Computer Performance Metrics

Course: Computer Architecture and Design

Study Materials: MIPS, MFLOPS, SPEC & Benchmark

Prepared by Academizz

November 28, 2025

Abstract

This document details the fundamental methodologies used to evaluate computer system performance, ranging from basic instruction counts (MIPS) to theoretical speedup limits (Amdahl's Law) and industry-standard benchmarking (SPEC).

Contents

This is an exclusive content of Academizz. To get access to more resources, subscribe to Academizz (app.academizz.com).

1 MIPS (Millions of Instructions Per Second)

1.1 Concept Overview

MIPS is one of the earliest and most straightforward metrics used to gauge the execution speed of a processor. It quantifies performance by calculating how many millions of machine-level instructions a CPU can process in a single second. While useful for general comparisons, students should note that MIPS does not always account for the *complexity* of instructions—some instructions take longer to execute than others.

1.2 The Mathematical Model

To calculate MIPS, we relate the instruction count to the execution time. Alternatively, we can use the Clock Rate and CPI (Cycles Per Instruction).

Key Formulas

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

$$MIPS = \frac{\text{Clock Rate}}{\text{CPI} \times 10^6}$$

Definitions of Variables:

- **Instruction Count (IC):** The total number of instructions contained in the program being executed.
- **Execution Time:** The total wall-clock time required to run the program (measured in seconds).
- **Clock Rate:** The frequency of the processor, measured in Hertz (Hz).
- **CPI (Cycles Per Instruction):** The average number of clock cycles the processor requires to finish a single instruction.

1.3 Illustrative Examples

Example 1: Basic MIPS Calculation

Scenario: A processor executes a specific program containing 500×10^6 instructions. The execution takes 0.25 seconds.
Step-by-Step Solution: We apply the basic MIPS formula using the Instruction Count and Time.

$$MIPS = \frac{500 \times 10^6}{0.25 \times 10^6} = \frac{500}{0.25} = 2000 \text{ MIPS}$$

This is an exclusive content of Academizz. To get access to more resources, subscribe to Academizz (app.academizz.com).

Example 2: Comparing Architectures (Clock Speed vs. Efficiency)

Scenario: We are comparing two processors:

- **Processor A:** 3 GHz Clock Rate, Average CPI of 1.5.
- **Processor B:** 2.5 GHz Clock Rate, Average CPI of 1.0.

Analysis: At a glance, Processor A has a faster clock speed. However, Processor B is more efficient per cycle (lower CPI). We calculate MIPS to see which is actually faster.
Processor A Performance:

$$MIPS_A = \frac{3 \times 10^9}{1.5 \times 10^6} = \frac{3000}{1.5} = 2000 \text{ MIPS}$$

Processor B Performance:

$$MIPS_B = \frac{2.5 \times 10^9}{1.0 \times 10^6} = 2500 \text{ MIPS}$$

Conclusion: Despite having a slower clock speed, **Processor B** is the superior performer (2500 MIPS vs 2000 MIPS) because its instruction execution is more efficient (lower CPI).

This is an exclusive content of Academizz. To get access to more resources, subscribe to Academizz (app.academizz.com).

2 MFLOPS (Millions of Floating-Point Operations Per Second)

2.1 Concept Overview

While MIPS measures all instructions, MFLOPS is a specific metric that focuses solely on floating-point operations—such as addition, subtraction, multiplication, or division of non-integer numbers. This metric is crucial for scientific, engineering, and graphics applications where mathematical precision is more important than general logic.

2.2 The Mathematical Model

MFLOPS Formula

$$MFLOPS = \frac{\text{Floating-Point Operation Count (FLOP)}}{\text{Execution Time} \times 10^6}$$

2.3 Illustrative Examples

Example 1: Scientific Computation

Scenario: A scientific program requires 30×10^6 floating-point operations and takes 0.05 seconds to complete.
Solution:

$$MFLOPS = \frac{30 \times 10^6}{0.05 \times 10^6} = \frac{30}{0.05} = 600 \text{ MFLOPS}$$

Example 2: Mixed Instruction Types

Scenario: A complex program has a total Instruction Count (IC) of 2×10^9. Only 20% of these instructions are floating-point operations. The program runs for 1.5 seconds.
Analysis: We must calculate both the general speed (MIPS) and the floating-point speed (MFLOPS).
1. Calculate MIPS (General Performance):

$$MIPS = \frac{2 \times 10^9}{1.5 \times 10^6} = \frac{2000}{1.5} \approx 1333.33 \text{ MIPS}$$

2. Calculate MFLOPS (Floating-Point Performance): First, determine the count of Floating-Point (FLOP) instructions.

$$FLOP = 20\% \text{ of } 2 \times 10^9 = 0.20 \times 2 \times 10^9 = 400 \times 10^6$$

Now, apply the MFLOPS formula:

$$MFLOPS = \frac{400 \times 10^6}{1.5 \times 10^6} = \frac{400}{1.5} \approx 266.67 \text{ MFLOPS}$$

This is an exclusive content of Academizz. To get access to more resources, subscribe to Academizz (app.academizz.com).

