

---

# Rechnerstrukturen

Vorlesung im Sommersemester 2006

Prof. Dr. Wolfgang Karl

Universität Karlsruhe (TH)

Fakultät für Informatik

Institut für Technische Informatik



## Kapitel 1: Grundlagen

### 1.3 Einführung: Bewertung der Leistungsfähigkeit



- Ziele

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

- **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



- **Standardisierte Benchmarks**
  - Standardisierungsorganisationen
    - EEMBC (Embedded Microprocessor Benchmark Consortium)
      - <http://www.eembc.org>
      - Anwendungen aus dem Bereich Eingebettete Systeme
      - Automotive/Industrial (Automatisierungstechnik)
        - » 6 Microbenchmarks (arithmetische Operationen, Pointer, Speicherleistung, Matrizenarithmetik, Tabellensuche, Bitmanipulation), 5 Steuerungsprogramme aus dem Automobilbereich, 5 Filter und FFT-Benchmarks
      - Consumer (Unterhaltungselektronik)
        - » 5 Multimedia-Benchmarks (JPEG compress/decompress, Filtering, RGB Conversions)
      - Networking (Netzwerk)
        - » Kürzeste-Wege-Berechnung, IP Routing, Paketfluss-Operationen
      - Office Automation
        - » Graphik- und Text-Benchmarks, (Bézier-Kurvenberechnung, Dithering, Bildrotation, Textverarbeitung)
      - Telecommunication
        - » Filtering und DSP-Anwendungen



- Standardisierte Benchmarks
  - 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

- **Verschiedene Untergruppen in SPEC**
  - Open Systems Group (OSG)
    - SPEC CPU 2000: Prozessorleistung
    - 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



- **Verschiedene Untergruppen in SPEC**
  - High Performance Group (HPG)
    - SPEChpc96: Industrielle große Anwendungen
  - Graphics Performance Characterization Group (GPC)
    - Zusammenschluss mit SPECopc: OpenGL Leistungsbewertung



- **SPEC CPU 2000**
  - 12 nichtnumerische Programme in C/C++ (CINT2000)
  - 14 numerische Programme in FORTRAN/C (CFP2000)
  - Strenge, genau festgelegte Regeln
    - Ab CPU95: vollautomatische Messung und Protokollierung
  - Regelmäßige Aktualisierungen (CPU92, CPU95, CPU2000)
    - Laufzeiten werden zu kurz
    - Caches werden größer: größere Datensätze
    - Mehr Praxisnähe: Programme mit schlechterer Datenlokalität

- SPEC CINT2000**

<b>Benchmark</b>	<b>Referenzzeit</b>	<b>Bemerkung</b>
164.gzip	1400	Datenkompression
175.vpr	1400	FPGA Layout und Routing
176.gcc	1100	C Compiler
181.mcf	1800	Lineare Optimierung, KFZ-Scheduling
186.crafty	1000	Schachprogramm
197.parser	1800	Sprachverarbeitung, Parser für Englisch
252.eon	1300	Ray Tracing
253.perlbmk	1800	Perl
256.gap	1100	Gruppentheorie, Interpreter
255.vortex	1900	Datenbank, objektorientiert
256.bzip2	1500	Datenkompression
300.twolf	3000	IC Platzierung und Routing

Referenzzeit bezieht sich auf Sun Ultra10 (300MHz, 256MB)



- SPEC CFP2000

Benchmark	Referenzzeit	Bemerkung
168.wupwize	1600	Quantenphysik
171.swim	3100	Gleichungslösung, Flachwasser-Modellierung
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	Bildererkennung, neuronales Netzwerk Simulation

- SPEC CFP2000

Benchmark	Referenzzeit	Bemerkung
183.equake	1300	Seismografische Wellenausbreitung, Finite Elemente Simulation
187.facerec	1900	Bildverarbeitung, Gesichtserkennung
188.amp	2200	Chemie, Molekulardynamik-Sim.
189.lucas	2000	Zahlentheorie, Mersenne-Primzahltest
191.fma3d	2100	Finite Elemente: Crash-Simulation
200.sixtrack	1100	Teilchenbeschleunigermodell
300.apsi	2600	Sim. Schadstoffausbreitung

