EECS 322
Computer Architecture

Benchmarks

Instructor: Francis G. Wolff
wolff@eecs.cwru.edu
Case Western Reserve University
This presentation uses powerpoint animation: please viewshow

SPEC 2000 FAQ

Reference: http://www.specbench.org/

- **What is SPEC CPU2000?**
- A non-profit group that includes computer vendors, systems integrators, universities and consultants from around the world.
- **What do CINT2000 and CFP2000 measure?**
- Being compute-intensive benchmarks, they measure performance of the
  - **(1) computer's processor,**
  - **(2) memory architecture and**
  - **(3) compiler.**
- It is important to remember the contribution of the latter two components -- performance is more than just the processor.
- **What is not measured?**
- The CINT2000 and CFP2000 benchmarks do not stress: I/O (disk drives), networking or graphics.

SPECint2000 (Number of processors = 1)

Company	System	Clock, CPU	SPEC	L2 cache
Dell	Precision Ws 330	1.50 GHz P4	526	256KB(I+D)
Dell	Precision Ws 330	1.40 GHz P4	505	256KB(I+D)
Intel	VC820	1.13 GHz P3	464	256KB(I+D)
SGI	SGI 2200 2X	400MHz R12k	347	8M(I+D)
Intel	SE440BX-2	800 MHz P3	344	256KB(I+D)
Intel	SE440BX-2	750 MHz P3	330	256KB(I+D)
SGI	Origin200	360MHz R12k	298	4M(I+D)

- Pitfall: Using MIPS or Clock speed as performance metric

Doom benchmark results

Reference: http://www.complang.tuwien.ac.at/misc/doombench.html
Doom,Quake games: http://www.idsoftware.com

"The Doom benchmark is more important than SPEC"

(paraphrased) John Hennessy in his plenary talk at FCRC '99.

avg. fps	Processor	L1 Cache	Mother Board
304.3	MIPS R4400-250	16+16k	SGI Indigo2
201.9	PentiumIIIE-800	16+16K	ASUS P3B-F
197.1	PentiumIIIE-787	16+16K	Abit BH6R1.01
196.0	MIPS R10000-195	32+32k	SGI Indigo2
190.5	PentiumIII-644	16+16K	Abit BX6 2,0
188.1	PentiumIII-800	16+16K	ASUS CU4VX

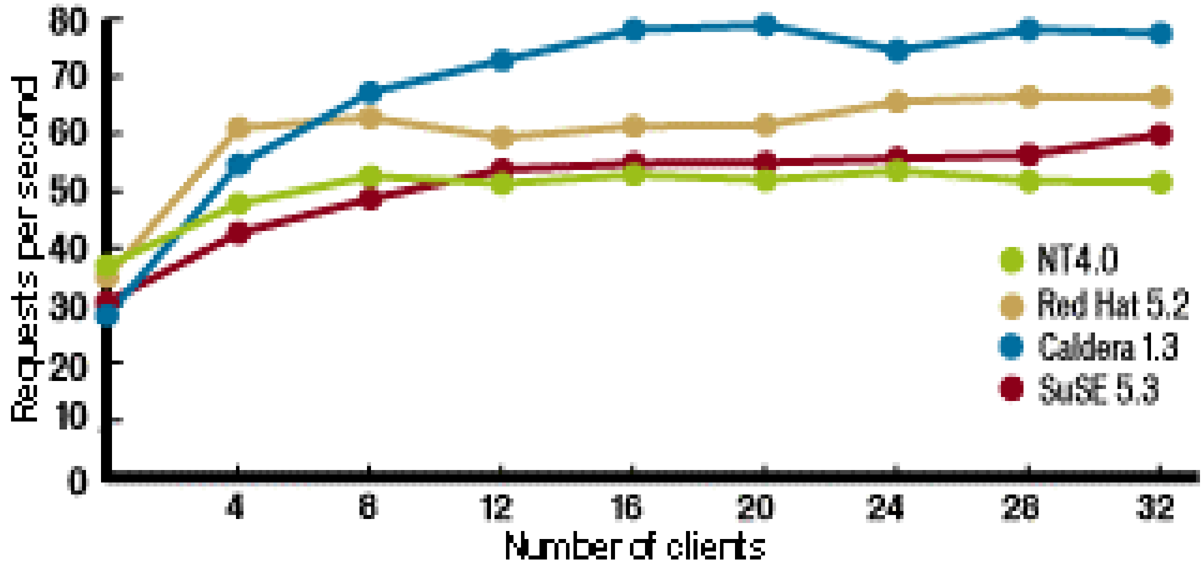
Wow! 250 Mhz MIPS beats the 800 Mhz Pentium.

