Types of Workloads

Raj Jain
Washington University in Saint Louis
Saint Louis, MO 63130
Jain@cse.wustl.edu

These slides are available on-line at:

http://www.cse.wustl.edu/~jain/cse567-13/

Overview

- Terminology
- Test Workloads for Computer Systems
  - Addition Instruction
  - Instruction Mixes
  - Kernels
  - Synthetic Programs
  - Application Benchmarks: Sieve, Ackermann's Function, Debit-Credit, SPEC

Part II: Measurement Techniques and Tools

Measurements are not to provide numbers but insight

- Ingrid Bucher

1. What are the different types of workloads?
2. Which workloads are commonly used by other analysts?
3. How are the appropriate workload types selected?
4. How is the measured workload data summarized?
5. How is the system performance monitored?
6. How can the desired workload be placed on the system in a controlled manner?
7. How are the results of the evaluation presented?

Terminology

- **Test workload**: Any workload used in performance studies. Test workload can be real or synthetic.
- **Real workload**: Observed on a system being used for normal operations.
- **Synthetic workload**:
  - Similar to real workload
  - Can be applied repeatedly in a controlled manner
  - No large real-world data files
  - No sensitive data
  - Easily modified without affecting operation
  - Easily ported to different systems due to its small size
  - May have built-in measurement capabilities.

Desired Characteristics of a Benchmark

1. **Representative**: Of an application area, e.g., databases, scientific computing
2. **Portable**: Run on many different architectures and implementations
3. **Unbiased**: Not designed to favor a particular system
4. **Scalable**: Run on both small and large systems
5. **Measurable**: Easy to measure ⇒ Broad acceptance
6. **Repeatable**: Minimum variance
7. **Explainable**: Single number easy to understand.

Ref: R. Weicker, "Benchmarking," in M.C.Calzaroosa and S. Tucci (Eds.), Performance 2002, Lecture Notes in Computer Science 2459, Springer-Verlag Berlin Heidelberg, 2002, pp. 179-207.

Test Workloads for Computer Systems

1. Addition Instruction
2. Instruction Mixes: Usage frequency
3. Kernels: Key functions, e.g., sorting. Matches order = micro-benchmark
4. Synthetic Programs: Bigger programs
5. Application Benchmarks: Matches I/O

Note: Please read about these in the book.

Synthetic Programs

- Examples:
  - **Whetstone** (1972): Floating point intensive numerical code
  - **Linpack** (1976): Linear algebra package (Still used by top500.org to rank supercomputers)
  - **Dhrystone** (1984): Integer only, system code
- Problem:
  - Single author
  - Easily manipulated by targeted compiler optimizations
  - Not subject to memory hierarchy issues

Vendor Specific Benchmarks

- Windows System Assessment Tool (WinSAT)
  - Control Panel → Performance Information and Tools
  - Reports Windows Experience Index (WEI)
  - Five sub-scores: Processor, memory, 2D graphics, 3D graphics, disk
  - Overall = **min** of five sub-scores
- SAP, Lotus/IBM, Oracle, Baan/Infor Global Solutions
- Use binary/machine code

Industry Benchmarks Associations

- Systems Performance Evaluation Cooperative (SPEC)
- Business Applications Performance Corporation (BAPCo):
  - Personal computer benchmarks
  - Controversial SYSmark 2012 benchmark
- The Embedded Microprocessor Benchmark Consortium (EEMBC)
  - For processors used in autos, printers, networking devices, cameras, smart phones
  - Performance and energy consumption
- Transaction Processing Council (TPC): Database systems
- Storage Performance Council (SPC): Storage Systems

Ref: http://en.wikipedia.org/wiki/Benchmark_%28computing%29

SPEC Benchmark Suite

- Non-profit corporation formed in 1988 by leading computer vendors to develop a standardized set of benchmarks.
- Portable: Written in a platform neutral programming language, e.g., C, Java, Fortran
- Subject to compiler optimizations
- Includes run rules about measurement conditions and documentation

Current SPEC Benchmarks

- **SPECapc**: Graphics rendering using Autodesk 3ds Max 2011, NewTec LightWave 3D v9.6, Autodesk Maya 2012, PTC Creo 2.0 (3D CAD), Siemens NX 6, SolidWorks 2007
- **SPECviewperf 11**: Graphics performance on Open GL 3d Systems
- **SPEC CPU2006**: Integer and floating point performance using compilers, word processors, 3D graphics, etc
- **SPECjbb2005**: Server side Java
- **SPECjEnterprise2010**: Java 2 Enterprise Edition servers
- **SPECjms 2007**: Java message service
- **SPECjvm2008**: Java runtime environment on clients and servers
- **SPEC MPI2007**: Message passing interface on parallel systems
- **SPEC OMP2012**: Using OpenMP (parallel programming API)
- **SPECpower_ssj2008**: Energy efficiency of server systems
- **SPECsfs2008**: NFS and CIFS protocol performance
- **SPECsip_Infrastructure2011**: SIP server perfromance
- **SPECvirt_sc2010**: Virtualized server consolidation in data centers

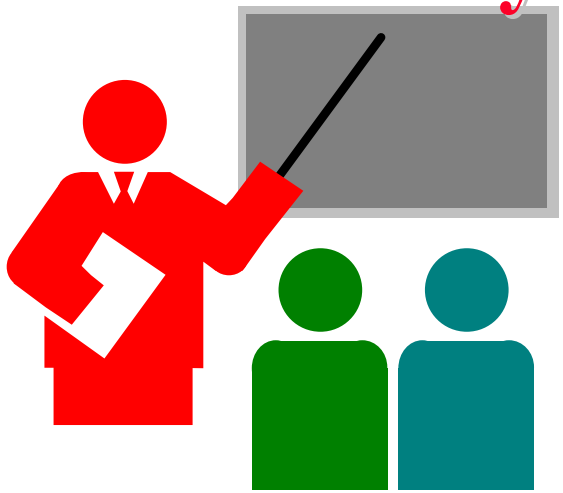
SPEC CPU Benchmark

- Consists of many programs contributed by members, open source, academics
- Ported to various platforms to remove dependency on I/O or operating systems
- Compilation and execution is automated
  Tester supplies a configuration file with location of C compiler, compilation flags, etc.
- Speed = Execution time on the system normalized by a well known system
- Geometric mean of ratios is reported as SPECint or SPECfp
- Rate = Throughput = jobs/sec. Geometric mean ⇒SPECrate

Benchmarking Issues

- Optimized compilations not representative of real-world
  ⇒ Baseline and peak numbers
- Vendors publicize only peak numbers and not baseline numbers
- n-CPU systems run n copies in parallel
  ⇒Not representative since there is no queue of waiting jobs
- Source codes of commonly used programs are not available for use as benchmarks ⇒Mostly open source
- Long run times allow jobs to run from cache
  ⇒ Not representative
- High performance on high priced systems
  ⇒ Price/performance ratio
  Price easily manipulated and varied by configurations
- Algorithms/systems developed to optimize benchmarks

Summary



1. Benchmarks have moved from instructions (MIPS), kernels, synthetic benchmarks to industry standard benchmarks
2. Benchmarks should be representative, portable, unbiased, scalable, measurable, repeatable, and explainable
3. Difficult to ensure representativeness of the benchmark (Compiler flags, run time, configuration, can be manipulated)
4. Price and energy consumption are important along with performance

Homework 4

- Make a list of 25 benchmarks not discussed in this lecture. Write one sentence description of each benchmark in your own words.

- Hint: http://en.wikipedia.org/wiki/Benchmark_%28computing%29