- Mögliche Kategorien

	<b>Geschwindigkeit</b>	<b>Durchsatz</b>
<b>Aggressive Optimierung</b>	SPECint2000	SPECint_rate2000
	SPECfp2000	SPECfp_rate2000
<b>Konservative Optimierung</b>	SPECint_base2000	SPECint_rate_base2000
	SPECfp_base2000	SPECfp_rate_base2000

- SPEC CPU Benchmark-Metrik: Geschwindigkeit

$$\text{SPECratio} = \frac{\text{Referenzzeit}_x}{\text{Laufzeit}_x \text{ auf Testsystem}}$$

Benchmark x

- Endwerte: je ein geometrisches Mittel der SPECratio's über alle CINT2000 und CFP2000 Benchmarks
  - SPECint2000, SPECfp2000
    - Aggressive, individuelle Optimierungen erlaubt
  - SPECint\_base2000, SPECfp\_base2000
    - Nur mit konservativer Standardoptimierung
    - Identische Compileroptionen für alle Programme

- SPEC CPU Benchmark-Metrik
  - Warum geometrisches Mittel:

$$\sqrt[n]{\prod_{i=1}^n \text{Execution time ratio}_i}$$

mit *Execution time ratio*<sub>*i*</sub>:  
Ausführungszeit der Programms *i*  
einer Last von *n* Programmen,  
normalisiert bezüglich der  
Referenzmaschine

- Eigenschaft des geometrischen Mittels:

$$\frac{\text{geometrisches Mittel } (X_i)}{\text{geometrisches Mittel } (Y_i)} = \text{geometrisches Mittel} \left( \frac{X_i}{Y_i} \right)$$

- Geometrisches Mittel ist konsistent, unabhängig von Referenzmaschine!





- SPEC CPU Benchmark-Metrik: Durchsatz

$$\text{SPECrate}_x = \frac{\text{Sekunden pro Stunde}}{\text{Laufzeit}_x \text{ von } n_x \text{ Kopien auf Testsystem}} \times \frac{\text{Referenzzeit}_x}{\text{längste Laufzeit}}$$

– Endwerte: je ein geometrisches Mittel der SPECrate's über alle CINT2000 bzw. CFP2000 Benchmarks

- SPECint\_rate2000, SPECfp\_rate2000
- SPECint\_base2000, SPECfp\_base2000
- $n_x$  kann frei gewählt werden, muss aber dokumentiert werden



