IV. Measuring and Reporting Performance (Cont'd)

- b. Measuring Performance
    - i. Execution time
        1. Wall-clock time, response time or elapsed time: latency to complete a task
        2. CPU time: time the CPU is computing
        3. User CPU time: time spent in user program
        4. System CPU time: time spent in OS performing tasks requested by the program
    - ii. System performance: elapsed time on an unloaded system
    - iii. CPU performance: User CPU time on an unloaded system
        1. THIS IS THE FOCUS OF THIS BOOK
- c. Choosing Programs to Evaluate Performance (Benchmarks)
    - i. Ideal is measuring execution time of user's workload
    - ii. Other methods:
        1. Real applications (Not usually portable)
            - a. Language compilers
            - b. Applications like Photoshop
        2. Modified or scripted applications: Scripts eliminate user I/O
        3. Kernels: small key pieces of real programs
            - a. Livermore Loops
            - b. Linpack
        4. Toy benchmarks: very small programs with known results
            - a. Sieve of Eratosthenes
            - b. Quicksort
        5. Synthetic benchmarks: try to match average frequency of operations and operands (not necessarily from real programs, as are kernels.) Easily defeated!
            - a. Whetstone
            - b. Dhrystone
- d. Benchmark suites
    - i. Desktop benchmarks
        1. CPU-intensive
            - a. SPEC CPU2000: real applications
                - i. CINT2000: 11 integer benchmarks
                    1. C-compiler
                    2. VLSI router, e.g.
                - ii. CFP2000: 14 floating-point benchmarks
                    1. Finite element modeling
                    2. Fluid dynamics, e.g.
        2. Graphics-intensive
            - a. SPECviewperf: 3D rendering using OpenGL
            - b. SPECapc: several large applications for rendering, I/O intensive and CPU intensive 3D CAD/CAM …
    - ii. Server benchmarks

1. CPU throughput measurement
    a. SPECrate: measurement of rate of running multiple copies of SPEC2000 applications
2. I/O intensive application throughput measurement
    a. File server benchmark
        i. SPECSFS (NFS performance)
    b. Web server benchmark
        i. SPECWEB: simulates multiple clients
    c. Transaction-processing benchmarks (measure transactions per second)
        i. TPC-C: complex query environment
        ii. TPC-R: queries for business decision support system
        iii. TPC-W: queries Web-based business-oriented transaction system

iii. Embedded benchmarks
1. Dhrystone performance (easily optimized)
2. Application-specific benchmarks from manufacturers
3. EEMBC (EDN Embedded Microprocessor Benchmark Consortium) benchmarks
    a. Five types: Automotive/industrial, Consumer (multimedia), Networking, Office automation, Telecommunications

e. Reporting Performance Results
    i. Guiding principle: reproducibility
    ii. System's configuration and compiler technology greatly influence results
        1. Baseline performance measurement
            a. One set of compiler flags for all programs
        2. Optimized performance measurement
            a. Different settings for each program
    iii. Source code modifications
        1. SPEC allows NO source code modifications
        2. NAS supercomputer benchmarks and EEMBC benchmarks allow source code optimizations
        3. EEMBC allows assembly coding of routines (embedded computers)

f. Comparing and Summarizing Performance: how do you compare different machine's results???
    i. Relative performance can be confusing
    ii. Total execution
    iii. Average execution time (assumes equal weights)
    iv. Weighted execution time: weight by frequency of execution in real workload
    v. Normalized execution time
        1. SPEC normalizes to SUN performance

2. DEC normalized to VAX 11/780 architecture
3. Geometric means preserve individual performance ratios:
    a. Ratio of means = mean of ratios
    b. Independent of running times of individual programs
    c. Doesn't matter which machine is used to normalize
    d. Violate fundamental concept: They DO NOT PREDICT EXECUTION TIMES. There is NO workload for three or more machines that will match the performance predicted by geometric means of normalized execution times!!!
    e. Encourages mfg's to work on what is easiest to improve rather than on the slowest benchmarks
    f. <u>Used by SPEC</u>

vi. MIPS (Millions of instructions per second)
vii. MFLOPS (Millions of floating-point operations per second)

V. Quantitative Principles of Computer Design
  b. Make the common case fast: It doesn't help to optimize something that is never executed!!
  c. Amdahl's law
    i. Formula for speedup
    ii. Expresses the law of diminishing returns
  d. CPU Performance Equation
    i. CPU time
    ii. Clocks per instruction (CPI)
    iii. Clock rate v/s Clock cycle time

VI. Principle of Locality
  b. Temporal locality: reuse of data and instructions that have been used recently
  c. Spacial locality: items near to one another tend to be referenced close together in time