3 Amdahl's Law: The Limits of Optimization

3.1 Concept Overview

Amdahl's Law is a fundamental principle in computer architecture used to predict the maximum theoretical speedup of a system when only a fraction of the system is improved (e.g., via parallelization).

Key Insight: The improvement is always limited by the "serial fraction"—the part of the program that *cannot* be improved. Even if the enhanced part becomes infinitely fast, the serial part remains a bottleneck.

3.2 The Formula

Amdahl's Law

$$Speedup_{overall} = \frac{1}{(1 - Fraction_{enhanced}) + \frac{Fraction_{enhanced}}{Speedup_{enhanced}}}$$

Variables:

- $Fraction_{enhanced}$ (F): The percentage of execution time that can be improved.
- $Speedup_{enhanced}$ (S): How many times faster the enhanced part runs.
- $(1 - F)$: The serial fraction that remains unchanged.

3.3 Illustrative Examples

Example 1: Theoretical Speedup

Scenario: A program spends 80% of its time in a section that can be optimized to run 5 times faster.
Analysis:

- $F = 0.80$
- $S = 5$

$$Speedup_{overall} = \frac{1}{(1 - 0.80) + \frac{0.80}{5}} = \frac{1}{0.20 + 0.16} = \frac{1}{0.36} \approx 2.78$$

Result: The overall program is roughly 2.78 times faster. The speedup is capped by the 20% of the program that was not optimized.

This is an exclusive content of Academizz. To get access to more resources, subscribe to Academizz (app.academizz.com).

Example 2: Execution Time Calculation

Scenario: An application runs in 100 seconds. Floating-point operations take up 40% of this time ($F = 0.40$). We upgrade the hardware to make the floating-point unit 8 times faster ($S = 8$).

Step 1: Calculate New Execution Time (T_{new})

- **Unenhanced Part:** Stays the same.

$$(1 - 0.40) \times 100\ s = 60\ s$$

- **Enhanced Part:** Becomes 8 times faster.

$$\frac{0.40 \times 100\ s}{8} = 5\ s$$

- **Total New Time:** $T_{new} = 60\ s + 5\ s = 65\ s$

Step 2: Calculate Speedup

$$Speedup_{overall} = \frac{T_{old}}{T_{new}} = \frac{100\ s}{65\ s} \approx 1.538$$

This is an exclusive content of Academizz. To get access to more resources, subscribe to Academizz (app.academizz.com).

4 SPEC & Benchmarks

4.1 Understanding Benchmarks

A benchmark is a standard program or set of programs run to assess the relative performance of a computer. It provides a repeatable way to compare different hardware configurations.

Characteristics of a Good Benchmark:

- **Representativeness:** It must mimic real-world usage.
- **Portability:** It must run on different systems easily.
- **Repeatability:** It must give consistent results on the same machine.
- **Measurability:** It must provide clear metrics (Time, MIPS, etc.).
- **Scalability:** It must handle inputs for small PCs and supercomputers alike.

4.2 SPEC (Standard Performance Evaluation Corporation)

SPEC is a non-profit that produces standard benchmarks for various workloads, such as CPU-intensive tasks, graphics, or web servers.

4.3 The SPECratio Metric

The SPECratio compares the tested machine against a standard "Reference Machine."

SPECratio

$$SPECratio = \frac{Execution\ Time_{ReferenceMachine}}{Execution\ Time_{Tested\ Machine}}$$

Note: A higher SPECratio indicates better performance.

4.4 Illustrative Example: SPEC Calculation

Example: Comparing Systems

Scenario:

- **Reference System:** Completes the task in 120 seconds.
- **System A:** Completes the task in 80 seconds.
- **System B:** Completes the task in 150 seconds.

Analysis:
1. System A Performance:

$$SPECratio_A = \frac{120}{80} = 1.5$$

Interpretation: System A is 1.5 times faster than the reference.
2. System B Performance:

$$SPECratio_B = \frac{120}{150} = 0.8$$

Interpretation: System B is 0.8 times the speed of the reference (or 20% slower).

