ICS 233
COMPUTER ARCHITECTURE

Computer Performance

Lecture 18

Comparing & Summarizing Performance

Consider the Execution times of two programs on two different machines:

	Computer A	Computer B
Program 1 (seconds)	**1**	**10**
Program 2 (seconds)	**1000**	**100**
Total Time (seconds)	**1001**	**110**

Taken individually, each of these statements is true:

- A is 10 times faster than B for Program1
- B is 10 times faster than A for Program2

<u>Total Execution Time : A consistent Summary Measure</u>

The simplest approach to summarizing **relative performance** is to use total execution time of the two programs.

Thus, $\frac{\text{Performance}_B}{\text{Perfromance}_A} = \frac{\text{Execution Time}_A}{\text{Execution Time}_B} = \frac{1001}{110} = 9.1$

That is, B is 9.1 times faster than A for programs 1& 2 together.

Arithmetic Mean (AM)

The average of the execution times that tracks total execution time is the Arithmetic Mean (AM)

$$AM = \frac{1}{n} \sum_{i=1}^{n} Time_i$$

where $Time_i$ is the execution time for the i^{th} program of a total of 'n' in the workload.

Since it is the mean of execution times, a smaller mean indicates a smaller execution time and thus improved performance.

❑ Dawbacks

- All programs are treated equal
- valid if workload consists of equal mix of programs (i.e., running programs equally number of times)

Weighted Arithmetic Mean (WAM)

❑ Weighted Execution Time

- When given an unequal mix of programs in the workload, assign a weighting factor w_i to each program to indicate the relative frequency of the program in that workload.
- By summing the products of weighting factors and execution times, a clear picture of performance of the workload is obtained.
- This is called the Weighted Arithmetic Mean (WAM)

$$WAM = \sum_{i=1}^{n} Weight_i \times Time_i$$

where $Weight_i$ is the frequency of the ith program in the workload and $Time_i$ is the execution time for the i^{th} program.

Weighted Arithmetic Mean (WAM)

Example : **For unequal mix of programs in the workload**

	Programs			Weightings		
	A	B		W(1)	W(2)	W(3)
Program P1(secs)	1	10		0.5	0.909	0.999
Program P2(secs)	1000	100		0.5	0.091	0.001
Arithmetic Mean: W(1)	500.5	55				
Arithmetic Mean: W(2)	91.91	18.19				
Arithmetic Mean: W(3)	2	10.09				

Marketing Metrics

- **MIPS (Million Instructions per Second)**
- **MFLOPs (Million Floating-point operations per Second)**

Peak - maximum able to achieve

Native - average for a set of benchmarks

Relative - compared to another platform

MIPS as a Performance Measure

One alternative to time as the metric is **MIPS (Million Instructions Per Second).**

For a given program, MIPS is simply

$$\text{MIPS} = \frac{\text{Instruction Count}}{\text{Execution Time} \times 10^6}$$

Since MIPS is an instruction execution rate, MIPS specifies performance inversely to execution time; faster machines have a higher MIPS rating.

MIPS

Problems with using MIPS as a measure for comparing machines

There are three problems with using MIPS as a measure for comparing machines

- First, MIPS specifies the instruction execution rate but does not take into account the capabilities of the instructions. We cannot compare computers with different instruction sets using MIPS, since the instruction counts will certainly differ.
- Second, MIPS varies between programs on the same computer; thus a machine cannot have a single MIPS rating for all programs.
- Finally and most importantly, MIPS can vary inversely with performance.

Question 6:

Consider the machine with three instruction classes and CPI measurements:

Instruction Class	CPI for this Instruction Class
A	1
B	2
C	3

Now suppose we measure the code for the same program from two different compilers and obtain the following data:

Code from	Instruction Counts (in billion) for each instruction class		
	A	B	C
Compiler1	5	1	1
Compiler2	10	1	1

Assume that the machines clock rate is 500 MHz. Which code sequence will execute faster according to MIPS? According to execution time?

