Lecture 2: Performance Evaluation

Performance definition, benchmark, summarizing performance, Amdahl's law, and CPI

What Does Performance Mean?

- Response time
  - A simulation program finishes in 5 minutes
- Throughput
  - A web server serves 5 million request per second
- Other metrics
  - MIPS (million instruction per second)
  - MFLOPS
  - Clock frequency

Execution Time

- Processor design is concerned with processor consumed by program execution. Shorter execution time=>
  - Shorter response time
  - Higher throughput
- **Execution time = #inst×CPI×Cycletime**
  - What affects #inst, CPI, and cycle time?
  - Almost all designs can be interpreted
- Any other metrics is meaningful only if consistent with execution time

Performance of Computers

Performance is defined for ***a program and a machine.***

How to compare computers? Need benchmark programs:

- Real applications: scientific programs, compilers, text-processing software, image processing
- Modified applications: providing portability and focus
- Kernels: good to isolate performance of individual features
  - Lmbench: measure latency and bandwidth of memory, file system, networking, etc.
- Toy benchmarks
- Synthetic benchmarks: matching average execution profile

Performance Comparison

"X is n times faster than Y":

$$\frac{\text{Performance}_x}{\text{Performance}_y} = \frac{\text{Execution time}_y}{\text{Execution time}_x} = n$$

- *n*: speedup if we are considering an enhancement, optimization, etc.
- What does "improving" mean?
  - Improve performance: decrease execution time, increase throughput
  - Improve execution time: decrease execution time
  - Degrade performance: the reverse of the above; brings negative speedup

Benchmark Suite

- Benchmark suite is a collection of benchmarks with a variety of applications
  - Alleviating weakness of a single benchmark
  - More representative for computer designers to evaluate their design
  - ***Benchmarks test both computer and compilers, and OS in many cases***
- Desktop benchmarks: CPU, memory, and graphics performance
- Sever benchmarks: throughput-oriented, I/O and OS intensive
- Embedded benchmarks: measuring the ability to meet deadline and save power

Summarizing Performance

Given the performance of a set of programs, how to evaluate the performance of machines?

	A	B	C
P1 (secs)	1	10	20
P2 (secs)	1000	100	20
Total (secs)	1001	110	40

- Which computer is the "best" one?

Arithmetic Mean

- Total execution time / (number of programs)

$$\frac{1}{n}\sum_{i=1}^{n} \text{Time}_i$$

  - Simple and intuitive
  - Representative if the user run the programs an equal number of times

Weighted Arithmetic Mean

- Give (different) weights to different programs

$$\sum_{i=1}^{n} \text{Weight}_i \times \text{Time}_i, \qquad \sum_{i=1}^{n} \text{Weight}_i = 1$$

  - Considering the frequencies of programs in the workload

Geometric Means

- Based on relative performance to a reference machine

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

- Relative performance is consistent with different reference machines

$$\frac{\text{Geometric mean}(X_i)}{\text{Geometric mean}(Y_i)} = \text{Geometric mean}(\frac{X_i}{Y_i})$$

  - If C is 2x faster than B (using B as the reference), B is 2x faster than A (A as the reference), then C is 4x faster than A (A as the reference)

Harmonic Mean

- Given speedups s1, s2, …, s_n, the average speedup by harmonic mean is

  n / (1/s1 + 1/s2 + … + 1/s_n)

Why not arithmetic mean?

Amdahl's Law

We know about performance: defining, measuring, and summarizing

How to maximize performance gains from the beginning in our design?

- Principle: Make the Common Case Fast!

Amdahl's Law

- Predict overall speedup from "local speedup" by an enhancement, provided the frequency to use the enhancement is know.
  - "Local speedup" is related to design and optimization objectives, like to double CPU frequency, to reduce cache latency by half

Amdahl's Law

$$\text{Execution time}_{new} = \text{Execution Time}_{old} \times \left((1 - \text{Fraction}_{enhanced}) + \frac{\text{Fraction}_{enhanced}}{\text{Speedup}_{enhance}} \right)$$

$$\text{Speedup}_{overall} = \frac{\text{Execution time}_{old}}{\text{Execution time}_{new}} = \frac{1}{(1 - \text{Fraction}_{enhanced}) + \frac{\text{Fraction}_{enhanced}}{\text{Speedup}_{enhanced}}}$$

Amdahl's Law

Assume we need to improve the performance of a graphics engine

Choice one: Speed up FP Square root by 10x

Choice two: Speed up all FP instruction by 1.6x

Assume 20% inst are FP Square root, 50% for all FP inst

Which choice is better?

Implication: Optimizing for the common case first

Equation Based on Instruction Types

$$\text{CPU time} = \text{CPU Clock Cycles} \times \text{Clock cycle time}$$

$$\text{CPU Clock Cycles} = \left(\sum_{i=1}^{n} IC_i \times CPI_i \right)$$

$$\Rightarrow \text{CPU time} = \left(\sum_{i=1}^{n} IC_i \times CPI_i \right) \times \text{Clock cycle time}$$

$$CPI = \sum_{i=1}^{n} \text{Instruction frequency}_i \times CPI_i$$

Make Design Choice Using CPU Time Equation

	FP	FPSQR	Other
Frequency	25%	2%	75%
CPI	4.0	20	1.33

Alternative 1: CPI_{FPSQR} 20 $\rightarrow$ 2

Alternative 2: CPI_{FP} 4 $\rightarrow$ 2.5

Which one is better? Calculate speedups.

SPEC CPU Benchmark

- SPEC: Standard Performance Evaluation Corporation
- CPU-intensive benchmark for evaluating processor performance of workstation
- Four generations: SPEC89, SPEC92, SPEC95, and SPEC2000
- Two types of programs: INT and FP
- Emphasizing memory system performance in SPEC2000

SPEC CPU2000 Profiling

Dynamic instruction mix

Instruction	Int avg	FP avg
Load int	26%	15%
Store int	10%	2%
Load fp	-	15%
Store fp	-	7%
Add	19%	23%
All fp inst	-	41%
Cond br.	12%	4%
All ctrl inst	16%	4%

Other SPEC Benchmarks

- SPECviewperf and SPEapc: 3D graphics performance
- SPEC JVM98: performance of client-side Java virtual machine
- SPEC JBB2000: Server-cline Java application
- SPEC WEB99: evaluating WWW servers
- SPEC HPC96: parallel and distributed computing

Server Benchmarks

- SPEC CPU2000, WBB99, SFS97
- TPC Measuring the ability of a system to handle transactions
  - TPC-C: online transaction processing (OLTP) benchmark (for bank systems)
  - TPC-H: ad hoc decision make support
  - TPC-R: decision make support with standard queries
  - TPC-W: simulating business-oriented transactional web server

Embedded Benchmark

- EEMBC (Embedded Microprocessor Benchmark Consortium) benchmarks
  - Based on kernel performance
  - Five classes: automotive/industrial, consumer networking, office automation, and telecommunications

Embedded benchmarks are not mature