avg. fps The average number of video frames per second

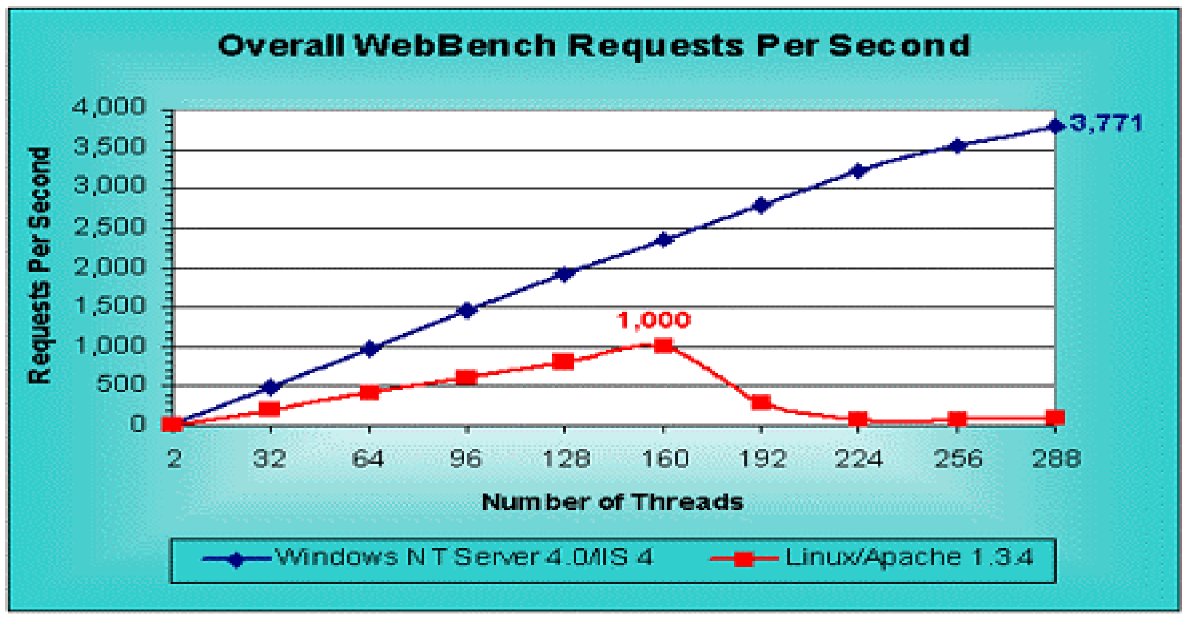
Benchmark wars: Internet Servers

http://www.kegel.com/nt-linux-benchmarks.html

Sm@rt Reseller's January 1999 article, "Linux Is The Web Server's Choice" said "Linux with Apache beats NT 4.0 with IIS, hands down."

