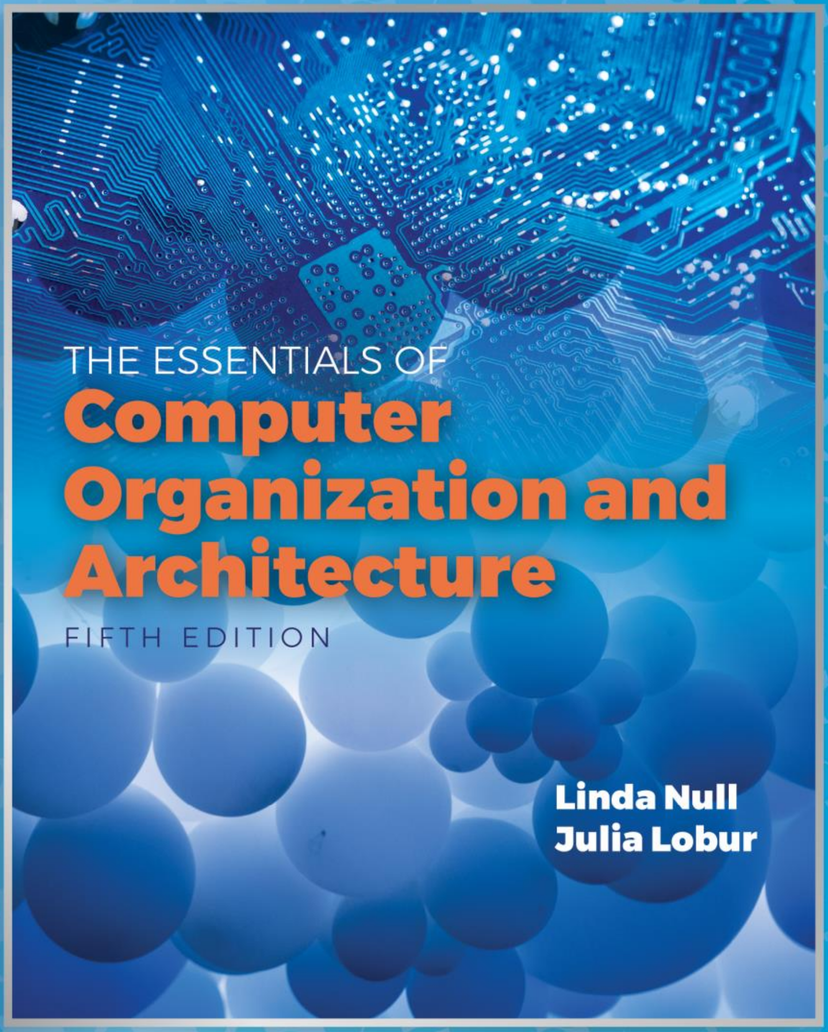
This is the third lecture of Chapter 11

Chapter 11

Performance Measurement and Analysis (C)



Quick review of last lecture

- Mathematical Preliminaries
  - System A is n times as fast as System B
  - System A is x% faster than System B
  - Arithmetic mean
  - Weighted arithmetic mean
  - Geometric mean
  - Harmonic mean

Mean	Uniformly Distributed Data	Skewed Data	Data Expressed as a Ratio	Indicator of System Performance Under a Known Workload	Data Expressed as a Rate
Arithmetic	X			X	
Weighted Arithmetic		X		X	
Geometric		X	X		
Harmonic				X	X

11.3 Mathematical Preliminaries (16 of 17)

- The objective assessment of computer performance is most critical when deciding which one to buy.
  - For enterprise-level systems, this process is complicated, and the consequences of a bad decision are serious.
- Unfortunately, computer sales are as much dependent on good marketing as on good performance.
- The cautious buyer will understand how objective performance data can be slanted to the advantage of anyone giving a sales pitch.

