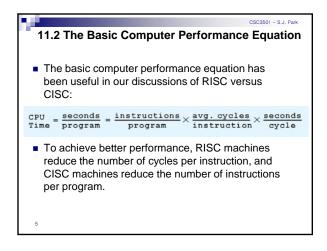


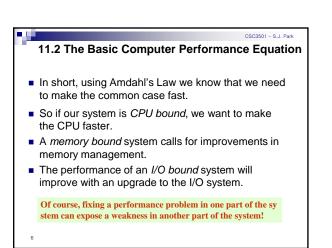
11.1 Introduction

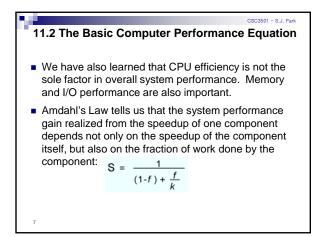
 The ideas presented in this chapter will help you to understand various measurements of computer performance.

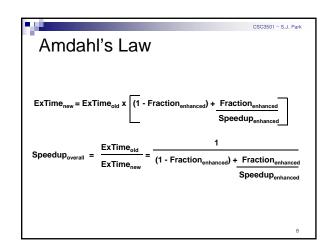
CSC3501 - S.J. Park

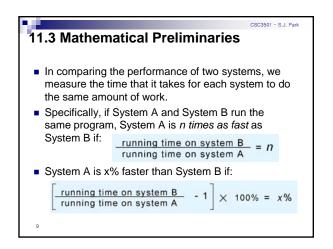
- You will be able to use these ideas when you are purchasing a large system, or trying to improve the performance of an existing system.
- We will discuss a number of factors that affect system performance, including some tips that you can use to improve the performance of programs.

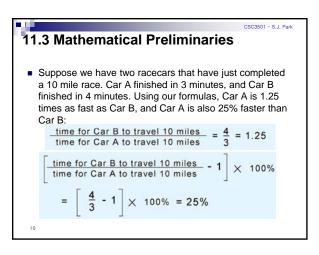


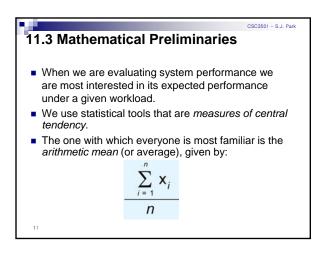


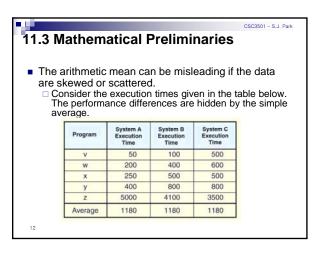












11.3 Ma	athema	atical F	Prelimi	naries	CSC3501 - S.J. Park
known □ The	n, a <i>weigl</i> weighteo .5 + 200 ×	<i>hted aver</i> d average	e (expecte rage can l for Syster × 0.1 + 400	be revea n A is:	
300.	Program	Execution Frequency	System A Execution Time	System C Execution Time]
1	v	50%	50	500	1
1	w	30%	200	600	1
1	x	10%	250	500	1
1	у	5%	400	800	1
	z	5%	5000	3500	1
Í	Weighted	Average	380 seconds	695 seconds	

However, workloads can change over time. A system optimized for one workload may perform poorly when the workload changes, as illustrated below.

Program	Execution Time	Execution Frequency #1	Execution Frequency #2
v	50	50%	25%
w	200	30%	5%
x	250	10%	10%
У	400	5%	5%
z	5000	5%	55%
Weighte	d Average	380 seconds	2817.5 seconds

11.4 Benchmarking Performance benchmarking is the science of making objective assessments concerning the performance of one system over another. *Price-performance ratios* can be derived from standard benchmarks.

The troublesome issue is that there is no definitive benchmark that can tell you which system will run your applications the fastest (using the least wall clock time) for the least amount of money.

CSC3501 - S.J. Park

CSC3501 - S.J. Park

11.4 Benchmarking

 Many people erroneously equate CPU speed with performance.

CSC3501 - S.J. Park

CSC3501 - S.J. Park

- Measures of CPU speed include cycle time (MHz, and GHz) and millions of instructions per second (MIPS).
- Saying that System A is faster than System B because System A runs at 1.4GHz and System B runs at 900MHz is valid only when the ISAs of Systems A and B are identical.
 - With different ISAs, it is possible that both of these systems could obtain identical results within the same amount of wall clock time.

