

# Leistungsmessung

## Leistungsmessung – einfache Leistungsmaße (2/3)

- Floating Point Operations Per Second (FLOPS)
  - Bemerkungen wie bei MIPS
  - häufig für Supercomputer benutztes Maß, z. B.
    - IBM BlueGene/L - eServer, 2005:  
 $2,8 \cdot 10^{14}$  FLOPS = 280 TFLOPS;
    - Grafikkarte ATI Radeon HD 5970: 4,64 TFLOPS
- Cycles per Instruction (CPI)
- Instructions Per Cycle (IPC)
  - Guter Kompromiss, wenn als Durchschnittswert zusammen mit Taktfrequenz genannt.



## Leistungsmessung – einfache Leistungsmaße (1/3)

- Taktfrequenz
  - Übliches Maß für Marketing
  - Keine Aussage über die Bearbeitungsdauer eines einzelnen Befehls
  - Prozessortakt üblicherweise schneller als Bus-/Speichertakt
- Million Instructions Per Second (MIPS)
  - Klassisches Maß; heute aus der Mode gekommen
  - Abhängig vom Befehlssatz
  - Programmabhängig (auch auf der gleichen Maschine)
  - Nicht proportional zur CPU-Zeit

## Leistungsmessung – einfache Leistungsmaße (3/3)

- Leistungsaufnahme
  - z. B. Joule/Instruktion oder umgekehrt:
    - GOps/Watt (P4: 0,1 GOps/W; DSP: 5 GOps/W)
- Weiteres Kriterium könnte auch die **Codedichte** (Anzahl nötiger Instruktionen für eine Anwendung) sein...



## Frames Per Second

- Im Gaming durchgesetzt hat sich das Maß der pro Sekunde dargestellten Bilder (Frames Per Second)
- Einfach messbar mit FRAPS (nur für Windows, [www.fraps.com](http://www.fraps.com))
- Viele Programme können die aktuellen FPS anzeigen



Bild aus First-Person-Shooter „Sauerbraten“, <http://sauerbraten.org/>



## Leistungsmessung (Benchmarks)

- Historisch:
  - Whetstone (1976, Gleitkomma-Berechnungen) und
  - Dhrystone (Reinhold Weicker „Mr. Dhrystone“, 1984, Ganzzahl-Berechnungen)
- Seit 1988: SPEC (Standard Performance Evaluation Corporation)
- TPC (Transaction Processing Performance Council): Datenbanken-Benchmark, transactions per minute
- Problematisch: Neutralität von Messungen und Ergebnisdarstellung sowie Praxisrelevanz der Benchmarks



## Benchmarks: synthetisch vs. Anwendung

Benchmarks für die Bewertung von Gesamtsystemen; in zwei Klassen unterscheidbar:

- **„synthetische“ Benchmarks:** spezielle Tests, die für die Leistungsermittlung programmiert wurden
- **Anwendungs-Benchmarks:** Zusammenstellung verschiedener Real-World-Aufgaben; liefert realistischere Ergebnisse

## Leistungsmessung (Benchmarks)

- Die Werte hängen stets von drei Größen ab:
  - Prozessor (CPU)
  - Speicherarchitektur
  - Compiler (mit Code-Optimierungen)



# SPEC

- Eigenschaften:
  - Mehrere Suiten für verschiedene Zwecke definiert (Integer, Gleitpunkt, Grafik, ...)
  - Strenge Regeln (Run and Reporting Rules) für das Ermitteln und Veröffentlichen konkreter Werte.
  - Die Integer-Benchmark-Suite umfasst z. B. Kompression, C-Compiler, Schachprogramm, objektorientierte Datenbank, Ray tracing, Routenplanung etc. (siehe [www.spec.org/cpu2000/docs/readme1st.html#Q5](http://www.spec.org/cpu2000/docs/readme1st.html#Q5))