Answer:

First, we find the execution time for the two different compilers using the following equation:

$$\text{Execution Time} = \frac{\text{CPU Clock cycles}}{\text{Clock rate}}$$

We can use an earlier formula for CPU clock cycles:

$$\text{CPU Clock cycles} = \sum_{i=1}^{n} (CPI_i \times C_i)$$

CPU Clock Cycles$_1$ = (5x1 + 1x2 + 1x3) x 10^9 = 10 x 10^9

CPU Clock Cycles$_2$ = (10x1 + 1x2 + 1x3) x 10^9 = 15 x 10^9

Now, we find the execution time for the two compilers:

$$\text{Execution Time1} = \frac{10 \times 10^9}{500 \times 10^6} = 20 \text{ seconds}$$

$$\text{Execution Time2} = \frac{15 \times 109}{500 \times 106} = 30 \text{ seconds}$$

So we conclude that compiler1 generates the faster program, according to execution time.

Answer (Continued):

Now, let us compute the MIPS rate for each version of the program using:

$$MIPS = \frac{\text{Instruction Count}}{\text{Execution Time} \times 10^6}$$

$$MIPS_1 = \frac{(5+1+1) \times 10^9}{20 \times 10^6} = 350$$

$$MIPS_2 = \frac{(10+1+1) \times 10^9}{30 \times 10^6} = 400$$

So, the code from compiler2 has a higher MIPS rating , but the code from compiler1 runs faster.

As this example shows, MIPS can fail to give a true picture of performance even when comparing two versions of the same program on the same machine.

MFLOPS as Performance Metrics

Another popular alternative to execution time is **million floating point operations per second, abbreviated megaflops or MFLOPS** but always pronounced "megaflops".

The formula for MFLOPS is simply the definition of the acronym:

$$\textbf{MFLOPS} = \frac{\textbf{Number of floating-point operations in a program}}{\textbf{execution Time} \times 10^6}$$

- A floating point operation is an addition, subtraction, multiplication or division operation applied to a number in a single or double precision floating point representation.
- Such data items are heavily used in scientific calculations and are specified in programming languages using keywords like float, real, double or double precision.

Evaluating Performance

❑Using set of Benchmarks

- Benchmarks are programs specifically chosen to measure performance
- Benchmarks form a workload that will predict the performance of the actual workload.
- Best type of programs to use for benchmarks are real applications

  - Engineering & Scientific applications
  - Business applications

Programs to Evaluate Processor Performance

❑Toy Benchmarks

- 10-100 line program
- e.g.: sieve, puzzle, quicksort

❑Synthetic Benchmarks

- Attempt to match average frequencies of real workloads
- e.g., Whetstone, dhrystone

❑Kernel Benchmarks

- Time critical excerpts, computational intense piece of real programs
- e.g., Livermore loops and Linpack

❑Real Benchmarks

- More generally: a collection of programs that possess the characteristics of the real workload
- e.g., SPEC suite of benchmarks

Evaluating Performance

❑ Synthetic Benchmarks

- **Artificial programs that are constructed to try to match the characteristics of a large set of programs**
- **Goal is to create a single benchmark program where the execution frequency of statements in the benchmark matches the statement frequency in a large set of benchmarks.**
- **Most popular synthetic benchmarks are**
  - **- Whetstone**
  - **- Dhrystone**

Evaluating Performance

❑ Synthetic Benchmarks

- **Whetstone**
  - based on measurements of Algol programs in a scientific and engineering environment
  - later converted to Fortran and became popular
- **Dhrystone**
  - created as a benchmark for systems programming environments and was based on a set of published frequency measurements
  - originally written in Ada and later converted to C, after which it became popular.

Evaluating Performance

❑ Kernel Benchmarks

- **Kernels are small, time-intensive pieces from real programs that are extracted and then used as benchmarks**
- **This approach was developed primarily for benchmarking high-end machines, especially supercomputers.**
- **Best known examples are**
  - Livermore loops and Linpack

  Livermore Loops consist of a series of 21 small loop fragments

  Linpack consists of a portion of a linear algebra subroutine package.
- **Because scientific applications often use small pieces of code that execute for a long period of time, characterizing performance with kernels is most popular in this application class.**

