|  |  |  |  |
| --- | --- | --- | --- |
| Instructor | ***Luke Papademas*** | Due Date | **7/21** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Part | **1** | **2** | **3** | **4** | Total |
| *Maximum Points* | **25** points | **25** points | **25** points | **25** points | **100**G101010 pointsG |
| ***Your Score*** |  |  |  |  |  |

**Textbook Reading Assignment**

Thoroughly read Chapter(s) 11 in your Computer Architecture and Organization textbook.

**Part 1 Glossary Terms - Performance Measurement and Analysis**

Define, in detail, each of these glossary terms from the realm of computer architecture and computer topics, in general. If applicable, use examples to support your definitions. Consult your notes

or course textbook(s) as references or the Internet by visiting Web sites such as:

[**http://www.ask.com**](http://www.ask.com) or [**http://www.webopedia.com**](http://www.webopedia.com/)

**(a) Delayed branching**

|  |
| --- |
| Delayed branching occurs when one or more instructions after the conditional branch are executed regardless of the outcome of the branching. This is to utilize otherwise wasted cycles following a branch. |

**(b) Elevator Algorithm**

|  |
| --- |
| The elevator algorithm is a method to avoid he starvation risk of SSTF. It involves the disk arm continually sweeping over the surface of the disk, stopping when it reaches a track for which it has a request in its service queue. |

**(c) FLOPS**

|  |
| --- |
| FLOPS stands for floating point operations per seconds. It is an outdated measure of computer performance for which there is no clear definition. |

**(d) Loop Peeling**

|  |
| --- |
| Loop peeling is a type of loop fission that involves removing the beginning or ending statements from a loop. |

**(e) SSTF**

|  |
| --- |
| SSTF stands for shortest seek time first. It is a disk scheduling algorithm that arranges access request so that the disk arm services the track nearest its current location. |

**Part 2 Exercises - Performance Measurement and Analysis**

For each of the following, enter True or False.

\_\_T\_\_\_ **(1)** Mathematical and statistical tools give us many ways to rate the overall performance of a system and its components.

\_\_T\_\_\_ **(2)** A CPU running at double the clock speed of another is likely to give better CPU throughput.

\_\_T\_\_\_ **(3)** Branch prediction is the process of attempting to guess the next instruction in the instruction stream.

\_\_T\_\_\_ **(4)** Disk scheduling can be a function of either the disk controller or the host operating system.

\_\_T\_\_\_ **(5)** The elevator algorithm works much similar to how skyscraper elevators service their passengers.

\_\_T\_\_\_ **(6)** When throughput is more important than reliability, a system may employ the write - back cache policy.

\_\_T\_\_\_ **(7)** More time - consuming programs have greater influence on the harmonic mean.

\_\_T\_\_\_ **(8)** Clock speed, MIPS and FLOPS are the metrics in comparing relative performance across a line of similar computers offered by the same vendor.

\_\_F\_\_\_ **(9)** Simulation is very useful for estimating the performance of systems or system configurations that only exist.

\_\_T\_\_\_ **(10)** Disk utilization, the measure of the percentage of time that the disk is busy servicing I / O requests.

**Part 3 Exercises - Performance Measurement and Analysis**

**(1)** **( Mean Values )**

Some types of averages include the arithmetic mean, the geometric mean and the harmonic mean.

The arithmetic mean is defined as the sum of the data values divided by their count.

The geometric mean of *n* positive numbers is the *n* - th root of their product.

The harmonic mean of *n* numbers is the reciprocal of the sum of the reciprocals of the numbers.

The execution times for a system running four benchmarks is shown in the table below. Compute the arithmetic, geometric and harmonic means of this data.

|  |  |
| --- | --- |
| *program* | System A Execution Time |
| W | 60 |
| X | 85 |
| Y | 70 |
| Z | 90 |

|  |
| --- |
| **Arithmetic mean: 76.25**  **Geometric mean: 75.28**  **Harmonic mean: 74.31** |

**(2)** **( Benchmarks and the Geometric Mean )**

The execution times for three systems running five benchmarks are shown in the table below. Compare the relative performance of each of these systems ( i.e., A to B , B to C

and A to C ) using the arithmetic and geometric means. Are there any surprises? Explain your analysis.

|  |  |  |  |
| --- | --- | --- | --- |
| *program* | System A  Execution Time | System B  Execution Time | System B  Execution Time |
| V | 150 | 200 | 80 |
| W | 200 | 250 | 150 |
| X | 275 | 170 | 200 |
| Y | 400 | 750 | 500 |
| Z | 900 | 1100 | 1200 |

|  |
| --- |
| **System C (the right most system) has the second highest arithmetic mean but actually the lowest geometric mean. It is likely that System C’s geometric mean is lower because of the lower execution times for the small values and how they are much less than the other two systems. This has a disproportionate influence on the overall result.** |

