COMPUTER ORGANIZATION AND ARCHITECTURE (COA)

EET 2211
4TH SEMESTER – CSE & CSIT
CHAPTER 2, LECTURE 7

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

CALCULATING THE MEAN

- In evaluating some aspect of computer system performance, it is often the case that a single number, such as execution time or memory consumed, is used to characterize performance and to compare systems.

- Especially in the field of benchmarking, single numbers are typically used for performance comparison and this involves calculating the mean value of a set of data points related to execution time.

- It turns out that there are multiple alternative algorithms that can be used for calculating a mean value, and this has been the source of controversy in the benchmarking field.

- In this section, we define these alternative algorithms and comment on some of their properties.

- The three common formulas used for calculating a mean are:

Arithmetic Mean

Geometric Mean

Harmonic Mean

❖ ***Given a set of n real numbers ($x_1, x_2, \ldots, x_n$), the three means are defined as follows:***

1. Arithmetic Mean

- An AM is an appropriate measure if the sum of all the measurements is a meaningful and interesting value. The AM is a good candidate for comparing the execution time performance of several systems.
- The AM used for a time-based variable (e.g., seconds), such as program execution time, has the important property that it is directly proportional to the total time.

$$AM = \frac{x_1 + \ldots + x_n}{n} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

- We can conclude that the AM execution rate is proportional to the sum of the inverse execution time.

2. Harmonic Mean

- For some situations, a system's execution rate may be viewed as a more useful measure of the value of the system. This could be either the instruction execution rate, measured in MIPS or MFLOPS, or a program execution rate, which measures the rate at which a given type of program can be executed.

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

- The HM is inversely proportional to the total execution time, which is the desired property.

- Let us look at a basic example and first examine how the AM performs. Suppose we have a set of n benchmark programs and record the execution times of each program on a given system as t_1 , t_2, …, t_n.

- For simplicity, let us assume that each program executes the same number of operations Z; we could weight the individual programs and calculate accordingly but this would not change the conclusion of our argument.

- The execution rate for each individual program is $R_i = Z / t_i$. We use the AM to calculate the average execution rate.

- If we use the AM to calculate the average execution rate.

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

We see that the AM execution rate is proportional to the sum of the inverse execution times, which is not the same as being inversely proportional to the sum of the execution times. Thus, the AM does not have the desired property.

- The HM yields the following result

$$HM = \frac{n}{\sum_{i=1}^{n} \left(\frac{1}{R_i}\right)} = \frac{n}{\sum_{i=1}^{n} \left(\frac{1}{Z/t_i}\right)} = \frac{nZ}{\sum_{i=1}^{n} t_i}$$

The HM is inversely proportional to the total execution time, which is the desired property.

- A simple numerical example will illustrate the difference between the two means in calculating a mean value of the rates, shown in Table below. The table compares the performance of three computers on the execution of two programs. For simplicity, we assume that the execution of each program results in the execution of 10^8 floating-point operations.

- The left half of the table shows the execution times for computer running each program, the total execution time and the AM of the execution times. Computer A executes in less total time than B, which executes in less total time than C, and this is also reflected by the AM.

- The right half of the table shows a comparison in terms of MFLOPS rate.

Table2.1 A Comparison of Arithmetic and Harmonic Means for Rates

	Computer A time (secs)	Computer B time (secs)	Computer C time (secs)	Computer A rate (MFLOPS)	Computer B rate (MFLOPS)	Computer C rate (MFLOPS)
Program 1 (10^8 FP ops)	2.0	1.0	0.75	50	100	133.33
Program 2 (10^8 FP ops)	0.75	2.0	4.0	133.33	50	25
Total execution time	2.75	3.0	4.75	—	—	—
Arithmetic mean of times	1.38	1.5	2.38	—	—	—
Inverse of total execution time (1/sec)	0.36	0.33	0.21	—	—	—
Arithmetic mean of rates	—	—	—	91.67	75.00	79.17
Harmonic mean of rates	—	—	—	72.72	66.67	42.11

- ✓ The greatest value of AM is for computer A, which means computer A is the fastest computer. B is also slower than C, where as B is faster than C.
- ✓ In terms of total execution time, A has minimum time, so it is the fastest computer out of the three.
- ✓ The HM values correctly reflect the speed ordering of the computers. This confirms that the HM is preferred when calculating the rates.

- There are two reasons for doing the individual calculations rather than only looking at the aggregate numbers:

❶ A customer or researcher may be interested not only in the overall average performance but also performance against different types of benchmark programs, such as business applications, scientific modelling, multimedia applications and system programs.

❷ Usually, the different programs used for evaluation are weighted differently. In Table 2.1 it is assumed that the two test programs execute the same number of operations. If that is not the case, we may want to weight accordingly. Or different programs could be weighted differently to reflect importance or priority.