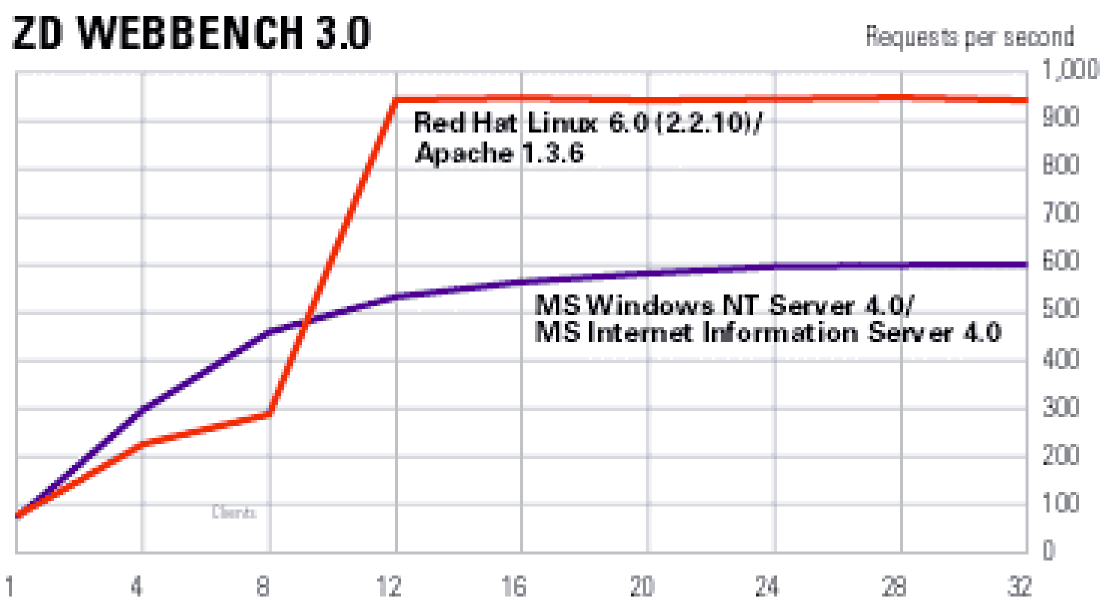


In March 1999, Microsoft *commissioned* Mindcraft to carry out a comparison between NT and Linux.

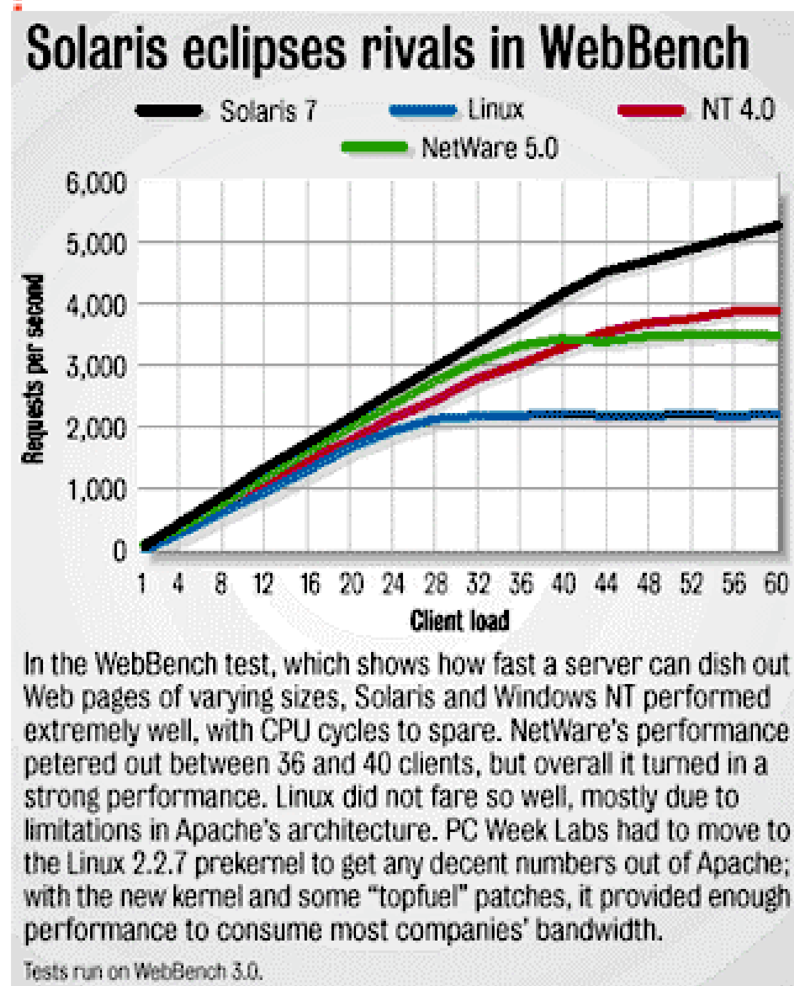


Benchmark Wars: Linux/Solaris

PC Magazine, September 1999



Sun Microsystems SPARC architecture now jumps in!



...found that NT did a lot more disk accesses than Linux, which let Linux score about 50% better than NT.