Evaluating Performance

❑ SPEC suite of Benchmarks

- **SPEC - System Performance Evaluation Cooperation**
- Provides most popular and comprehensive set of CPU benchmarks
- Created by a set of computer companies (founders being Apollo/Hewlett-Packard, DEC, MIPS and Sun) in 1989 to improve the measurement process and the use of more realistic benchmarks.
- **Example : SPEC benchmark suites**
  SPEC89, SPEC92, SPEC95, SPEC2000, SPEC2006

Evaluating Performance

- The Standard Performance Evaluation Corporation (SPEC) is a non-profit corporation formed to establish, maintain and endorse a standardized set of relevant benchmarks that can be applied to the newest generation of high-performance computers.
- SPEC develops suites of benchmarks and also reviews and publishes submitted results from member organizations and other benchmark licensees.
- Benchmark suites cover the following areas :
  - CPU
  - Graphics/Applications
  - Mail Servers
  - Network File System
  - Web Servers
  - Java Client/Server
  - HPC/OMP (High Performance Computing, Open MP, MPI)

SPEC suite of Benchmarks

- **First Round SPEC CPU89**
  - 10 programs yielding a single number
- **Second Round SPEC CPU92**
  - SPEC CINT92 (6 integer programs) and SPEC CFP92 (14 fp programs)
  - Compiler flags can be set differently for different programs
- **Third Round SPEC CPU95**
  - new set of programs: SPEC CINT95 (8 integer programs) and SPEC CFP95 (10 floating point)
  - Single flag setting for all programs
- **Fourth Round SPEC CPU2000**
  - new set of programs: SPEC CINT2000 (12 integer programs) and SPEC CFP2000 (14 floating point)
  - Single flag setting for all programs; C, C++, Fortran 77, and Fortran 90
  - Report both baseline (same compile options for all) and best (best compile options or each program) performance
- **Fifth Round SPEC CPU2006**
  - formerly CPU2004

SPEC suite of Benchmarks

SPEC2006 benchmark description	SPEC2006	SPEC2000	SPEC95	SPEC92	SPEC89
		Benchmark name by SPEC generation			
GNU C compiler					gcc
Interpreted string processing			perl		espresso
Combinatorial optimization		mcf			li
Block-sorting compression		bzip2		compress	eqntott
Go game (AI)	go	vortex	go	sc	
Video compression	h264avc	gzip	ijpeg		
Games/path finding	astar	eon	m88ksim		
Search gene sequence	hmmer	twolf			
Quantum computer simulation	libquantum	vortex			
Discrete event simulation library	omnetpp	vpr			
Chess game (AI)	sjeng	crafty			
XML parsing	xalancbmk	parser			
CFD/blast waves	bwaves				fpppp
Numerical relativity	cactusADM				tomcatv
Finite element code	calculix				doduc
Differential equation solver framework	dealII				nasa7
Quantum chemistry	gamess				spice
EM solver (freq/time domain)	GemsFDTD			swim	matrix300
Scalable molecular dynamics (~NAMD)	gromacs		apsi	hydro2d	
Lattice Boltzman method (fluid/air flow)	lbm		mgrid	su2cor	
Large eddie simulation/turbulent CFD	LESlie3d	wupwise	applu	wave5	
Lattice quantum chromodynamics	milc	apply	turb3d		
Molecular dynamics	namd	galgel			
Image ray tracing	povray	mesa			
Spare linear algebra	soplex	art			
Speech recognition	sphinx3	equake			
Quantum chemistry/object oriented	tonto	facerec			
Weather research and forecasting	wrf	ammp			
Magneto hydrodynamics (astrophysics)	zeusmp	lucas			
		fma3d			
		sixtrack			

© 2007 Elsevier, Inc. All rights reserved.

