MODULE 14: Selected Topics 2

Lecture 14.3
Benchmarking

Prepared By:
- Scott F. Midkiff, PhD
- Luiz A. DaSilva, PhD
- Kendall E. Giles, PhD

Electrical and Computer Engineering
Virginia Tech

Lecture 14.3 Objectives

- Describe the role of benchmarks in assessing computer system performance
- List important benchmarks used for different sectors including scientific computing, transaction processing, and more general computing
- Discuss potential pitfalls of benchmarks

Benchmarking

- Benchmarking, generally, is the systematic comparison of one system (approach, etc.) to another under normalized conditions
- Benchmarking, specifically to computer systems, is the systematic comparison of one computer system to another using standardized workloads and parameters
- Goals of benchmarking include:
  - Selecting the best system for a given use scenario
  - Comparing an existing system to a potential upgrade
  - Differentiating systems in the market
  - Driving system design to best meet user needs

Issues with Metrics

- Common simple metrics for computer system performance
  - Clock rate, such as 2.5 Gigahertz (GHz)
  - Instruction rate, such as 100 million instructions per second (MIPS)
  - Floating point operation rate, such as 1 billion (giga) floating point instructions per second (1 GFLOP)
- What happens during one clock cycle?
  - Execution of an entire instruction? That does what?
  - Execution of just part of an instruction?

Issues with Metrics (cont'd)

- What does an "instruction" do?
  - Simple operations, as on a RISC processor?
  - Complex operations, as on a CISC processor?
- What is a floating point operation?
  - A full operation?
  - A partial operation?

Benchmarks, if properly defined and used, can provide a normalized ("apples to apples") comparison, at least for the stated scope of the benchmark.

Benchmarking and Metrics

- A good benchmark considers metrics that are associated with user (or service provider's) context versus the context of the computer
- Number of transactions completed per second versus number of instructions per second
- Number of floating point operations as defined by a set of applications versus the number of processor floating point operations per second
- Good benchmarks also provide a consistent procedure for measuring performance – what is measured and how it is to be measured

Types of Benchmarking Metrics

- Throughput – operations per unit of time
- Response time – time to complete a set of tasks
- Price-performance ratio – the cost for a unit of performance

Synthetic Benchmarks

- A synthetic benchmark is a single application that is constructed specifically to exercise a range of types of computations and processor functions
  - Specified in a high-level language (the semantics are functional rather than associated with a processor)
  - Compiled for different architectures
- Largely focused on scientific computing
  - Whetstone, 1976 – trigonometric and exponential functions
  - Linear Algebra Package (LINPACK), 1984 – use for supercomputer evaluation
  - Dhrystone, 1984 – CPU bound (no I/O)

Synthetic Benchmarks (cont'd)

- Advantage
  - Simple to apply and understand
  - Promoted the concept of benchmarks
- Disadvantage
  - Limited range of operations, so it is not representative for many applications
  - Too easy to "game" through architectural optimizations and, especially, compiler optimizations
  - Small program can fit in cache, so memory management is not exercised

CHECK POINT

As a checkpoint of your understanding, please pause the video and make sure you can do the following:

- Describe the role of benchmarks in assessing computer system performance

If you have any difficulties, please review the lecture video before continuing.

SPEC

- Benchmarks from the Standard Performance Evaluation Corporation (SPEC) attempt to address the limitations of synthetic benchmarks
  - Non-profit corporation founded in 1988 by computer manufacturers and the Electrical Engineering Times
  - Many companies and organizations are members and associates
  - SPEC on the web: http://www.spec.org/
- Three main groups, each with their own benchmarks
  - Open Systems Group (OSG)
  - High Performance Group (HPG)
  - Graphics and Workstation Performance Group (GWPG)

SPEC Benchmarks*

- CPU
  - SPEC CPU
- Graphics and workstation performance
  - SPECviewperf
  - SPECwpc
  - SPECapc
- High performance computing, OpenMP, MPI
  - SPEC ACCEL
  - SPEC MPI
  - SPEC OMP
  - SPEC HPC

SPEC Benchmarks* (cont'd)

- Java Client/Server
  - SPECjbb
  - SPECjms
  - SPECjEnterprise
  - SPECjvm
- Cloud Computing
  - SPEC Cloud_IaaS
- File Server
  - SPEC SFS

SPEC Benchmarks* (cont'd 2)

- Power
  - SPECpower
- Virtualization
  - SPEC VIRT

SPEC CPU Benchmarks

- Suites of CPU-intensive applications (kernels) that are representative of actual applications
- Two suites
  - SPECint – measures compute-intensive integer performance
  - SPECfp – measures compute-intensive floating point performance
- Both kernels represent a variety of applications from word processing, to design automation, to compression, to scripts

Transaction Processing and the TPC

- Transaction processing is input/output intensive, but I/O is not exercised by the SPEC CPU benchmarks
- The Transaction Processing Performance Council (TPC) has developed benchmarks to assess transaction processing systems
  - Non-profit industry group founded in 1988
  - More than 25 members worldwide
  - On the web at http://www.tpc.org/

TPC Benchmarks

- TPC-C
  - Database transactions
- TPC-DI
  - Data integration
- TPC-DS
  - Decision support w/ big data
- TCP-E
  - Online transaction processing
- TPC-H
  - Decision support
- And others

Good Benchmarks

- Benchmarks can be helpful in evaluating and comparing performance
- Good benchmarks:
  - Should be representative
  - Should not provide a non-representative advantage to a particular architecture or configuration
  - Should be performed in a fair and consistent manner
  - Must present results in a uniform manner
  - Should be open to scrutiny
- Not all benchmarks have all properties and no benchmark can represent every application – caveat emptor!

CHECK POINT

As a checkpoint of your understanding, please pause the video and make sure you can do the following:

- List important benchmarks used for different sectors including scientific computing, transaction processing, and more general computing
- Discuss potential pitfalls of benchmarks

If you have any difficulties, please review the lecture video before continuing.

Summary

- Simple measures for computer performance, such as MIPS, FLOPS, and clock rate, are inadequate for assessing performance across different architectures and application sectors
- Benchmarks are used to compare one computer system to another using standardized workloads and parameters
- Benchmarks may assess response time, throughput, and price-performance
- Benchmarks have been developed by user communities (e.g., synthetic benchmarks) and industry groups (SPEC and TPC)

Summary (cont'd)

- Synthetic benchmarks (e.g., Whetstone and LINPACK) are programs created to be representative, while other benchmarks (e.g., the SPEC and TPC benchmarks) are created from representative and popular applications
- Benchmarks help in assessing and comparing performance, but they are based on certain assumptions which one should understand before fully accepting results

MODULE 14: Selected Topics 2

Lecture 14.3
Benchmarking

Prepared By:
- Scott F. Midkiff, PhD
- Luiz A. DaSilva, PhD
- Kendall E. Giles, PhD

Electrical and Computer Engineering
Virginia Tech