Computer Organization and Architecture

K. R. Kaza, IIIT Allahabad.

Performance Evaluation

Performance Evaluation

- How to tell which computer is better?

Performance Evaluation

- How to tell which computer design/architecture is better?
- What metrics are appropriate for comparing computers?
-

Performance Evaluation

- How to tell which computer design/architecture is better?
- What metrics are appropriate for comparing computers?
- Are the same metrics useful for all kinds of computers?
-

Performance Evaluation

- How to tell which computer design/architecture is better?
- What metrics are appropriate for comparing computers?
- Are the same metrics useful for all kinds of computers?
- Personal computers ?
- Servers ?

Performance Evaluation

- How to tell which computer design/architecture is better?
- What metrics are appropriate for comparing computers?
- Are the same metrics useful for all kinds of computers?
- Personal computers ?
  - Execution time of a program?

Performance Evaluation

- How to tell which computer design/architecture is better?
- What metrics are appropriate for comparing computers?
- Are the same metrics useful for all kinds of computers?
- Personal computers ?
  - Execution time of a program ?
  - Wall clock time

Performance Evaluation

- How to tell which computer design/architecture is better?
- What metrics are appropriate for comparing computers?
- Are the same metrics useful for all kinds of computers?
- Personal computers ?
  - Execution time of a program ?
  - Wall clock time
- Servers ?

Performance Evaluation

- How to tell which computer design/architecture is better?
- What metrics are appropriate for comparing computers?
- Are the same metrics useful for all kinds of computers?
- Personal computers ?
  - Execution time of a program ?
  - Wall clock time?
- Servers ?
  - Transactions per unit time - Throughput

Performance Evaluation

- Personal computers ?
  - Execution time of a program ?
- Servers ?
- Transactions per unit time - Throughput
- Suppose, I say C_1 is n times faster than C_2.

Performance Evaluation

- Personal computers ?
  - Execution time of a program ?
- Servers ?
- Transactions per unit time - Throughput
- Suppose, I say C_1 is n times faster than C_2. It means,
- Executing time of C_1 : Executing time of C_2 = 1 : n
-

Performance Evaluation

- Personal computers ?
  - Execution time of a program ?
- Servers ?
- Transactions per unit time - Throughput
- Suppose, I say C_1 is n times faster than C_2. It means,
- Executing time of C_1 : Executing time of C_2 = 1 : n
- Throughput of C_1 : Throughput of C_2 = n : 1

Performance Metrics

- Wall-clock time
- Response time
- CPU time
- Throughput
- Speed-up

Performance Metrics

- Wall-clock time - total elapsed physical time a process takes from start to finish
- Response time - time taken for a system to respond to a specific user/system request or input
- CPU time - time for which the CPU actually worked on program execution
  - Excludes disk I/O, network input, etc.
- Throughput - tasks completed per unit time
- Speed-up - relative performance between two systems.

Performance Metrics

- Benchmarks
  - Standard programs used to compare performance of multiple computers
- Benchmark Suites
- Different suites for CPUs, GPUs, Mobile, Embedded, etc.
- Examples of Commonly used Benchmark Suites for CPUs
- SPEC CPU 2017

SPEC Benchmarks

- https://www.spec.org/cpu2017/Docs/overview.html#suites

SPECrate®2017 Integer	SPECspeed®2017 Integer	Language[1]	KLOC[2]	Application Area
500.perlbench_r	600.perlbench_s	C	362	Perl interpreter
502.gcc_r	602.gcc_s	C	1,304	GNU C compiler
505.mcf_r	605.mcf_s	C	3	Route planning
520.omnetpp_r	620.omnetpp_s	C++	134	Discrete Event simulation - computer network
523.xalancbmk_r	623.xalancbmk_s	C++	520	XML to HTML conversion via XSLT
525.x264_r	625.x264_s	C	96	Video compression
531.deepsjeng_r	631.deepsjeng_s	C++	10	Artificial Intelligence: alpha-beta tree search (Chess)
541.leela_r	641.leela_s	C++	21	Artificial Intelligence: Monte Carlo tree search (Go)
548.exchange2_r	648.exchange2_s	Fortran	1	Artificial Intelligence: recursive solution generator (Sudoku)
557.xz_r	657.xz_s	C	33	General data compression

Performance Metrics

- SPEC ratio
- $SPECratio(A) = \dfrac{Execution \quad Time \quad on \quad Reference \quad Machine}{Execution \quad Time \quad on \quad A}$