SPEC 2000

CINT2000 - 12 integer programs:

- 2 Compression
- 2 Circuit Placement and Routing
- C Programming Language Compiler
- Combinatorial Optimization
- Chess, Word Processing
- Computer Visualization
- PERL Programming Language
- Group Theory Interpreter
- Object-oriented Database.
- Written in C (11) and C++ (1)

CFP 2000 - 14 floating point programs:

- Quantum Physics
- Shallow Water Modeling
- Multi-grid Solver
- 3D Potential Field
- Parabolic / Elliptic PDEs
- 3-D Graphics Library
- Computational Fluid Dynamics
- Image Recognition
- Seismic Wave Simulation
- Image Processing
- Computational Chemistry
- Number Theory / Primality Testing
- Finite-element Crash Simulation
- High Energy Nuclear Physics
- Pollutant Distribution
- Written in Fortran (9) and C (5)

Graphics-intensive benchmarks

SPECviewperf -	used for benchmarking systems supporting the OpenGL graphics Library
SPECapc –	consists of applications that make extensive use of graphics such as
Pro/Engineer–	solid modeling application (3D rendering)
SolidWorks2001-	3D CAD/CAM design tool
Unigraphics V15 –	aircraft design tool

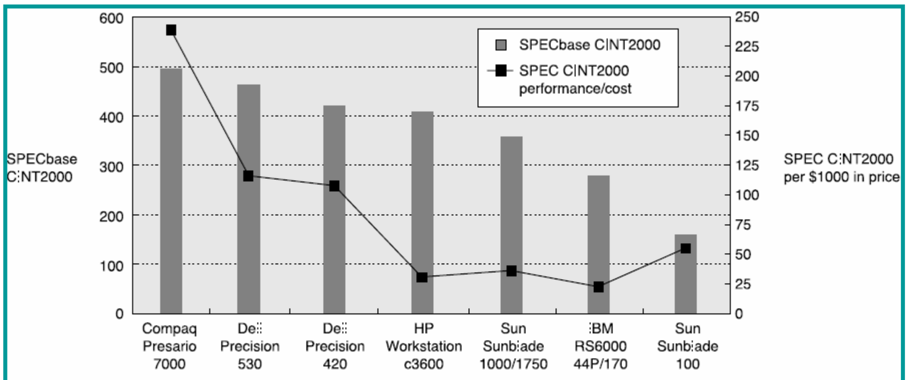
Performance and Price-Performance for Desktop systems

Seven different desktop systems from five vendors using seven different microprocessors showing the processor, its clock rate, and the selling price

Vendor	Model	Processor	Clock Rate MHz	Price
Compaq	Presario 7000	AMD Athlon	1,400	$2091
Dell	Precision 420	Pentium III	1,000	$3,834
Dell	Precision 530	Pentium 4	1,700	$4,175
HP	c3600	PA8600	552	$12,631
IBM	RS6000 44P/170	IBM Power III-2	450	$13,889
Sun	Sunblade 100	UltraSPARC II-e	500	$2,950
Sun	Sunblade 1000	UltraSPARC III	750	$9,950

Performance and Price-Performance for Desktop systems

Performance and price-performance for seven systems are measured using SPEC CINT2000 benchmark

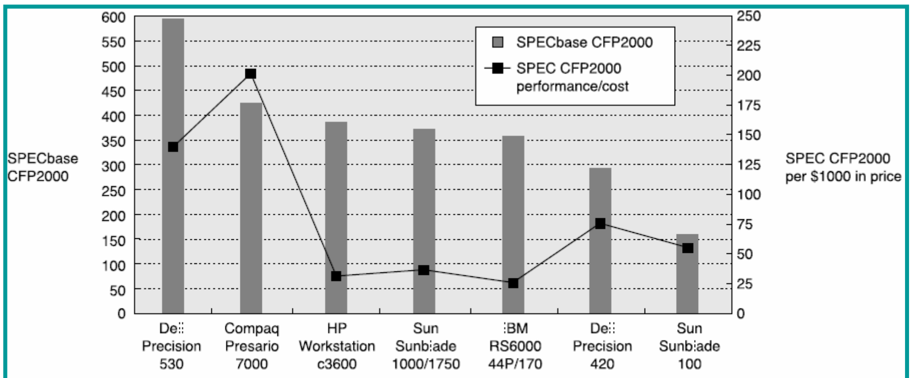


AMD Athlon has higher performance

© 2003 Elsevier Science (USA). All rights reserved.

Performance and Price-Performance for Desktop systems

Performance and price-performance for seven systems are measured using SPEC CFP2000 benchmark



Pentium 4 has higher floating-point performance

© 2003 Elsevier Science (USA). All rights reserved.