# Beispiel SPEC CPU2000

Processor	Alpha 21364	AMD Athlon XP	HP PA-8700	IBM Power 4+	Intel Itanium 2	Intel XeonMP	Intel Xeon	MIPS R14000	Sun UltraSPARC III
System or Motherboard	Alpha GS1280/7	ASUS A7N8X	HP9000 C3750	pSeries 650 6M2	HP RX2600	Dell PwrEdg 6650	Dell Prec. 350	SGI 3200	Sun Blade 2050
Clock Rate	1.15GHz	2.17GHz	870MHz	1.45GHz	1.0GHz	2.0GHz	3.06GHz	600MHz	1.05GHz
External Cache	None	None	None	16MB	None	None	None	8MB	8MB
164.gzip	583	1,026	588	673	583	758	1,138	322	433
175.vpr	822	653	688	902	704	625	606	572	460
176.gcc	859	755	906	914	1,014	1,100	1,236	445	577
181.mcf	712	420	494	1,391	834	599	773	783	659
186.crafty	982	1,292	751	884	781	712	1,179	502	558
197.parser	514	905	495	381	660	778	1,025	409	488
252.eon	958	1,483	592	1,150	1,004	920	1,387	507	527
253.perlbmk	768	1,306	619	712	815	952	1,381	367	540
254.gap	636	1,059	339	936	680	722	1,417	308	372
255.vortex	1,094	1,608	1,196	1,428	1,193	1,118	1,658	679	738
256.bzip2	824	840	534	965	759	712	856	493	629
300.twolf	1,018	887	911	1,198	880	1,009	900	645	570
<b>SPECint_base2000</b>	<b>795</b>	<b>960</b>	<b>642</b>	<b>909</b>	<b>810</b>	<b>816</b>	<b>1,085</b>	<b>483</b>	<b>537</b>
168.wupside	883	1,131	446	1,532	1,003	816	1,406	434	659
171.swim	3,590	1,006	931	1,417	3,205	848	1,837	529	980
172.mgrid	708	799	621	850	1,720	449	1,047	379	487
173.applu	1,518	654	702	979	2,033	496	1,168	381	310
177.mesa	928	1,103	694	737	642	814	1,165	425	543
178.galgel	2,105	738	1,603	3,186	2,505	1,200	1,536	1,398	1,713
179.art	2,014	495	670	1,864	4,226	1,147	716	1,436	9,389
183.quake	519	730	413	2,098	1,871	449	1,291	347	645
187.facerec	1,105	1,008	430	1,515	1,152	762	1,315	647	958
188.amp	735	587	553	923	788	729	644	573	509
189.lucas	1,522	853	448	1,306	1,206	682	1,522	442	371
191.fma3d	1,019	850	404	898	747	551	1,089	306	400
200.sixtrack	469	538	471	621	894	376	564	298	366
301.aspi	1,242	705	696	966	678	695	833	406	471
<b>SPECfp_base2000</b>	<b>1,124</b>	<b>776</b>	<b>600</b>	<b>1,221</b>	<b>1,356</b>	<b>677</b>	<b>1,092</b>	<b>499</b>	<b>701</b>

Quelle: MDR, Microprocessor Report, April 28, 2003



- Hennessy/Patterson: A Quantative Approach:  
Kap. 1.5 – 1.9



- **Messung während des Betriebs von Anlagen**

- **Monitore**

- Aufzeichnungselemente, die zum Zweck der Rechnerbewertung die Verkehrsverhältnisse während des normalen Betriebs beobachten und untersuchen.
- **Hardware-Monitore**
  - Unabhängige physikalische Geräte
  - Keine Beeinflussung
- **Software-Monitore**
  - Einbau in das Betriebssystem
  - Beeinträchtigung der normalen Betriebsverhältnisse

- **Monitore**

- Aufzeichnungstechniken:

- Kontinuierlich oder sporadisch
- Gesamtdatenaufzeichnung (Tracing)
- Realzeitauswertung
- Unabhängiger Auswertungslauf (Post Processing)

- **Modelltheoretische Verfahren**

- Unabhängig von der Existenz eines Rechners

- **Modellbildung**

- aufgrund von Annahmen über die Struktur und Betrieb eines Rechners und über die Prozesse
- Darstellung der für die Analyse relevanten Merkmale des Systems:
  - Systemkomponenten
  - Datenverkehr zwischen den Systemkomponenten
- Abstrahierung komplexer Systeme
  - Nur die interessierenden Größen werden erfasst
- Ziel:
  - Aufdecken von Beziehungen zwischen Systemparametern
  - Ermitteln von Leistungsgrößen (Auslastung von Prozessoren und Kanälen, mittlere Antwortzeiten, Warteschlangenlängen, ...)

- **Modellbildung**

- **Analytische Methoden**

- versuchen auf mathematischem Weg, Beziehungen zwischen relevanten Leistungskenngrößen und fundamentalen Systemparametern herzuleiten
    - oft nur minimaler Aufwand, aber dafür weniger aussagekräftig

- **Warteschlangenmodelle**

- Leistungsanalyse von Rechensystemen

- **Petrinetze**

- theoretische Untersuchungen

- **Diagnosegraphen**

- Zuverlässigkeitsanalysen

- **Netzwerkflussmodelle**

- Kapazitätsüberlegungen

- **Modellbildung**

- **Analytische Methoden**

- **Beispiel Warteschlangenmodelle**

- **Deterministische Warteschlangenmodelle**

- » Beispiele für Systemparameter: Rechenzeit, Gerätebedienzeit, Ankunftszeit eines Jobs

