CSE 2021
Computer Organization

Hugh Chesser, CSEB 1012U

W2-M

From Last Time…

Define the following terms:

- Instruction Set Architecture (ISA)
- Compiler
- DRAM
- Cache memory

Measuring Performance

Agenda:

- Performance Definition
- Performance Metrics: CPU Execution Time and Throughput
- Benchmarks: SPEC 2006
- Alternative Performance Metrics: MIPS and FLOPS

Patterson: Sections 1.4 – 1.10.

Analogy with Commercial Airplanes

What does it mean do say that one computer or airplane is better than another?

Airplane	Passenger Capacity	Cruising range (miles)	Cruising speed (mph)	Passenger throughput (passengers × mph)
Boeing 777	375	4630	610	375 × 610 = 228,750
Boeing 747	470	4150	610	470 × 610 = 286,700
Airbus A3xx	656	8400	600	656 × 600 = 393,600
Concorde	132	4000	1350	132 × 1350 = 178,200
Douglas DC-8-50	146	8720	544	146 × 544 = 79,424

— To know which of the four planes exhibits the best performance, we need to define a criteria for measuring "performance".

Performance Criteria:	**Winner:**
Speed	Concorde
Capacity	Boeing 747
Range	Douglas DC-8-50
Throughput	Airbus A3xx

Computer Performance (1)

— Performance of a computer is based on the following criteria:

1. Execution Time: Elapsed time between the start and the end of one task
2. Throughput: Total number of tasks finished in a given interval of time.

— An IT manager will be interested in having a higher overall throughput while a computer user will like to have a lower execution time for his task.

— Using execution time as the criteria, the performance of a machine *X* is defined as

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

— Performance ratio (n) between two machines *X* and *Y* is defined as

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

Computer Performance (2)

Activity 1: If machine X runs a program in 30 seconds and machine Y runs the same program in 45 seconds, how much faster is X than Y?

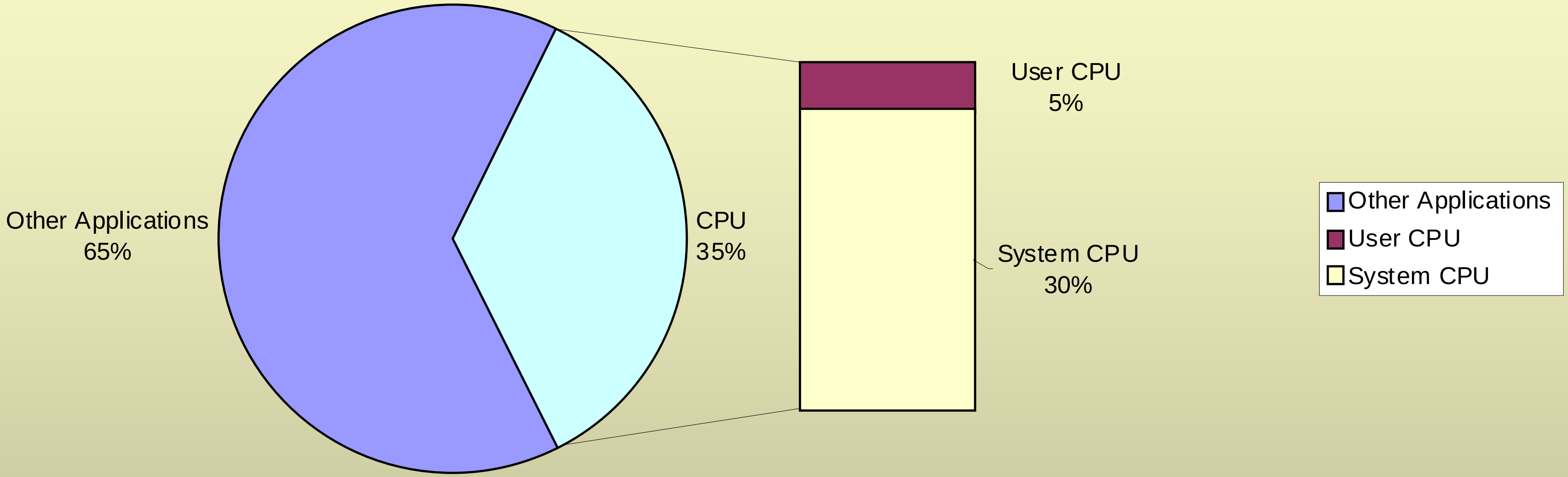
Activity 2: Discuss which of the following two options is better suited to enhance performance from a user's perspective:

(a) Upgrading a machine to a faster CPU

(b) Adding additional processors to the machine so that multiple processors are used for different tasks.

Repeat for an IT manager?

What is Execution Time (1)?