SPEC suite of Benchmarks

❑ Server Benchmarks

- **SPECSFS**
  – File server benchmarks for measuring NFS (Network File System) performance. It tests the performance of the I/O system (both disk and network I/O) as well as the CPU.
- **SPECWeb**
  – Web server benchmarks that simulate multiple clients requesting both static and dynamic pages from a server as well as clients posting data to the server
- **TPC**
  - Transaction Processing benchmarks created by Transaction Processing Council (TPC) to measure the ability of a system to handle transactions, which consist of database accesses and updates.
  - many variants depending on transaction complexity
  - TPC-A: simple bank teller transaction style
  - TPC -C: complex database query
  - TPC-H: decision support
  - TPC-R: decision support but with stylized queries (faster than -H)
  - TPC-W: web based transaction

Other SPEC suite of Benchmarks

❑ JAVA
- JVM98 (JVM); jAppServer2004 (J2EE app servers); JBB2005 (typical Java business apps; order processing app for a wholesale supplier)

❑ Network File Systems
- SFS97_R1 (NFS)

❑ WEB
- WEB2005 (World Wide Web applications)

❑ HPC
- HPC2002 (large, industrial applications; OpenMP and MPI)

❑ Graphics
- APC (graphics applications)

❖**For more information about the SPEC benchmarks** see **www.spec.org**

Performance and Price-Performance for Transaction-Processing Servers

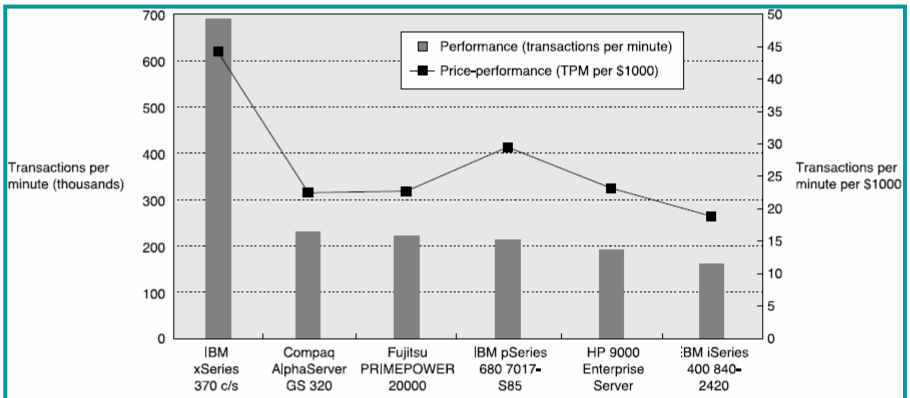
The characteristics of a 12 OLTP systems with first 6 high total performance and next 6 superior price-performance systems

(IBM xSeries with 280 Pentium IIIs is a cluster while all other systems are tightly coupled multiprocessors)

Vendor & System	CPU's	Database	OS	Price
IBM xSeries 370 c/s	280 P3 @ 700MHz	MS SQL	MS Win	$15,543,346
Compaq AlphaSrv GS320	32 2126r @ 1 GHz	Oracle 9i	Tru64 Unix	$10,286,029
Fujitsu PRIMEPOWER 20000	48 SPARC64 @563MHz	SymfoWARE	Solaris 8	$9,671,742
IBM pSeries 680	24 RS64-IV @ 600MHz	Oracle 8v8.1.7.1	AIX 4.3.3	$7,546,837
HP9000 Enterprise Server	48 PA8600 @ 552 MHz	Oracle 8v8.1.7.1	HPUX 11.i 64-bit	$8,522,104
IBM iSeries	24 iSeries400 @ 450MHz	IBM DB/2	OS/400	$8,448,137
Dell PowerEdge 6400	3 PIII's @ 700MHz	MS SQL	Win2000	$131,275
IBMxSeries 250 c/s	4 PIII's @ 700MHz	MS SQL	MS Win adv. srv.	$297,277
Compaq ML570	4 PIII's @ 700MHz	MS SQL	MS Win adv. srv.	$375,016
HP NetServer LH6000	6 PIII's @ 550MHz	MS SQL	NT enterprise	$372,805
NEC Express 5800/180	8 PIII's @ 800MHz	MS SQL	MS Win adv. srv	$682,724
HP9000/L2000	4 PA8500	Sybase	HPUX 11.0 64-bit	$368,367