Performance

To maximize performance,

we want to minimize response time or execution time

$$\text{Performance} = \frac{1}{\text{Execution time}}$$

To compare the relative performance, n,
between machine X and Y, we use

$$\frac{\text{Performance}_X}{\text{Performance}_Y} = \frac{\text{Execution time}_Y}{\text{Execution time}_X} = n$$

Measuring Performance

$$\textbf{Execution time} = \frac{\text{Total program clock cycles executed}}{\text{Clock frequency rate (MHz)}}$$

$$= \frac{\text{Total program instructions exec x CPI}}{\text{Clock frequency rate (MHz)}}$$

CPI = Average number of clock cycles per instruction

$$\textbf{Clock cycle time (us)} = \frac{1}{\text{Clock frequency rate (Mhz)}}$$

CPI Example

Given the following instruction class execution times:

alu=6ns, loads=8ns, stores=7ns, branches=5ns, jumps=2ns

CPI = (6ns+8ns+7ns+5ns+2ns)/5 = 28/5 = 5.6 ns

= (0.2*6ns+0.2*8ns+0.2*7ns+0.2*5ns+0.2*2ns) = 5.6 ns

Given the following instruction class execution times:

alu=60%, loads=20%, stores=10%, branches=5%, jumps=5%

alu=6ns, loads=8ns, stores=7ns, branches=5ns, jumps=2ns

CPI = (0.6*6ns+0.2*8ns+0.1*7ns+0.05*5ns+0.05*2ns) = 6.25

Performance example

(PH page 64)

Benchmark	A	B	L	Total
1	2	1	2	=5
2	4	1	1	=6

Instruction class	CPI
ALU	1
Branches	2
Load/Stores	3

Total CPU cycles$_1$ = (2xA) + (1xB) + (2xL)
= (2x1) + (1x2) + (2x3) = 10 cycles

CPI$_1$ = 10 cycles/5 = 2 average cycles per instruction

Total CPU cycles$_2$ = (4x1) + (1x2) + (1x3) = 9 cycles

CPI$_2$ = 9 cycles/6 = 1.5 average cycles per instruction

- Benchmark 2 executed more instructions, but was faster.

MIPS Performance example

(PH page 78)

Benchmark	A	B	L	Total
Compiler 1	$5x10^9$	10^9	10^9	$=7x10^9$
Compiler 2	10^{10}	10^9	10^9	$=12x10^9$

Instruction class	CPI
ALU	1
Branches	2
Load/Stores	3

Total CPU cycles$_1$ = (5xA) + (1xB) + (1xL) = $10x10^9$ cycles

Execution time$_1$ = $10x10^9$ cycles/500Mhz = 20 seconds

CPI$_1$ = $10x10^9$ cycles/ $7x10^9$ = 1.43

MIPS$_1$ = Clock rate/CPI = 500Mhz/1.43 = 350 MIPS

Total CPU cycles$_2$ = (10xA)+(1xB)+(1xL) = $15x10^9$ cycles

Execution time$_2$ = $15x10^9$ cycles/500Mhz = 30 seconds

CPI$_2$ = $15x10^9$ cycles/$12x10^9$=1.25 MIPS$_2$= 500Mhz/1.25 = 400 MIPS

Although MIPS$_2$ > MIPS$_1$ but execution time is unexpected!

Amdahl's Law (the law of dimishing returns)

Execution Time After Improvement
= Execution Time Unaffected
+ (Execution Time Affected / Amount of Improvement)

Example:

"Suppose a program runs in 100 seconds on a machine, with multiply responsible for 80 seconds of this time.

How much do we have to improve the speed of multiplication if we want the program to run 4 times faster?"

How about making it 5 times faster?

Principle: Make the common case fast
Well, let's speed up the multiply!

Amdahl's Law (the law of dimishing returns)

Execution Time After Improvement =

(Execution Time Affected / Amount of Improvement)

\+ Execution Time Unaffected

Let Execution Time After Improvement be

old time / speed up =

100 seconds / 5 times faster = 20 seconds =

Execution Time needed

= 80 seconds/n + (100-80 seconds)

Equating both sides

20 = 80 seconds/n + (100-80 seconds)

0 = 80 seconds/n

No amount of multiplier speed up can make a 5 fold increase

Sources of improvement

- For a given instruction set architecture,
- increases in CPU performance can come from three sources
  - 1. Increase the clock rate
  - 2. Improve the hardware organization that lower the CPI
  - 3. Compiler enhancements that
    - lower the instruction count or
    - generate instructions with a lower average CPI

- In addition to the above, in order to improve CPU efficiency of software benchmarks.
  - Improve the software organization (data structures, …)

Performance Summary

- **Execution time is the only valid and unimpeachable measure of performance.**

- **Any measure that summarizes performance should reflect execution time.**

- **Designers must balance high-performance with low-cost.**

- **You should not always believe everything you read! Read carefully! (see newspaper articles, e.g., Exercise 2.37)**