— Command time in Unix can be used to determine the elapsed time and CPU time for a particular program
Syntax: time <name_of_program>
Result:

$$\underbrace{1.180\mathrm{u}}_{\text{CPU user time}} \quad \underbrace{0.130\mathrm{s}}_{\text{CPU system time}} \quad \underbrace{0:0:44.95}_{\text{Elapsed time in h:m:s}} \quad \underbrace{2.9\%}_{\text{CPU time/Elapsed time}} \quad \ldots$$

CPU Performance

— Performance based on User CPU time is called the CPU performance

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

— Performance based on System time is called the system performance

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

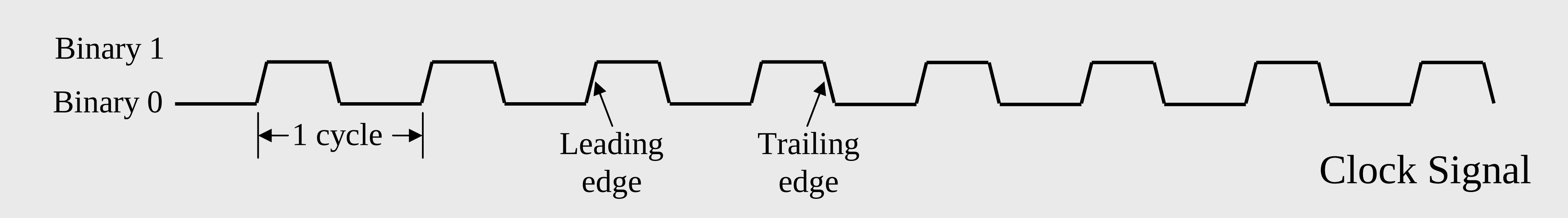
— Vendors specify the speed of a computer in terms of clock cycle time or clock rate.

For example, a 1GHz Pentium is generally faster than a 500MHz Pentium all other factors being the same. We define the clock cycles formally next.

Multiples defined:

$$\begin{array}{lll}
1\text{K} = 2^{10} & \approx 10^{3} & \text{(kilo)} \\
1\text{M} = 2^{20} & \approx 10^{6} & \text{(mega)} \\
1\text{G} = 2^{30} & \approx 10^{9} & \text{(giga)} \\
1\text{T} = 2^{40} & \approx 10^{12} & \text{(tera)} \\
1\text{P} = 2^{50} & \approx 10^{15} & \text{(penta)}
\end{array}$$

Clock (1)



- All events in a computer are synchronized to the clock signal
- Clock signal is therefore received by every HW component in a computer
- Clock cycle time: is defined as the duration of 1 cycle of the clock signal
- Clock cycle rate: is the inverse of the clock cycle time.

- CPU execution time is therefore defined as

$$\left\{\begin{matrix}\text{CPU Execution Time} \\ \text{for a program}\end{matrix}\right\} = \left\{\begin{matrix}\text{CPU Clock Cycles} \\ \text{for a program}\end{matrix}\right\} \times \text{Clock Cycle Time}$$
$$= \frac{\text{CPU Clock Cycles for a program}}{\text{Clock Rate}}$$

Clock (2)

— Timing Programs generally return the average number of clock cycle needed per instruction (denoted by CPI)

$$\begin{Bmatrix}\text{CPU Execution Time} \\ \text{for a program}\end{Bmatrix} = \frac{\text{CPU Clock Cycles for a program}}{\text{Clock Rate}}$$
$$= \frac{\text{Instruction Count in a program} \times \text{CPI}}{\text{Clock Rate}}$$

Activity 3: For a CPU, instructions from a high-level language are classified in 3 classes

Instruction Class	A	B	C
CPI for the instruction class	**1**	**2**	**3**

Two SW implementations with the following instruction counts are being considered

	Instruction counts per instruction class		
	A	**B**	**C**
Implementation 1	**2**	**1**	**2**
Implementation 2	**4**	**1**	**1**

Which implementation executes the higher number of instructions? Which runs faster?

What is the CPI count for each implementation?

Performance Comparison (1)

To compare performance between two computers,

1. Select a set of programs that represent the workload
2. Run these programs on each computer
3. Compare the average execution time of each computer

Activity 4: Based on the average (arithmetic mean) execution time, which of the two computer is faster?

Execution Time	Computer A	Computer B
Program 1	2	10
Program 2	100	105
Program 3	1000	100
Program 4	25	75

Performance Comparison: Benchmarks (2)

- Benchmarks: are standard programs chosen to compare performance between different computers.
- Benchmarks are generally chosen from the applications that a user would typically use the computer to execute.
- Benchmarks can be classified in three categories:
  1. Real applications reflecting the expected workload, e.g., multimedia, computer visualization, database, or macromedia director applications
  2. Small benchmarks are specialized code segments with a mixture of different types of instructions
  3. Benchmark suites containing a standard set of real programs and applications. A commonly used suite is SPEC (System Performance Evaluation Corporation – www.spec.org) with different versions available, e.g., SPEC'89, SPEC'92, SPEC'95, SPEChpc96, and SPEC CPU2006 suites.

