	1100	Gruppentheorie
255.vortex	1900	Datenbank
256.bzip2	1500	Datenkompression
300.twolf	3000	Platzierung und Routing Simulator

Beispiel SPEC CPU2000

□ SPEC CFP2000

Benchmark	Referenzzeit	Bemerkung
168.wupwize	1600	Quantenphysik
171.swim	3100	Wettervorhersage
172.mgrid	1800	3D-Mehrgitterverfahren
173.applu	2100	Lösung parabolischer u. elliptischer part. DG
177.mesa	1400	3D Graphikbibliothek
178.galgel	2900	Strömungsmechanik
179.art	2600	Neuronales Netzwerk Simulation
183.equake	1300	Finite Elemente Simulation: Erdbeben
187.facerec	1900	Bildererkennung
188.ammmp	2200	Chemie
189.lucas	2000	Zahlentheorie
191.fma3d	2100	Finite Elemente: Crash-Simulation
200.sixtrack	1100	Teilchenbeschleunigermodell
300.apsi	2600	Schadstoffverteilung

Beispiel SPEC CPU2000

□ Mögliche Kategorien

	Geschwindigkeit	Durchsatz
Agressive Optimierung	SPECint2000	SPECint_rate2000
	SPECfp2000	SPECfp_rate2000
Konservative Optimierung	SPECint_base2000	SPECint_rate_base2000
	SPECfp_base2000	SPECfp_rate_base2000

- Regeln sind genau festgelegt: *SPEC Run and Reporting Rules*
- Referenzmaschine: Sun Ultra10, 300 MHz Ultra-Iii Prozessor

Beispiel SPEC CPU2000

□ Metriken (Geschwindigkeit):

- ◆ Jeder Benchmark hat seine eigene **SPECratio**:
 $\text{xxx.benchmark Referenzzeit} / \text{xxx.benchmark Laufzeit}$
- ◆ **SPECint2000**: geometrisches Mittel der 12 „Integer“ SPECratios (aggressive Optimierung)
- ◆ **SPECint_base2000**: geometrisches Mittel der 12 „Integer“ SPECratios (konservative Optimierung)
- ◆ **SPECfp2000**: geometrisches Mittel der 14 „FP“ SPECratios (aggressive Optimierung)
- ◆ **SPECfp_base2000**: geometrisches Mittel der 14 „FP“ SPECratios (konservative Optimierung)

Beispiel SPEC CPU2000

□ Metriken (Durchsatz):

- ♦ Jeder Benchmark hat seine eigene **SPECrate**:
$$\text{Anzahl der ausgeführten Kopien} * (\text{Referenzzeit von xxx.benchmark} * \text{Anzahl der Sekunden} / \text{längste SPEC CPU2000 Referenzzeit}) / \text{Laufzeit aller xxx.benchmarks}$$
- ♦ **SPECint_rate2000**: geometrisches Mittel der 12 „Integer“ SPECrates (aggressive Optimierung)
- ♦ **SPECint_rate_base2000**: geometrisches Mittel der 12 „Integer“ SPECrates (konservative Optimierung)
- ♦ **SPECfp_rate_2000**: geometrisches Mittel der 14 „FP“ SPECrates (aggressive Optimierung)
- ♦ **SPECfp_rate_base2000**: geometrisches Mittel der 14 „FP“ SPECrates (konservative Optimierung)

Beispiel SPEC CPU2000

□ SPEC CPU 2000 (Base) Messungen

- ◆ Quelle: Microprocessor Report, April 30, 2001, <http://www.mpronline.com>
- ◆ Alpha ES40 Model 6, Alpha 21264B, 833 MHz, 8MB Cache (ext.):
 - ◆ SPECint_base2000: 518
 - ◆ SPECfp_base2000: 590
- ◆ AMD GA-7ZM, AMD Athlon, 1,2 GHz:
 - ◆ SPECint_base2000: 482
 - ◆ SPECfp_base2000: 414
- ◆ HP9000 j6000, HP PA-8600, 552 MHz:
 - ◆ SPECint_base2000: 417
 - ◆ SPECfp_base2000: 400
- ◆ IBM RS /6000 44P-170, IBM Power 3-II, 450 MHz, 8MB Cache (ext.):
 - ◆ SPECint_base2000: 286
 - ◆ SPECfp_base2000: 356

Beispiel SPEC CPU2000

□ SPEC CPU 2000 (Base) Messungen

- ◆ Quelle: Microprocessor Report, April 30, 2001, <http://www.mpronline.com>
- ◆ Dell Prec. 420, Intel PIII, 1 GHz:
 - ◆ SPECint_base2000: 454
 - ◆ SPECfp_base2000: 329
- ◆ Intel D850GB, Intel P4, 1,5 GHz:
 - ◆ SPECint_base2000: 573
 - ◆ SPECfp_base2000: 598
- ◆ Sun Enterprise 450, Sun Ultra-II, 480 MHz, 8MB Cache (ext.):
 - ◆ SPECint_base2000: 225
 - ◆ SPECfp_base2000: 274
- ◆ Sun Blade 1000, Sun Ultra-III, 900 MHz, 8MB Cache (ext.):
 - ◆ SPECint_base2000: 438
 - ◆ SPECfp_base2000: 427

Messungen während des Betriebs

□ Monitore

- ♦ Aufzeichnungselemente, die zum Zweck der Rechnerbewertung die Verkehrsverhältnisse während des normalen Betriebs beobachten und untersuchen.
- ♦ **Hardware-Monitore**
 - ♦ Unabhängige physikalische Geräte
 - ♦ Keine Beeinflussung
- ♦ **Software-Monitor**
 - ♦ Einbau in das Betriebssystem
 - ♦ Beeinträchtigung der normalen Betriebsverhältnisse
- ♦ Aufzeichnungstechniken:
 - ♦ Kontinuierlich oder sporadisch
 - ♦ Gesamtdatenaufzeichnung (Tracing)
 - ♦ Realzeitauswertung
 - ♦ Unabhängiger Auswertungslauf (Post Processing)

Modelltheoretische Verfahren

□ Analytische Methoden

- ◆ Deterministische Warteschlangenmodelle
 - ◆ Systemparameter: feste Werte
- ◆ Stochastische Warteschlangenmodelle
 - ◆ Systemparameter: Mittelwerte
- ◆ Operationelle Warteschlangenmodelle
 - ◆ Systemparameter: gemessene Werte

□ Simulationen

- ◆ Deterministische Simulation
- ◆ Stochastische Simulation
- ◆ Aufzeichnungsgesteuerte Simulation