# Beispiel: SPEC-CINT 2000

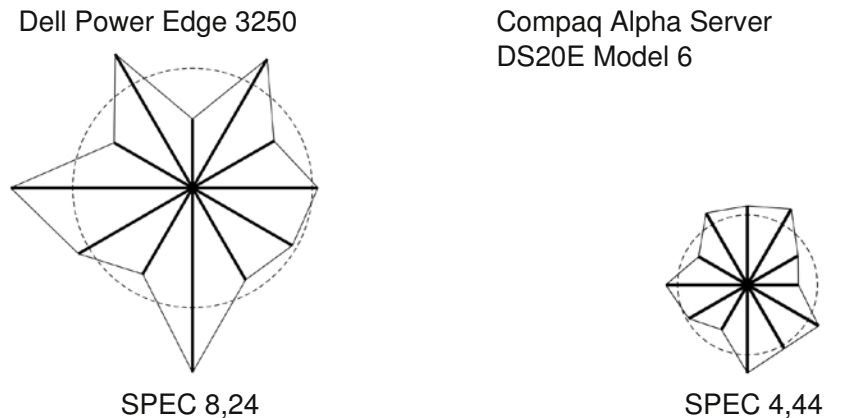
Programm	$r_i$	$b_i$	Verhältnis
164.gzip (Datenkompression)	1400	163	8,6
175.vpr (Schaltungsentwurf)	1400	217	6,45
176.gcc (C-Compiler)	1100	108	10,22
181.mcf (Optimierung)	1800	381	4,72
186.crafty (Schach)	1000	94,2	10,61
197.parser (Textverarbeitung)	1800	290	6,20
252.eon (Visualisierung)	1300	104	12,46
253.perlbnk (PERL Interpreter)	1800	199	9,03
254.gap (Gruppentheorie)	1100	161	6,83
255.vortex (OO-Datenbank)	1900	150	12,65
256.bzips2 (Datenkompression)	1500	206	7,27
300.twolf (Routing Simulation)	3000	380	7,90

## Ermittlung von SPEC-Werten

- Referenzwerte für bestimmten Rechnertyp als Vorgabe (SPEC2006: Sun Ultra Enterprise 2; 296 MHz UltraSPARC II)
- Bestimmung eines Messwerts für ein anderes Rechnersystem: dort alle Programme ausführen und ihre Laufzeiten messen.
- Einzelne Werte in Bezug zu den Referenzwerten setzen und diese Werte geometrisch mitteln: Sind  $r_i$  (für  $i=1...n$ ) die gemessenen Zeiten für  $n$  Programme auf dem Referenzsystem und  $b_i$  die Messwerte auf dem zu bewertenden System, dann ist der SPEC-Wert  $S$ :

$$S = \sqrt[n]{\prod_{i=1}^n \frac{r_i}{b_i}}$$

## Starplots für CINT 2000



# Beispiel: SPEC-CFP 2006

- 410.bwaves (Fluid Dynamics)
- 416.gamess (Quantum Chemistry)
- 433.milc (Physics: Quantum Chromodynamics)
- 434.zeusmp (Physics/CFD)
- 435.gromacs (Biochemistry/Molecular Dynamics)
- 436.cactusADM (Physics/General Relativity)
- 437.leslie3d (Fluid Dynamics)
- 444.namd (Biology/Molecular Dynamics)
- 447.dealII (Finite Element Analysis)
- 450.soplex (Linear Programming, Optimization)
- 453.povray (Image Ray-tracing)
- 454.calculix (Structural Mechanics)
- 459.GemsFDTD (Computational Electromagnetics)
- 465.tonto (Quantum Chemistry)
- 470.ibm (Fluid Dynamics)
- 481.wrf (Weather Prediction)
- 482.sphinx3 (Speech recognition)

# Rolle des Compilers (1/2)

- verschiedene Compiler liefern verschiedene Ergebnisse; z. B. *gcc* vs. *icc* (Intel)
- Programme lassen sich zudem durch optimierende Compiler-Optionen beschleunigen
- Darum gehört genaue Angabe zur eingesetzten Software zum SPEC-Test