- » Feste Werte → deterministische Ergebnisse für die Leistungsgrößen

- **Stochastische Warteschlangenmodelle**

- » Systemparameter statistisch verteilt, mit vorgegebenen Mittelwerten, Verteilungsfunktionen → statistisch verteilte Leistungsgrößen



- Modellbildung

- Analytische Methoden

- Beispiel Warteschlangenmodelle

- Operationelle Warteschlangenmodelle

- » Systemparameter: Gemessene Werte aus der Beobachtung eines Systems in einem festen Zeitintervall
        - » Vereinfachung der Gleichungen für die Bestimmung von Leistungsgrößen
        - » Relativ gute Aussagen über das Leistungsverhalten des Systems bei geeignet gewähltem Zeitintervall

- **Modellbildung**

- **Simulation**

- Vorgänge in einem Rechensystem werden nachgebildet
    - Verwendung üblicher Programmiersprachen oder spezieller Simulationssprachen
    - Verhalten des Simulationsmodells in Bezug auf die relevanten Parameter entspricht weitgehend dem Verhalten des realen Systems
    - Ermittlung der für die Leistungsbewertung interessierenden Größen

- **Modellbildung**

- **Simulation**

- **Deterministische Simulation**

- Alle an dem Modell beteiligten Größen sind exakt definiert oder berechenbar

- **Stochastische Simulation**

- Verwendung von zufallsabhängigen Größen
      - Oft: Einsatz von Zufallsgeneratoren

- **Aufzeichnungsgesteuerte Simulation**

- Verwendung von gemessenen Werten
      - Erfasst an aufgrund der Beobachtung eines Systems in einem festen Zeitintervall

- Modellbildung

- Simulation

- Nachteile:

- Vorbereitung und Ausführung der Simulationsmodelle zeitaufwendig und teuer
      - Planung der Experimente muss sorgfältig durchgeführt werden
      - Auswertung und Interpretation der Ergebnisse nicht immer einfach

- Beispiel:

- SimpleScalar Tool Set (<http://www.simplescalar.com>)
        - » „Standard“-Werkzeug zur Simulation von superskalaren Mikroprozessoren

- **Modellbildung**

- Simulation vs. Analytische Methoden

- Simulation:

- Realistischere Annahmen über das System bei der Simulation
- Berücksichtigung vieler verschiedener Systemgrößen
- Abdeckungen verschiedener Anwendungsbereiche

- Können sich gegenseitig gut ergänzen



- **Zusammenfassung**

- Rechnerleistung abhängig von

- Blickwinkel (Nutzer, Administrator)
- Nutzung des Rechners (Programm)
- System- und Betriebssoftware
- Compilertechnologie
- Befehlssatz
- Mikroarchitektur
- Halbleitertechnologie

- **Zusammenfassung**

- Einfache Maßzahlen sind wenig aussagekräftig

- Beschreiben nur einzelne Aspekte der Leistung
- Nie isoliert betrachten

- Zur Rechnerauswahl / Bewertung von Systemen vor bzw. beim Kauf:

- Benchmarks
  - Problem: Aufwand vs. Repräsentativität
  - De-facto-Standard für CPU-Leistung: CPU2000 Benchmark-Suite

- **Zusammenfassung**

- Bewertung von Systemen im Betrieb

- Monitore: Messung und Auswertung des Verhaltens

- Bewertung von Systemen, die noch nicht existieren

- Analytische Methoden und Simulation
- Problem: Aufwand vs. Genauigkeit



- Übersicht über Rechnerbewertungsverfahren

	Auswertung von Hardwaremaßen und Parametern	Laufzeitmessungen bestehender Programme	Messungen während des Betriebs der Anlagen	Modelltheoretische Verfahren
<b>Rechnerauswahl</b>	Maßzahlen für die Operationsgeschwindigkeit Mixe Kernprogramme	Benchmarks		
<b>Rechner-„Tuning“</b>			Hardware-Monitore Software-Monitore	
<b>Rechnerentwurf</b>				Analytische Methoden Simulation