Performance Comparison: SPEC Suite (2)

- SPEC'95 suite has a total of 18 programs (integer and floating point) which are called benchmarks. However, SPEC CPU2006 has a total of 29 programs – integer and floating point operations
- SPEC ratio for a program is defined as the ratio of the execution time of the program on a Sun UltraSPARC II (296 MHz processor) to the execution time on the measured machine.
- CINT2006 is the geometric mean of the SPEC ratios obtained from the integer programs. The geometric mean is defined as

$$\mathrm{CINT2000} = \sqrt[n]{\prod_{i=1}^{n} (\mathrm{SPEC\ ratio})_i}$$

- CFP2000 is the geometric mean of the SPEC ratios from the floating-point programs

Activity 5: Complete the following table to predict the performance of machines A and B

	Time on A (seconds)	Time on B (seconds)	Normalized to A		Normalized to B	
			A	B	A	B
Program 1	5	25				
Program 2	125	25				
Arithmetic Mean						
Geometric Mean						

Performance Comparison: SPEC'95 Benchmark (3)

Benchmark	Description
go	Artificial intelligence; plays the game of Go
m88ksim	Motorola 88k chip simulator; runs test program
gcc	The Gnu C compiler generating SPARC code
compress	Compresses and decompresses file in memory
li	Lisp interpreter
ijpeg	Graphic compression and decompression
perl	Manipulates strings and prime numbers in the special-purpose programming language Perl
vortex	A database program
tomcatv	A mesh generation program
swim	Shallow water model with 513 x 513 grid
su2cor	quantum physics; Monte Carlo simulation
hydro2d	Astrophysics; Hydrodynamic Naiver Stokes equations
mgrid	Multigrid solver in 3-D potential field
applu	Parabolic/elliptic partial differential equations
trub3d	Simulates isotropic, homogeneous turbulence in a cube
apsi	Solves problems regarding temperature, wind velocity, and distribution of pollutant
fpppp	Quantum chemistry
wave5	Plasma physics; electromagnetic particle simulation

Performance Comparison: SPEC CPU2006 Benchmarks (4)

Q11. What source code is provided? What exactly makes up these suites?

CINT2006 and CFP2006 are based on compute-intensive applications provided as source code. CINT2006 contains 12 benchmarks: 9 use C, and 3 use C++. The benchmarks are:

400.perlbench	C	PERL Programming Language
401.bzip2	C	Compression
403.gcc	C	C Compiler
429.mcf	C	Combinatorial Optimization
445.gobmk	C	Artificial Intelligence: go
456.hmmer	C	Search Gene Sequence
458.sjeng	C	Artificial Intelligence: chess
462.libquantum	C	Physics: Quantum Computing
464.h264ref	C	Video Compression
471.omnetpp	C++	Discrete Event Simulation
473.astar	C++	Path-finding Algorithms
483.xalancbmk	C++	XML Processing

CFP2006 has 17 benchmarks: 4 use C++, 3 use C, 6 use Fortran, and 4 use a mixture of C and Fortran. The benchmarks are:

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

Improving Performance

$$\left\{\begin{array}{c}\text{CPU Execution Time}\\ \text{for a program}\end{array}\right\} = \frac{\text{Instruction Count in a program} \times \text{CPI}}{\text{Clock Rate}}$$

Performance of a CPU can be improved by:

1. Increasing the clock rate (decreasing the clock cycle time)
2. Enhancements in the Compiler to decrease the instruction count in a program
3. Improvement in the CPU to decrease the clock cycle per instruction (CPI)

Unfortunately factors (1 – 3) are not independent. For example, if you increase the clock frequency then the CPI may also increase.

Performance: Don'ts (1)

1. Do not expect performance of one aspect of a machine to improve overall performance by an amount proportional to the size of improvement (Law of diminishing returns, Amdahl's Law).
2. Do not use MIPS (million instructions per second) as a performance metric

$$\text{MIPS} = \frac{\text{Instruction Count in a program}}{\text{Execution time} \times 10^6}$$

   a. Does not take into account capability of the instruction set – computers with different ISA will have different instructions for a given task.
   b. MIPS will be different for the same machine depending on the program

$$\text{MIPS} = \frac{\text{Instruction Count in a program}}{\text{Execution time} \times 10^6} = \frac{\mathit{Instruction\ Count}}{\left(\dfrac{\mathit{Instruction\ Count} \times \mathit{CPI}}{\mathit{Clock\ Rate}}\right) \times 10^6} = \frac{\mathit{Clock\ Rate}}{\mathit{CPI} \times 10^6}$$

Performance: Don’ts (2)

Activity 6: For a CPU, instructions from a high-level language are classified in 3 classes

Instruction Class	A	B	C
CPI for the instruction class	1	2	3

Two SW implementations with the following instruction counts are being considered

	Instruction counts (in billions) for each instruction class		
	A	B	C
Implementation 1	5	1	1
Implementation 2	10	1	1

Assuming that the clock rate is 500 MHz, calculate the (a) MIPS and (b) execution time.