**(3)** **( Benchmarks and the Geometric Mean )**

The execution times for three systems running five benchmarks are shown in the table below. Compare the relative performance of each of these systems ( i.e., A to B , B to C

and A to C ) using the arithmetic and geometric means. Are there any surprises? Explain your analysis.

|  |  |  |  |
| --- | --- | --- | --- |
| *program* | System A  Execution Time | System B  Execution Time | System B  Execution Time |
| V | 40 | 100 | 70 |
| W | 325 | 275 | 350 |
| X | 275 | 100 | 300 |
| Y | 400 | 200 | 400 |
| Z | 800 | 1000 | 600 |

|  |
| --- |
| **System A has the highest arithmetic mean, followed by System C, and then System B. System C has the highest geometric mean, followed by System A, and then System B. It is somewhat surprising that System B has both the lowest arithmetic and geometric means, even though the execution times for program V and program Z are much higher than for the other systems. It is likely that System B’s geometric mean is balanced by the lower execution times for programs W, X, and Y.** |

**(4)** **( Synthetic versus Real - World Benchmarks )**

Comment on any effectiveness of synthetic benchmarking versus real - world benchmarking.

|  |
| --- |
| **Synthetic benchmarking is benchmarking under highly controlled circumstances. Oftentimes synthetic benchmarking is done on a system with a fresh install of the operating system, with very few programs running in the background. Real-world benchmarking is more useful because it gives a more realistic idea of what the benchmark would be under normal conditions.** |

**(5)** **( Performance Factors )**

Discuss some of the factors that affect the performance of processors, programs and magnetic disk storage.

|  |
| --- |
| **Factors that affect the performance of processors, programs, and magnetic disk storage include:**   * **Branching – delayed branching and branch prediction can minimize the delay on executing the instruction in the pipeline** * **Algorithms – using efficient algorithms can minimize the amount of operations and therefore improve performance** * **Disk scheduling – using an algorithm such as shortest seek time first can help improve performance** |

**Part 4 Exercises - Performance Measurement and Analysis**

Write a complete answer for each of these.

**(1) ( Benchmarks )**

What would you say to a vendor that tells you that his system runs 50 % of the SPEC benchmark kernel programs twice as fast as the leading competitive system? Which statistical fallacy is at work here?

|  |
| --- |
| **If this occurred I would ask the vendor what the results of the other system were. This is the incomplete information fallacy.** |

**(2) ( Synthetic Benchmarks )**

What are the limitations of synthetic benchmarks such as Whetstone and Dhrystone?   
 Do you think that the concept of a synthetic benchmark could be extended do overcome these limitations? Explain your answer.

|  |
| --- |
| **The limitation of synthetic benchmarks such as Whetstone and Dhrystone is that they are simple and easy to understand. This enables compiler writes to equip their products with Whetstone or Dhrystone compilation switches that invoke special code that is optimized for the benchmarks. Synthetic benchmarking could be extended to require that the code being benchmarked be built using the same compiler.** |

**(3) ( The Retail Business Sector )**

Besides the retail business sector, what other organizations would need good performance from a transaction - processing system. Justify your answer.

|  |
| --- |
| **Organizations such as high frequency trading firms need good performance from transaction processing systems because they require up to date data. The environment of high frequency trading is such that the data changes many times in a short amount of time, and being able to make decisions on accurate data is important.** |

**(4) ( Comparing Computer Models )**

Suppose a friend has asked you to help him to make a choice as to what kind of computer he should buy for his personal use at home. What would you look for in comparing various makes and models? How does your line of thinking differ in this situation than if you were to help your employer purchase a Web server to accept customers’ orders over the Internet?

|  |
| --- |
| **I would look at clock speed, MIPS, and FLOPs and evaluate those against what my friend is going to use the computer for. If he is not looking for high performance, then a computer with high ratings on those criteria is not as important. My line of thinking is this situation differs from an employer purchasing a web server because my friend’s computer does not really need to support a heavy transaction load, therefore his system does not need to perform as well.** |

**(5) ( Branching Predictors )**

In reference to branching, static prediction and fixed prediction are the same. Explain why this is so.

|  |
| --- |
| **Static branching is feeding the code into the system and receiving a result. If we continue to feed the same coed into the system, the result should be the same. Fixed prediction is when the outcome is always the same.** |