COMPUTER ORGANIZATION AND ARCHITECTURE (COA)

EET 2211
4TH SEMESTER – CSE & CSIT
CHAPTER 2, LECTURE 8

CHAPTER 2 – PERFORMANCE ISSUES

TOPICS TO BE COVERED

- Designing for performance
- Multicore, MICs and GPGPUs
- Amdahl’s & Little’s Law
- Basic measures of Computer performance
- Calculating the mean

LEARNING OBJECTIVES

After studying this chapter, you should be able to:

- Understand the key performance issues that relate to computer design.
- Explain the reasons for the move to multicore organization, and understand the trade-off between cache and processor resources on a single chip.
- Distinguish among multicore, MIC and GPGPU organizations.
- Summarize some of the issues in computer performance assessment.
- Explain the differences among arithmetic, harmonic and geometric means.

Overview of Previous Lecture

1. $$\text{CPI} = \frac{\sum_{i=1}^{n}(CPI_i \times I_i)}{I_c}$$

2. $$\text{T} = \text{I}_\text{c} \times \text{CPI} \times \tau$$

3. $$\text{T} = \text{I}_\text{c} \times [\text{p} + (\text{m} \times \text{k})] \times \tau$$

4. $$\text{MIPS rate} = \frac{I_c}{T \times 10^6} = \frac{f}{CPI \times 10^6}$$

5. $$\text{MIPS rate} = \frac{Number\ of\ ececuted\ floating{-}point\ operations\ in\ a\ program}{Execution\ time \times 10^6}$$

$$AM = \frac{1}{n}\sum_{i=1}^{n} R_i = \frac{1}{n}\sum_{i=1}^{n}\frac{Z}{t_i} = \frac{Z}{n}\sum_{i=1}^{n}\frac{1}{t_i}$$

$$GM = \sqrt[n]{x_1 \times \ldots . \times x_n} = \left(\prod_{i=1}^{n} x_i\right)^{1/n} = e^{\left(\frac{1}{n}\sum_{i=1}^{n}\ln(x_i)\right)}$$

$$HM = \frac{n}{\left(\frac{1}{x_1}\right) + \ldots . + \left(\frac{1}{x_n}\right)} = \frac{n}{\sum_{i=1}^{n}\left(\frac{1}{x_i}\right)}$$

$$FM = f^{-1}\left(\frac{f\left(x_1 + \ldots . + f(x_n)\right)}{n}\right) = f^{-1}\left(\frac{1}{n}\sum_{i=1}^{n} f(x_i)\right)$$

Benchmark Principles

Measures such as MIPS and MF LOPS have proven Inadequate to evaluating the performance of processors. Because of differences in instruction sets, the instruction execution rate is not a valid means of comparing the performance of different architectures.

Characteristics of a benchmark program:

1. It is written in a high-level language, making it portable across different machines.

2. It is representative of a particular kind of programming domain or paradigm, such as systems programming, numerical programming, or commercial programming.

3. It can be measured easily.

4. It has wide distribution.

SPEC Benchmarks

- The common need in industry and academic and research communities for generally accepted computer performance measurements has led to the development of standardized
  benchmark suites.

- A benchmark suite is a collection of programs, defined in a high-level language, that together attempt to provide a representative test of a computer in a particular application or system programming area.

- The best known such collection of benchmark suites is defined and maintained by the **Standard Performance Evaluation Corporation (SPEC),** an industry consortium.

Review Questions

2.1 List and briefly discuss the obstacles that arise when clock speed and logic density increase.

2.2 What are the advantages of using a cache?

2.3 Briefly describe some of the methods used to increase processor speed.

2.4 Briefly characterize Amdahl's law.

2.5 Define clock rate. Is it similar to clock speed?

2.6 Define MIPS and FLOPS.

2.7 When is the Harmonic mean an appropriate measure of the value of a system?

2.8 Explain each variable that is related to Little's Law.

PROBLEMS

2.1 What will be the overall speed up if N =10 and f =0.9

Here N =10 f = .9 1-f = 0.1

$$\text{Speedup} = \frac{\text{Time to execute program on a single processor}}{\text{Time to execute program on } N \text{ parallel processors}}$$

$$= \frac{T(1-f) + Tf}{T(1-f) + \dfrac{Tf}{N}} = \frac{1}{(1-f) + \dfrac{f}{N}}$$

Speedup = 100/19 = 5.2632

2.2 What fraction of the execution time involves code that is parallel to achieve an overall speedup of 2.25. Assume 15 numbers of parallel processors?