- The reference machine is a Sun Microsystems server, the Sun Fire V490 with 2100 MHz UltraSPARC-IV+ chips
-

Performance Metrics

- SPEC ratio
- $$SPECratio(A) = \frac{Execution \quad Time \quad on \quad Reference \quad Machine}{Execution \quad Time \quad on \quad A}$$

- The reference machine is a Sun Microsystems server, the Sun Fire V490 with 2100 MHz UltraSPARC-IV+ chips

$$\text{Geometric mean} = \sqrt[n]{\prod_{i=1}^{n} sample_i}$$

-

Performance Metrics

- SPEC ratio
- $$SPECratio(A) = \frac{Execution \quad Time \quad on \quad Reference \quad Machine}{Execution \quad Time \quad on \quad A}$$
- The reference machine is a Sun Microsystems server, the Sun Fire V490 with 2100 MHz UltraSPARC-IV+ chips

$$\text{Geometric mean} = \sqrt[n]{\prod_{i=1}^{n} sample_i}$$

-
- Sample_i is the SPECratio for program i

Performance Metrics

- SPEC ratio
- The ratio of the geometric means is equal to the geometric mean of the per formance ratios.
- This implies that the choice of the reference computer is irrelevant.

Amdahl's Law

- It defines the *speedup* that can be gained by using a particular feature or part of the computer

$$\text{Speedup} = \frac{\text{Execution time for entire task without using the enhancement}}{\text{Execution time for entire task using the enhancement when possible}}$$

-

- Fraction enhanced : The fraction of the computation time in the original computer that can be converted to take advantage of the enhancement

Amdahl's Law

- It defines the *speedup* that can be gained by using a particular feature or part of the computer

- $$\text{Speedup} = \frac{\text{Execution time for entire task without using the enhancement}}{\text{Execution time for entire task using the enhancement when possible}}$$

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

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

Amdahl's Law

- Suppose that we want to enhance the processor used for Web service. The new processor is 10 times faster on computation in the Web service application than the original processor.
- Assuming that the original processor is busy with computation 40% of the time and is waiting for I/O 60% of the time, what is the overall speedup gained by incorporating the enhancement?
-

Amdahl's Law

- Suppose that we want to enhance the processor used for Web service. The new processor is 10 times faster on computation in the Web service application than the original processor.
- Assuming that the original processor is busy with computation 40% of the time and is waiting for I/O 60% of the time, what is the overall speedup gained by incorporating the enhancement?
- $\text{Fraction}_{\text{enhanced}} = 0.4$
- $\text{Speedup}_{\text{enhanced}} = 10$
- $$\text{Speedup}_{\text{overall}} = \frac{1}{0.6 + \frac{0.4}{10}}$$

Processor Performance Equation

$$\text{CPU time} = \text{CPU clock cycles for a program} \times \text{Clock cycle time}$$

$$\text{CPU time} = \frac{\text{CPU clock cycles for a program}}{\text{Clock rate}}$$

-

Processor Performance Equation

$$\text{CPU time} = \text{CPU clock cycles for a program} \times \text{Clock cycle time}$$

$$\text{CPU time} = \frac{\text{CPU clock cycles for a program}}{\text{Clock rate}}$$

-
- The number of clock cycles needed to execute a program
- Depends on the number of **instruction count**

Processor Performance Equation

$$\text{CPU time} = \text{CPU clock cycles for a program} \times \text{Clock cycle time}$$

$$\text{CPU time} = \frac{\text{CPU clock cycles for a program}}{\text{Clock rate}}$$

-
- The number of clock cycles needed to execute a program
- Depends on the number of instruction count
- **Clock cycles per instruction (CPI)**

Processor Performance Equation

$$\text{CPU time} = \text{CPU clock cycles for a program} \times \text{Clock cycle time}$$

- $$\text{CPU time} = \frac{\text{CPU clock cycles for a program}}{\text{Clock rate}}$$
- The number of clock cycles needed to execute a program
- Depends on the number of instruction count
- **Clock cycles per instruction (CPI)**
- $$\text{CPI} = \frac{\text{CPU clock cycles for a program}}{\text{Instruction count}}$$

Processor Performance Equation

- $$\text{CPI} = \frac{\text{CPU clock cycles for a program}}{\text{Instruction count}}$$
- **D**ifficult to change one parameter in complete isolation
- *Clock cycle time*—Hardware technology and computer organization
- .*CPI*—Computer Organization and instruction set architecture
- *Instruction count*—Instruction set architecture and compiler design

Numerical Example 1

- Suppose program runs on C1 (at 4 GHz clock rate) in 10 seconds. You are planning to help design a new computer C2 which can run the program in 6 seconds. You want to increase the clock rate as much as possible, but the designer tells you, it would increase the clock cycles taken (on C2) for your program by 1.2 times that of C1. What clock rate should we specigy for designing C2 ?