- Let us see what the result is if test programs are weighted proportional to the number of operations. *The* weighted HM is therefore:

$$WHM = \frac{1}{\sum_{i=1}^{n}\left(\left(\frac{Z_i}{\sum_{j=1}^{n} Z_j}\right)\left(\frac{1}{R_i}\right)\right)} = \frac{n}{\sum_{i=1}^{n}\left(\left(\frac{Z_i}{\sum_{j=1}^{n} Z_j}\right)\left(\frac{t_i}{Z_i}\right)\right)} = \frac{\sum_{j=1}^{n} Z_j}{\sum_{i=1}^{n} t_i}$$

- We can see that the weighted HM is the quotient of the sum of the operation count divided by the sum of the execution times.

3. Geometric Mean

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

- Here we note that

i. with respect to changes in values, the GM gives equal weight to all of the values in the data set

ii. and the GM of the ratios equals the ratio of the GMs (equation is given below)

$$GM = \left(\prod_{i=1}^{n} \frac{z_i}{t_i} \right)^{1/n} = \frac{\left(\prod_{i=1}^{n} z_i \right)^{1/n}}{\left(\prod_{i=1}^{n} t_i \right)^{1/n}}$$

- For use with execution times, as opposed to rates, one **drawback** of the GM is that it may be non-monotonic relative to the AM.

- One property of the GM that has made it appealing for benchmark analysis is that it provides consistent results when measuring the relative performance of machines.

- This is in fact what benchmarks are used for i.e. to compare one machine with another in terms of performance metrics. The results are expressed in terms of normalized values to a reference machine.

- A simple example will illustrate the way in which the GM exhibits consistency for normalized results. In Table 2.2, we use the same performance results as were used in Table 2.1.

Table 2.2 A Comparison of Arithmetic and Geometric Means for Normalized Results

(a) Results normalized to Computer A

	Computer A time	Computer B time	Computer C time
Program 1	2.0 (1.0)	1.0 (0.5)	0.75 (0.38)
Program 2	0.75 (1.0)	2.0 (2.67)	4.0 (5.33)
Total execution time	2.75	3.0	4.75
Arithmetic mean of normalized times	1.00	1.58	2.85
Geometric mean of normalized times	1.00	1.15	1.41

(b) Results normalized to Computer B

	Computer A time	Computer B time	Computer C time
Program 1	2.0 (2.0)	1.0 (1.0)	0.75 (0.75)
Program 2	0.75 (0.38)	2.0 (1.0)	4.0 (2.0)
Total execution time	2.75	3.0	4.75
Arithmetic mean of normalized times	1.19	1.00	1.38
Geometric mean of normalized times	0.87	1.00	1.22

Table 2.3 Another Comparison of Arithmetic and Geometric Means for Normalized Results

(a) Results normalized to Computer A

	Computer A time	Computer B time	Computer C time
Program 1	2.0 (1.0)	1.0 (0.5)	0.20 (0.1)
Program 2	0.4 (1.0)	2.0 (5.0)	4.0 (10.0)
Total execution time	2.4	3.00	4.2
Arithmetic mean of normalized times	1.00	2.75	5.05
Geometric mean of normalized times	1.00	1.58	1.00

(b) Results normalized to Computer B

	Computer A time	Computer B time	Computer C time
Program 1	2.0 (2.0)	1.0 (1.0)	0.20 (0.2)
Program 2	0.4 (0.2)	2.0 (1.0)	4.0 (2.0)
Total execution time	2.4	3.00	4.2
Arithmetic mean of normalized times	1.10	1.00	1.10
Geometric mean of normalized times	0.63	1.00	0.63

Why to choose GM?

1. As mentioned, the GM gives consistent results regardless of which system is used as a reference. Because benchmarking is primarily a comparison analysis, this is an important feature.

2. The GM is less biased by outliers than the HM or AM.

3. Distributions of performance ratios are better modelled by lognormal distributions than by normal ones, because of the generally skewed distribution of the normalized numbers. The GM can be described as the back-transformed average of a lognormal distribution.

- It can be shown that the following inequality holds:

$$AM \geq GM \geq HM$$

The values are equal only if $x_1 = x_2 = \ldots . x_n$.

- We can get a useful insight into these alternative calculations by defining the **Functional mean (FM)**.

- Let f(x) be a continuous monotonic function defined in the interval $0 \leq y < \infty$…. The functional mean with respect to the function f(x) for n positive real numbers ($x_1, x_2, \ldots, x_n$) is defined as:

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

where $f^{-1}(x)$ is the inverse of $f(x)$.

- The mean values are also special cases of the functional mean as defined as follows:

i. AM is the FM with respect to $f(x) = x$

ii. GM is the FM with respect to $f(x) = \ln x$

iii. HM is the FM with respect to $f(x) = 1/x$

REVIEW QUESTIONS

1. List and briefly discuss the obstacles that arise when clock speed and logic density increases.
2. What are the advantages of using a cache?
3. Briefly describe some of the methods used to increase processor speed.
4. Briefly characterize Little's law.
5. How can we determine the speed of a processor?
6. With respect to the system clock define the terms of clock rate, clock cycle and cycle time.
7. Define MIPS and MFLOPS.
8. When is harmonic mean an appropriate measure of the value of a system?
9. Explain each variable that is related to Little's law.

THANK YOU