Here N=15 speedup=2.25

$$\text{Speedup} = \frac{\text{Time to execute program on a single processor}}{\text{Time to execute program on } N \text{ parallel processors}}$$

$$= \frac{T(1-f) + Tf}{T(1-f) + \frac{Tf}{N}} = \frac{1}{(1-f) + \frac{f}{N}}$$

Hence f = 0.59

2.3. A doctor in a hospital observes that on average 6 patients per hour arrive and there are typically 3 patient in the hospital. What is the average range of time each patient spend in the hospital?

Here $\lambda = 6$ and $L = 3$

According to Little's Law i.e. $L = \lambda W$

Therefore, $W = L / \lambda = 0.5$ hrs = 30mins

2.4 Two benchmark programs are executed on three computers with the following results:

	Computer A	Computer B	Computer C
Program 1	50	20	10
Program 2	100	200	40

The table shows the execution time in seconds, with 10,000,000 instructions executed in each of the two programs. Calculate the MIPS values for each computer for each program. Then calculate the arithmetic and harmonic means assuming equal weights for the two programs, and rank the computers based on arithmetic mean and harmonic mean.

MIPS rate:

	Computer A	Computer B	Computer C
Program 1	.2	.5	1
Program 2	.1	.05	.25

Mean calculation

	Computer A	Computer B	Computer C
AM rate	.15	.275	.625
HM rate	.133	.09	0.4

Rank

	Computer A	Computer B	Computer C
AM rate	3rd	2nd	1st
HM rate	2nd	3rd	1st

2.5 Two benchmark programs are executed on three computers with the following result:

a. Compute the arithmetic mean value for each system using X as the reference machine and then using Y as the reference machine. Argue that intuitively the three machines have roughly equivalent performance and that the arithmetic mean gives misleading results.

b. Compute the geometric mean value for each system using X as the reference machine and then using Y as the reference machine. Argue that the results are more realistic than with the arithmetic mean.

Benchmark	Processor		
	X	Y	Z
1	20	10	40
2	40	80	20

Normalized w.r.t X

Benchmark	Processor		
	X	Y	Z
1	1	.5	2
2	1	2	.5
AM	1	1.25	1.25
GM	1	1	1

Normalized w.r.t Y

Benchmark	Processor		
	X	Y	Z
1	2	1	4
2	.5	1	.25
AM	1.25	1	2.125
GM	1	1	1

PRACTICE QUESTIONS:

1. Let a program have 40% of its code enhanced to yield a system speed of 4.3 times faster. What is the factor of improvement?
2. The following table, based on data reported in the literature [HEAT84], shows the execution times, in seconds, for five different benchmark programs on three machines.

a. Compute the speed metric for each processor for each benchmark, normalized to machine R. Then compute the arithmetic mean value for each system.

b. Repeat part (a) using M as the reference machine.

c. Which machine is the slowest based on each of the preceding two calculations?

d. Repeat the calculations of parts (a) and (b) using the geometric mean, Which machine is the slowest based on the two calculations?

Benchmark	Processor		
	R	M	Z
E	417	244	134
F	83	70	70
H	66	153	135
I	39,449	35,527	66,000
K	772	368	369

3. Early examples of CISC and RISC design are the VAX 11/780 and the IBM RS/6000, respectively. Using a typical benchmark program, the following machine characteristics result:

Processor	Clock Frequency (MHz)	Performance (MIPS)	CPU Time (secs)
VAX 11/780	5	1	12 x
IBM RS/6000	25	18	x

The final column shows that the VAX required 12 times longer than the IBM measured in CPU time.

a. What is the relative size of the instruction count of the machine code for this benchmark program running on the two machines?

b. What are the *CPI values for the two machines?*

4. A benchmark program is run first on a 200 MHz. The executed program consists of 1000,000 instruction executions, with the following instruction mix and clock cycle count:

Instruction Type	Instruction Count	Cycles per Instruction
Integer arithmetic	400000	1
Data transfer	350000	2
Floating point	200000	3
Control transfer	50000	2

Determine the effective CPI and MIPS rate.

THANK YOU