11.3 Mathematical Preliminaries (17 of 17)

- The most common deceptive practices include:
  - Selective statistics: Citing only favorable results while omitting others.
  - Citing only peak performance numbers while ignoring the average case.
  - Vagueness in the use of words like “almost,” “nearly,” “more,” and “less,” in comparing performance data.
  - The use of inappropriate statistics or “comparing apples to oranges.”
  - Implying that you should buy a particular system because “everyone” is buying similar systems.

Many examples can be found in business and trade journal ads.

11.4 Benchmarking (1 of 12)

- 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.

11.4 Benchmarking (2 of 12)

- Many people erroneously equate CPU speed with performance.
- 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 (3 of 12)

- 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.

11.4 Benchmarking (4 of 12)

- 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 CPU2017 benchmark suit has a total of 43 benchmarks that are organized into four suites, two for integers and two for floating point numbers.

11.4 Benchmarking (5 of 12)

- The SPEC benchmarks basically consist of a collection of kernel programs.
- These are programs that carry out the core processes involved in solving a particular problem.
  - Activities that do not contribute to solving the problem, such as I/O are removed.
- A list of these programs can be found in Table 11.7 on Pages 601–602.

SPEC CPU2017 Benchmarks (1)

SPECrate 2017 Integer	SPECspeed2017 Integer	Language	Line Count	Application Area
500.perlbench_r	600.perlbench_s	C	362	Perl interpreter
502.gcc_r	602.gcc_s	C	1,304	GNU C compiler
505.mcf_r	605.mcf_s	C	3	Route planning
520.omnetpp_r	620.omnetpp_s	C++	134	Discrete Event simulation - computer network
523.xalancbmk_r	623.xalancbmk_s	C++	520	XML to HTML conversion via XSLT
525.x264_r	625.x264_s	C	96	Video compression
531.deepsjeng_r	631.deepsjeng_s	C++	10	Artificial Intelligence: alpha-beta tree search (Chess)
541.leela_r	641.leela_s	C++	21	Artificial Intelligence: Monte Carlo tree search (Go)
548.exchange2_r	648.exchange2_s	Fortran	1	Artificial Intelligence: recursive solution generator (Sudoku)
557.xz_r	657.xz_s	C	33	General data compression

SPEC CPU2017 Benchmarks (2)

SPECrate 2017 Floating Point	SPECspeed 2017 Floating Point	Language	Line Count	Application Area
503.bwaves_r	603.bwaves_s	Fortran	1	Explosion modeling
507.cactuBSSN_r	607.cactuBSSN_s	C++, C, Fortran	257	Physics: relativity
508.namd_r		C++	8	Molecular dynamics
510.parest_r		C++	427	Biomedical imaging: optical tomography with finite elements
511.povray_r		C++, C	170	Ray tracing
519.lbm_r	619.lbm_s	C	1	Fluid dynamics
521.wrf_r	621.wrf_s	Fortran, C	991	Weather forecasting
526.blender_r		C++, C	1,577	3D rendering and animation
527.cam4_r	627.cam4_s	Fortran, C	407	Atmosphere modeling
	628.pop2_s	Fortran, C	338	Wide-scale ocean modeling (climate level)
538.imagick_r	638.imagick_s	C	259	Image manipulation
544.nab_r	644.nab_s	C	24	Molecular dynamics
549.fotonik3d_r	649.fotonik3d_s	Fortran	14	Computational Electromagnetics
554.roms_r	654.roms_s	Fortran	210	Regional ocean modeling

11.4 Benchmarking (6 of 12)

- On most systems, more than two 24 hour days are required to run the SPEC CPU benchmark suite.
- Upon completion, the execution time for each kernel is divided by the run time for the same kernel on a Sun Ultra Enterprise 2 workstation.
- 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.

11.4 Benchmarking (7 of 12)

- 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.
- 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 (8 of 12)

- 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 (9 of 12)

- 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.

11.4 Benchmarking (10 of 12)

- 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.

11.4 Benchmarking (11 of 12)

- 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 statistically useful system information.

11.4 Benchmarking (12 of 12)

- Devising a good simulator requires that one keep a clear focus as to the purpose of the simulator.
- A model that is too detailed is costly and time-consuming 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.