11.4 Benchmarking

- In an effort to describe performance independent of clock speed and ISAs, a number of synthetic benchmarks have been attempted over the years.
- Synthetic benchmarks are programs that serve no purpose except to produce performance numbers.
- The earliest synthetic benchmarks, Whetstone, Dhrystone, and Linpack (to name only a few) were relatively small programs that were easy to optimize.
 This fact limited their usefulness from the outset.
- These programs are much too small to be useful in evaluating the performance of today's systems.

17

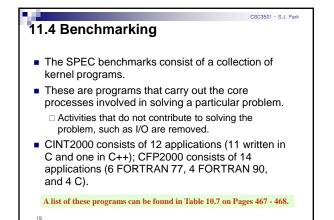
15

11.4 Benchmarking

16

18

- In 1988 the Standard Performance Evaluation Corporation (SPEC) was formed to address the need for objective benchmarks.
- SPEC produces benchmark suites for various classes of computers and computer applications.
- Their most widely known benchmark suite is the SPEC CPU benchmark.
- The SPEC CPU2000 benchmark consists of two parts, CINT2000, which measures integer arithmetic operations, and CFP2000, which measures floatingpoint processing.



- On most systems, more than two 24 hour days are required to run the SPEC CPU2000 benchmark suite.
- Upon completion, the execution time for each kernel (as reported by the benchmark suite) is divided by the run time for the same kernel on a Sun Ultra 10.

CSC3501 - S.J. Park

CSC3501 - S.J. Park

CSC3501 - S.J. Park

- The final result is the geometric mean of all of the run times.
- Manufacturers may report two sets of numbers: The peak and base numbers are the results with and without compiler optimization flags, respectively.

The SPEC CPU benchmark evaluates only CPU

 performance.
 When the performance of the entire system under high transaction loads is a greater concern, the *Transaction Performance Council* (TPC) benchmarks are more suitable.

CSC3501 - S.J. Park

CSC3501 - S.J. Park

- The current version of this suite is the TPC-C benchmark.
- TPC-C models the transactions typical of a warehousing and distribution business using terminal emulation software.

11.4 Benchmarking

20

22

11.4 Benchmarking

- The TPC-C metric is the number of new warehouse order transactions per minute (*tpmC*), while a mix of other transactions is concurrently running on the system.
- The tpmC result is divided by the total cost of the configuration tested to give a price-performance ratio.
- The price of the system includes all hardware, software, and maintenance fees that the customer would expect to pay.

11.4 Benchmarking

11.4 Benchmarking

- The Transaction Performance Council has also devised benchmarks for decision support systems (used for applications such as data mining) and for Web-based e-commerce systems.
- For all of the TPC benchmarks, the systems tested must be available for general sale at the time of the test and at the prices cited in a full disclosure report.
- Results of the tests are audited by an independent auditing firm that has been certified by the TPC.

23

21

11.4 Benchmarking

- TPC benchmarks are a kind of simulation tool.
- They can be used to optimize system performance under varying conditions that occur rarely under normal conditions.
- Other kinds of simulation tools can be devised to assess performance of an existing system, or to model the performance of systems that do not yet exist.
- One of the greatest challenges in creation of a system simulation tool is in coming up with a realistic workload.

24

To determine the workload for a particular system component, system traces are sometimes used. Traces are gathered by using hardware or software probes that collect detailed information concerning the activity of a component of interest. Because of the enormous amount of detailed information collected by probes, they are usually engaged for only a few seconds. Several trace runs may be required to obtain

- Several trace runs may be required to obtain statistically useful system information.
- 25

11.4 Benchmarking

26

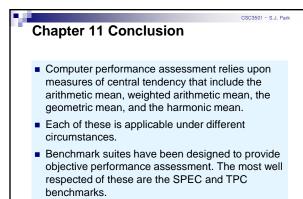
28

• Devising a good simulator requires that one keep a clear focus as to the purpose of the simulator

CSC3501 - S.J. Park

CSC3501 - S.J. Park

- A model that is too detailed is costly and timeconsuming to write.
- Conversely, it is of little use to create a simulator that is so simplistic that it ignores important details of the system being modeled.
- A simulator should be validated to show that it is achieving the goal that it set out to do: A simple simulator is easier to validate than a complex one.



Chapter 11 Conclusion

- CPU performance depends upon many factors.
- These include pipelining, parallel execution units, integrated floating-point units, and effective branch prediction.
- User code optimization affords the greatest opportunity for performance improvement.
- Code optimization methods include loop manipulation and good algorithm design.