Types of Workloads

• Test workload – denotes any workload used in performance study
• Real workload – one observed on a system while being used.
  - cannot be repeated (easily)
  - may not even exist (proposed system)
• Synthetic workload – similar characteristics to real workload
  - can be applied in a repeated manner
  - relatively easy to port
• Benchmark == Workload
  - Benchmarking is process of comparing 2+ systems with workloads

Outline

• Introduction
• Addition instructions
• Instruction mixes
• Kernels
• Synthetic programs
• Application benchmarks

Addition Instructions

• Early computers had CPU as most expensive component
• Most frequent operation was addition
• Computer with faster addition instruction performed better
• So, run many addition operations as test workload
• Problem
  - More instructions used
  - Some more complicated than others

Instruction Mixes

• Number and complexity of instructions increased
• Could measure instructions individually, but used in different amounts
  - Measure relative frequencies of various instructions on real systems
  - Use as weighting factors to get avg instruction time

→ Instruction mixes
• Units are
  - Millions of Instructions Per Second (MIPS)
  - Millions of Floating-Point Ops per Sec (MFLOPS)

Example: Gibson Instruction Mix

1. Load and Store 13.2
2. Fixed-Point Add/Sub 6.1
3. Compares 3.8
4. Branches 16.6
5. Float Add/Sub 6.9
6. Float Multiply 3.8
7. Float Divide 1.5
8. Fixed-Point Multiply 0.6
9. Fixed-Point Divide 0.2
10. Shifting 4.4
11. Logical And/Or 1.6
12. Instructions not using regs 5.3
13. Indexing 18.0

Total 100
Problems with Instruction Mixes

• In modern systems, instruction time variable depending upon
  - Addressing modes, cache hit rates, pipelining
  - Interference with other devices during processor-memory access
  - Distribution of zeros in multiplier
  - Times a conditional branch is taken
• Mixes do not reflect special hardware such as page table lookups
• Only represents speed of processor
  - Bottleneck may be in other parts of system

Kernels

• Used set of instructions that made up a service provided by processor. A kernel.
  - Early on, did not consider I/O so also called a processing kernel
• Set of operations for problem
  - Ex: Sieve, Tree Searching, Matrix Inversion
• Some problems such as zeros and branches don’t apply
• Problem
  - I/O still not considered

Synthetic Programs

• Add I/O request to test load
• Add control loop so can make request as frequently as needed
• Easy to port, distribute
• Can have measurement data built in
• Still, does not necessarily make representative memory or disk accesses
• Often small, so do not exercise virtual memory

Application Workloads

• For special-purpose system, may be able to run representative applications as measure of performance
  - Ex: airline reservation
  - Ex: banking
• Make use of entire system (I/O, etc).
• Issues may be
  - input parameters
  - multiuser
• Only applicable when specific applications are targeted

Popular Benchmarks: Sieve (1 of 2)

• Sieve of Eratosthenes (finds primes)
  • Write down all numbers 1 to \( n \)
  • Strike out multiples of \( k \) for \( k = 2, 3, 5 \ldots \sqrt{n} \)
    • In steps of remaining numbers
Popular Benchmarks: Sieve (2 of 2)

1. Write down all numbers from 1 to 20.
   - Mark all as prime:
     \[1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15 \ 16 \ 17 \ 18 \ 19 \ 20\]
2. Remove all multiples of 2 from the list of primes:
   \[1 \ 2 \ 3 \ 4 \ 5 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 15 \ 16 \ 17 \ 18 \ 19 \ 20\]
3. The next integer in the sequence is 3.
   - Remove all multiples of 3:
     \[1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 14 \ 16 \ 17 \ 18 \ 19 \ 20\]
4. \(5 \geq \sqrt{20}\) => Stop

Popular Benchmarks: Ackermann’s Function (1 of 2)

- Assess efficiency of procedure calling mechanisms
- Ackermann’s Function has two parameters, is recursive
  - Benchmark is to call \(\text{Ackermann}(3, n)\) for values of \(n = 1\) to 6
- Return value is \(2^{n^3} - 3\), can be used to verify implementation
- Number of calls:
  \[
  \frac{512 \times 4^{n-1} - 1}{5 	imes 2^{n+3} + 9n + 37}/3
  \]
  - Can be used to compute time per call
- Depth is \(2^{n^3} - 4\), stack space doubles \(n++\)

Popular Benchmarks: Ackermann’s Function (2 of 2)

(Prolog)

Popular Benchmarks: Whetstone

- Set of 11 modules designed to match observed frequencies in ALGOL programs
  - Array addressing, arithmetic, subroutine calls, parameter passing
  - Ported to Fortran, most popular in C, ...
- Many variations of Whetstone, so take care when comparing results
- Problems - specific kernel
  - only valid for small, scientific (floating) apps that fit in cache
  - Does not exercise I/O

Popular Benchmarks: LINPACK

- Programs that solve dense systems of linear equations
  - Many float adds and multiplies
  - Core is Basic Linear Algebra Subprograms (BLAS), called repeatedly
- Usually, solve 100x100 system of equations
- Represents mechanical engineering applications on workstations
  - Drafting to finite element analysis
  - High computation speed and good graphics processing

Popular Benchmarks: Dhrystone

- Pun on Whetstone
- Intent to represent systems programming environments
- Most common was in C, but many versions
- Low nesting depth and instructions in each call
- Large amount of time copying strings
- Mostly integer performance with no float operations
Popular Benchmarks: Lawrence Livermore Loops
- 24 vectorizable, scientific tests
- Floating point operations
  - Physics and chemistry apps have found 40-60% floating point operations
- Relevant for: fluid dynamics, airplane design, weather modeling

Popular Benchmarks: Debit-Credit
- Was Defacto Standard for Transaction Processing Systems
- Retail bank wanted 1000 branches, 10k tellers, 10000k accounts online with peak load of 100 TPS
- Performance in TPS where 95% of all transactions with 1 second or less of response time (arrival of last bit, sending of first bit)
- Now, Transaction Processing Council (TPC) has made more precise benchmarks
  - TPC-A, TPC-B, TCP-C

Popular Benchmarks: SPEC
- Systems Performance Evaluation Cooperative (SPEC) (http://www.spec.org)
  - Non-profit, leading computer vendors
  - Suite of benchmarks
- CPU2000: CPUINT and CPUFP
  - Making CPU2004
- Graphics
- Systems and Applications:
  - Web, Java Client-Server, Network Files System, Mail
- Results database
- Performance compared to baseline machine