Performance and Price-Performance for Transaction-Processing Servers

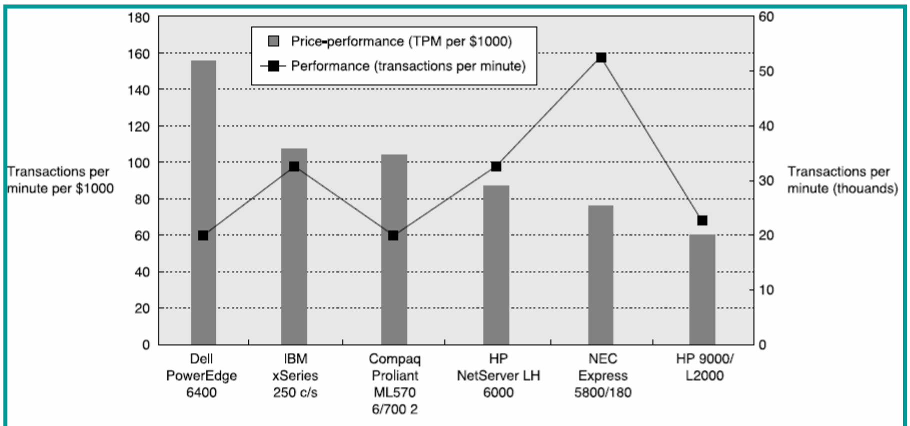
The performance (measured in thousands of transactions per minute) and the price-performance (measured in transactions per minute per $1000) are shown for six of the highest-performing systems using TPC-C benchmark



© 2003 Elsevier Science (USA). All rights reserved.

Performance and Price-Performance for Transaction-Processing Servers

Price-performance (plotted as transactions per minute per $1000 of system cost) and overall performance (plotted as thousands of transactions per minute)



© 2003 Elsevier Science (USA). All rights reserved.

SPEC suite of Benchmarks

❑ Embedded Benchmarks

- **Developed by Embedded Microprocessor Benchmark Consortium (EEMBC, pronounced "embassy") for evaluating kernel performance of embedded applications**
- **EEMBC benchmarks consists of 34 benchmarks falling into five classes :**
  - ➢ **Automotive/industrial (16)** - arithmetic, pointer chasing, table lookup, bit manipulation, etc
  - ➢ **Consumer (5)** - JPEG codec, RGB conversion, filtering
  - ➢ **Networking (3)** - shortest path, IP routing, packet classification
  - ➢ **Office automation (4)** - graphics and text processing
  - ➢ **Telecommunications (6)** - DSP style autocorrelation, FFT, decoder, encoder, FIR filter.

Performance and Price-Performance for Embedded Processors

Five different embedded processors spanning a range of performance (more than a factor of 10) and a wide range in price (roughly a factor of 4 and probably 50% higher than that if total system cost is considered)

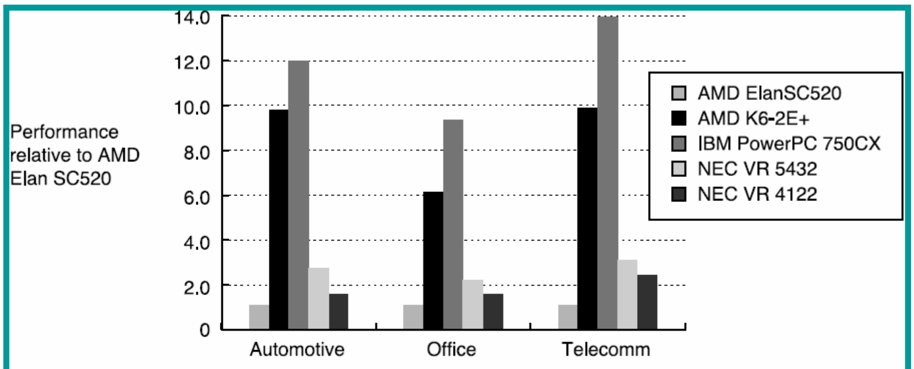
PowerPC 650 and AMD Elan used in applications such as network switches & high-end laptops;

NEC VR5432 series used in color laser printers; NEC VR 4122 used in PDAs.