	Software
Operating System:	SUSE Linux Enterprise Server 11 (x86_64), Kernel 2.6.27.19-5-default
Compiler:	Intel C++ and Fortran Professional Compiler for IA32 and Intel 64, Version 11.1 Build 20091130 Package ID: L_cproc_p_11.1.064, L_cprof_p_11.1.064
Auto Parallel:	Yes
File System:	ReiserFS
System State:	Run level 3 (multi-user)
<small>Continued on next page</small>	

## SPEC CFP 2006 Ergebnisse

ASUS RS700D-E6 (Z8NH-D12) server system (Intel Xeon X5680)

Benchmark	Base						Peak					
	Seconds	Ratio	Seconds	Ratio	Seconds	Ratio	Seconds	Ratio	Seconds	Ratio	Seconds	Ratio
410.bwaves	75.1	181	<b>74.5</b>	<b>182</b>	74.3	183	74.3	183	<b>74.1</b>	<b>183</b>	74.1	183
416.gamess	769	25.4	764	25.6	<b>765</b>	<b>25.6</b>	677	28.9	<b>678</b>	<b>28.9</b>	678	28.9
433.milc	<b>175</b>	<b>52.5</b>	175	52.5	175	52.4	<b>174</b>	<b>52.9</b>	173	53.0	177	51.7
434.zeusmp	243	37.5	239	38.1	<b>243</b>	<b>37.5</b>	243	37.5	239	38.1	<b>243</b>	<b>37.5</b>
435.gromacs	286	25.0	285	25.0	<b>285</b>	<b>25.0</b>	<b>282</b>	<b>25.3</b>	281	25.4	283	25.2
436.cactusADM	46.8	256	<b>45.0</b>	<b>266</b>	44.8	267	44.6	268	<b>44.4</b>	<b>269</b>	44.4	269
437.leslie3d	278	33.8	<b>247</b>	<b>38.1</b>	246	38.2	278	33.8	<b>247</b>	<b>38.1</b>	246	38.2
444.namd	<b>377</b>	<b>21.3</b>	377	21.3	377	21.3	382	21.0	<b>379</b>	<b>21.2</b>	379	21.2
447.dealII	279	41.0	<b>281</b>	<b>40.7</b>	281	40.7	274	41.8	274	41.7	<b>274</b>	<b>41.7</b>
450.soplex	247	33.8	<b>249</b>	<b>33.5</b>	251	33.2	246	33.9	<b>245</b>	<b>34.0</b>	245	34.1
453.povray	166	32.0	168	31.7	<b>168</b>	<b>31.7</b>	130	40.9	129	41.3	<b>129</b>	<b>41.1</b>
454.calculix	271	30.5	<b>270</b>	<b>30.6</b>	269	30.7	<b>257</b>	<b>32.1</b>	257	32.1	259	31.8
459.GemsFDTD	<b>196</b>	<b>54.0</b>	195	54.4	209	50.8	148	71.8	<b>148</b>	<b>71.7</b>	148	71.5
465.tonto	369	26.7	<b>370</b>	<b>26.6</b>	371	26.6	<b>277</b>	<b>35.6</b>	277	35.6	277	35.5
470.ibm	211	65.2	<b>211</b>	<b>65.2</b>	212	65.0	209	65.7	<b>209</b>	<b>65.7</b>	209	65.6
481.wrf	248	45.0	246	45.5	<b>246</b>	<b>45.4</b>	<b>223</b>	<b>50.0</b>	223	50.0	219	51.1
482.sphinx3	418	46.7	<b>408</b>	<b>47.8</b>	408	47.8	<b>469</b>	<b>41.5</b>	468	41.6	472	41.3

Results appear in the order in which they were run. Bold underlined text indicates a median measurement.

<http://www.spec.org/cpu2006/results/res2010q3/cpu2006-20100705-12176.pdf>

# Rolle des Compilers (2/2)

- Base Optimization vs. Peak Optimization
  - base metrics: einheitliche Compiler-Flags für alle Programme aus dem Test
  - peak metrics: jeden Benchmark von Hand optimieren

Blick auf eine Beispiel-SPEC-Auswertung (cpu2006-20100705-12176.pdf)

# Vorschau 11.11.2010

- Einführung ins Pipelining