Numerical Example 1

- Suppose program runs on C_1 (at 4 GHz clock rate) in 10 seconds. You are planning to help design a new computer C_2 which can run the program in 6 seconds. You want to increase the clock rate as much as possible, but the designer tells you, it would increase the clock cycles taken (on C_2) for your program by 1.2 times that of C_1. What clock rate should we specigy for designing C_2 ?
- CPU time C_1 = CPU clock cycles C_1 / Clock rate C_1
- CPU clock cycles C_1 / 4×10^9 cps = 10 seconds
- CPU clock cycles C_1 = 4×10^{10} cycles for the program

Numerical Example 1

- Suppose program runs on C1 (at 4 GHz clock rate) in 10 seconds. You are planning to help design a new computer C2 which can run the program in 6 seconds. You want to increase the clock rate as much as possible, but the designer tells you, it would increase the clock cycles taken (on C2) for your program by 1.2 times that of C1. What clock rate should we specigy for designing C2 ?
- CPU time C2 = CPU clock cycles C2 / Clock rate C2
- $1.2 \times 4 \times 10^{10}$/ Clock rate C2 = 6 seconds
- clock rate C2 = 8×10^9 cycles per second = 8 GHz

Numerical Example 2

- Suppose there are two implementations for the same ISA. C1 has a clock cycle time of 0.25 nanoseconds and a CPI of 2 for a benchmark program. C2 has a clock cycle time of 0.5 nanoseconds and and a CPI of 1.2 for the same benchmark. Which computer is faster for this benchmark and by how much ?

Numerical Example 2

- Suppose there are two implementations for the same ISA. C1 has a clock cycle time of 0.25 nanoseconds and a CPI of 2 for a benchmark program. C2 has a clock cycle time of 0.5 nanoseconds and and a CPI of 1.2 for the same benchmark. Which computer is faster for this benchmark and by how much ?
- CPU clock cycles = CPI x IC

Numerical Example 2

- Suppose there are two implementations for the same ISA. C1 has a clock cycle time of 0.25 nanoseconds and a CPI of 2 for a benchmark program. C2 has a clock cycle time of 0.5 nanoseconds and and a CPI of 1.2 for the same benchmark. Which computer is faster for this benchmark and by how much ?
- CPU clock cycles = CPI x IC
- CPU clock cycles C1 = 2 IC
- CPU clock cycles C2 = 1.2 IC

Numerical Example 2

- Suppose there are two implementations for the same ISA. C1 has a clock cycle time of 0.25 nanoseconds and a CPI of 2 for a benchmark program. C2 has a clock cycle time of 0.5 nanoseconds and and a CPI of 1.2 for the same benchmark. Which computer is faster for this benchmark and by how much ?
- CPU clock cycles = CPI x IC
- CPU clock cycles C1 = 2 IC
- CPU clock cycles C2 = 1.2 IC
- CPU time C1 = CPU clock cycles C1 x Clock rate C1 = (0.5 IC) nanoseconds
- CPU time C2 = CPU clock cycles C1 x Clock rate C1 = (0.6 IC) nanoseconds

Numerical Example 2

- Suppose there are two implementations for the same ISA. C1 has a clock cycle time of 0.25 nanoseconds and a CPI of 2 for a benchmark program. C2 has a clock cycle time of 0.5 nanoseconds and and a CPI of 1.2 for the same benchmark. Which computer is faster for this benchmark and by how much ?
- CPU time C1 = CPU clock cycles C1 x Clock rate C1 = (0.5 IC) nanoseconds
- CPU time C2 = CPU clock cycles C1 x Clock rate C1 = (0.6 IC) nanoseconds
- CPU time C2/CPU time C1 = 1.2

Processor Performance

- $Execution \quad time = \dfrac{seconds}{program} = \dfrac{Instructions}{program} \times \dfrac{clockcycles}{instruction} \times \dfrac{seconds}{clock \quad cycle}$

Memory Management

Memory Management

Thank you !