Processor	Instruction Set	Processor Clock rate (MHz)	Cache instruction/ data On-chip secondary cache	Processor Organization	Typical power (mW)	Price
AMD Elan SC520	x86	133	16K/16K	1 issue pipelined	1600	$38
AMD K6-2E	x86	500	32K/32K 128K	3+ issue pipelined	9600	$78
IBM PPC 750CX	PowerPC	500	32K/32K 128K	4 issue pipelined	6000	$94
NEC VR 5432	MIPS64	167	32K/32K	2 issue pipelined	2088	$25
NEC VR4122	MIPS64	180	32K/16K	1 issue pipelined	700	$33

Performance and Price-Performance for Embedded Processors

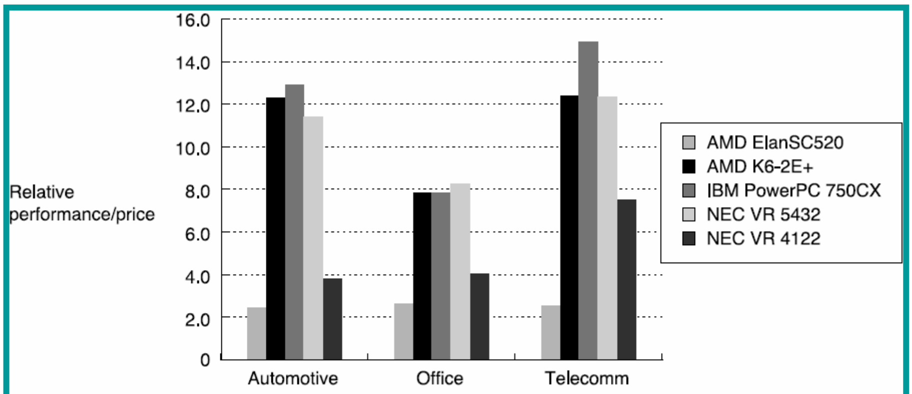
Relative performance of five different embedded processors for three of the five EEMBC benchmark suites



© 2003 Elsevier Science (USA). All rights reserved.

Performance and Price-Performance for Embedded Processors

Relative price-performance of five different embedded processors for three of the five EEMBC benchmark suites, using only the price of the processor

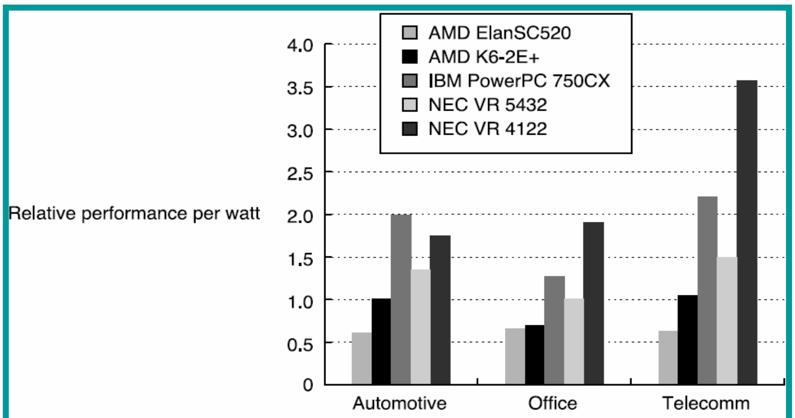


© 2003 Elsevier Science (USA). All rights reserved.

Naseer

Power consumption and Efficiency as the metric for Embedded Processors

Relative performance per watt for the five embedded processors



© 2003 Elsevier Science (USA). All rights reserved.

36

Naseer

Reading Assignments

- **Chapter 4 : Assessing and Understanding Performance**
  Sections 4.1 to 4.5

Refer CD – Ch.4 : In More Depth Section

For more information on SPEC benchmarks suite,
visit website : www.SPEC.org