This is an exclusive content of Academizz. To get access to more resources, subscribe to Academizz (app.academizz.com).

5 Additional Practice Problems

Problem 1: Calculating CPI from MIPS

Scenario: A processor has a clock rate of 4 GHz and measures 2000 MIPS on a specific benchmark. What is the average CPI?
Solution: Rearranging the MIPS formula: $MIPS = \frac{Clock\ Rate}{CPI \times 10^6} \Rightarrow CPI = \frac{Clock\ Rate}{MIPS \times 10^6}$

$$CPI = \frac{4 \times 10^9}{2000 \times 10^6} = \frac{4000}{2000} = 2.0$$

Problem 2: Execution Time from Instruction Count

Scenario: A program has 10^9 instructions. If a processor runs at 1000 MIPS, how long will the program take to execute?
Solution:

$$Time = \frac{Instruction\ Count}{MIPS \times 10^6} = \frac{1 \times 10^9}{1000 \times 10^6} = \frac{1000}{1000} = 1.0\ second$$

Problem 3: Simple MFLOPS Calculation

Scenario: A deep learning model performs 50×10^6 floating-point operations in 0.2 seconds. Calculate the MFLOPS.
Solution:

$$MFLOPS = \frac{50 \times 10^6}{0.2 \times 10^6} = 250\ MFLOPS$$

Problem 4: MFLOPS with Instruction Mix

Scenario: A program runs for 0.5 seconds with a total of 1×10^9 instructions. If 40% of these are floating-point operations, what is the MFLOPS rating?
Solution: 1. Find FLOP Count: $0.40 \times 1 \times 10^9 = 400 \times 10^6$ FLOPs.
2. Calculate MFLOPS:

$$MFLOPS = \frac{400 \times 10^6}{0.5 \times 10^6} = 800\ MFLOPS$$

Problem 5: Amdahl's Law - Infinite Parallelism

Scenario: 90% of a program is parallelizable ($F = 0.9$). What is the maximum theoretical speedup if we use an infinite number of processors ($S \to \infty$)?
Solution: As $S \to \infty$, $\frac{F}{S} \to 0$.

$$Speedup = \frac{1}{1 - F} = \frac{1}{1 - 0.9} = \frac{1}{0.1} = 10\times$$

This is an exclusive content of Academizz. To get access to more resources, subscribe to Academizz (app.academizz.com).

Problem 6: Amdahl's Law - Partial Enhancement

Scenario: A server spends 50% of its time on database queries ($F = 0.5$). We optimize the query engine to be 10 times faster ($S = 10$). What is the overall speedup?
Solution:

$$Speedup = \frac{1}{(1 - 0.5) + \frac{0.5}{10}} = \frac{1}{0.5 + 0.05} = \frac{1}{0.55} \approx 1.82\times$$

Problem 7: Comparing Systems via SPEC

Scenario: A reference machine takes 100s to run a benchmark. System A takes 50s. System B takes 200s. Calculate the SPECratio for both.
Solution:

- System A: $SPEC = 100/50 = 2.0$ (Faster)
- System B: $SPEC = 100/200 = 0.5$ (Slower)

Problem 8: Finding Required Enhancement

Scenario: We want an overall speedup of 2.0. If the enhanced part runs 10x faster ($S = 10$), what fraction of the code (F) must be enhanced?
Solution: $2 = \frac{1}{(1-F)+F/10} \Rightarrow 2(1 - 0.9F) = 1 \Rightarrow 2 - 1.8F = 1 \Rightarrow 1.8F = 1 \Rightarrow F \approx 0.556$ (55.6%)

Problem 9: Throughput Calculation

Scenario: A web server handles 15,000 requests in 3 minutes. What is the throughput in requests per second?
Solution: Time = $3 \times 60 = 180$ seconds.

$$Throughput = \frac{15000}{180} \approx 83.33 \text{ requests/sec}$$

Problem 10: Performance Comparison

Scenario: Two CPUs execute the same program with 10^9 instructions.

- CPU A: 2.0 GHz, CPI = 1.5
- CPU B: 2.5 GHz, CPI = 2.0

Which is faster and by how much?
Solution:

- Time A = $\frac{I \times CPI}{Clock} = \frac{10^9 \times 1.5}{2 \times 10^9} = 0.75s$
- Time B = $\frac{I \times CPI}{Clock} = \frac{10^9 \times 2.0}{2.5 \times 10^9} = 0.80s$

CPU A is faster. Speedup = $0.80/0.75 \approx 1.07\times$.

This is an exclusive content of Academizz. To get access to more resources, subscribe to Academizz (app.academizz.com).