MISP* (not MIPS!) *Meaningful Indicators of System Performance

Dr. Charles Sauer, Dell Computer Corporation

Everyone knows that a MIP is a "Meaningless Indicator of Performance," but MIPS (Millions of Instructions Per Second) is still the most widely quoted measure of system performance. In principle, meaningful indications of system performance can be obtained by running the intended applications on the systems under consideration and measuring the time to completion. Unfortunately, this is often impractical.

First, because it may not be possible to identify a representative set of applications, e.g., due to the diversity amongst users of a system. Second, the intended applications may not be written yet. Third, the intended applications may not have yet been ported to the systems under consideration.

If it is not possible to do performance measurements using the intended application, the next best thing is measuring the time to complete a meaningful benchmark, a program seen as representative of the intended applications. The problem is then how to get a meaningful indication of performance using benchmarks, since benchmarking has many problems as well.

It may not be obvious what system resources (CPU, disk, network, etc.) the applications will utilize most heavily, so a benchmark focused on any one of these resources may be irrelevant. Similarly, trying to weight the results of different benchmarks assumes it is possible to correlate the benchmark mix with the intended application.

To further compound the problem, there may not be any existing benchmarks that represent the desired resource usage (writing a representative benchmark is difficult in itself). And, the benchmarks may be subject to system characteristics such as optimizing compilers, "small" caches in the memory hierarchy, "small" caches in the disk subsystem, etc., that improve benchmark timings much more than they improve intended application performance.

Instruction-Mix Benchmarks

Most early attempts to measure processor performance were based on mixes of instructions which were believed to be characteristic of typical programs. Usually, these mixes were derived from histograms of instructions executed in snapshots of applications or operating systems. As long as the timing of instructions is easily determined, it is possible to calculate a MIPS rating as an estimate of system performance. However, it may not be easy to determine instruction timing in the presence of cache memories, pipelining, and other effects.

Instruction mixes are normally expressed in terms of machine instructions, so they are less intuitive than benchmarks written in high level languages. MIPS ratings do not reflect relative strengths and weaknesses of instruction sets. Higher MIPS ratings may actually be a result of reducing system performance. The following short history of benchmarks may help to clarify some of the confusion surrounding these performance claims.

The Whetstone Benchmark

The high level language analog of an instruction mix is the *Synthetic Benchmark*, a program written with the intention of emulating measurements of actual programs. Perhaps the first well known example is the Whetstone benchmark, often used as a measure of floating point performance. There are several limitations to the usage of Whetstone. One, the code makes heavy usage of transcendental library functions - the benchmark timing can be improved dramatically by library tuning. (Such tuning will have a lesser effect on applications with lesser use of these functions.)

Second, there is a substantial proportion of fixed point computation in the benchmark, thus reducing its effectiveness as a floating point benchmark. Third, optimizing compilers can discard significant portions of the code and/or replace function calls with inline code, in ways not possible with real applications. Nevertheless, the Whetstone benchmark was widely used for a number of years and is still in use as a floating point benchmark. Results are reported as "Whetstones per second" - most systems with floating point hardware will have ratings of more than a million Whetstones/second.

Dhrystone and Drhystone MIPS

A contender with MIPS for the most widely quoted measure of performance, the Dhrystone benchmark is intended to represent characteristics of systems programs and fixed point applications. Dhrystone has analogous limitations to Whetstone, e.g., impacts of library tuning, optimizers eliminating "unneeded" code, inlining, etc. There have been several versions intended to reduce these effects - the most recent is 2.1. The compiled program will fit in small caches, so impacts of memory subsystems are not measured. Results are quoted as "Dhrystones/second."

With a reasonably good compiler, say the GCC compiler from GNU, a VAX 11/780 does just under 1800 Dhrystones/second with Dhrystone version 2.1 (vs. just under 1900 Dhrystones/second with version 1.1). A common marketing practice for a new machine is to assume the VAX 11/780 to be a 1 MIPS machine (in native VAX MIPS it is closer to a .5 MIPS machine), then take the new machine's Dhrystone rating divided by the 780's Dhrystone rating to derive a "Dhrystone MIPS" rating.

These derived ratings bear little relationship to more direct measurements of the machine's speed in instructions per second, yet they are probably the most widely quoted ratings of machine speed in common practice today. In ways this is reminiscent of wattage claims for audio amplifiers a couple of decades ago. (The VAX figures used may be significantly lower than the above, since VAX ratings as low as 1400 Dhrystones/second are quoted for compilers with limited optimization.)

The Linpack Benchmark

One of the earliest independent collections of benchmark results for a wide array of machines is the so-called Linpack report from Argonne Labs. Linpack is a linear algebra package - the benchmark is based on solution of a 100x100 system of linear equations, with results reported in MFLOPS (millions of floating point operations). Though the benchmark is often a reasonable indicator of floating point performance, the results can be heavily influenced by cache size and effectiveness of the memory hierarchy when the matrix does not fit in the cache. Linpack has effectively displaced Whetstone to become the most widely quoted measure of floating point performance.

MIPS Performance Brief

John Mashey of MIPS Computer Systems periodically produces a broad overview of benchmark summaries and results for a variety of benchmarks and systems. This brief includes discussion and results for Whetstone, Dhrystone 1.1 and 2.1, and LINPACK, as well as a variety of lesser known benchmarks, for a cross-section of machines from the DEC VAX 11/780 on up through Cray supercomputers.

SPEC (Systems Performance Evaluation Cooperative)

Systems Performance Evaluation Cooperative is a non-profit corporation formed to establish, maintain, and endorse a standardized set of relevant benchmarks that can be applied to the newest generation of high-performance computers." With the exception of Linpack, most of the popular CPU benchmarks prior to SPEC were synthetic, and arguably not representative of real applications, and/or were fairly trivial. (Some of the benchmarks still in use today are literally only one line of source code.) SPEC has tried to establish much more rigorous standards for benchmarking, starting with processor-oriented benchmarks.

SPEC release 1.0 is a suite of ten real applications to be used as benchmarks. Four of the benchmarks are oriented toward fixed-point computation and six toward floating-point, with the former written in C and the latter written in Fortran. Most of the benchmarks are memory intensive, so results are usually quoted for machines with sixteen megabytes or more of main memory, to avoid paging. Given sufficient memory to avoid paging, the benchmark results are typically dominated by computation, though some do require non-trivial I/O; e.g., one of the benchmarks is based on compilation using the GNU C compiler. The benchmarks are all fairly long-running; e.g., the shortest running benchmark requires well over a quarter of an hour on a VAX 11/780.

Though using SPEC 1.0 is not the same as using the intended application, using these benchmarks eliminates many of the problems of the previously mentioned benchmarks. The benchmarks by definition preclude the use of "trick" optimizations (since any optimization that has an effect on one of the programs necessarily improves the performance of a "real" program). In principle, a vendor is supposed to quote the results of all ten benchmarks and the geometric mean of the ten results is the "SPECmark." Full descriptions of execution environments are supposed to be included with the stated results.

In practice, most of the "rules" seem to be followed, but in the never-ending quest for a single number, SPECmarks are often quoted out of context and used as a replacement for "MIPS." As machines with extreme floating point strength or weakness have been seen to distort the overall geometric mean, some vendors have begun to quote "SPEC integer," based on the geometric mean from the four integer applications alone, and "SPEC floating," based on the six floating point applications. This is not necessarily a bad practice, if you are more interested in specifics than in "overall" system performance.

> *"For systems with local disks, probably the most important performance factor after processor performance is the performance of the disk subsystem."*

What About System Performance ?

With the initial set of SPEC benchmarks, there is a reasonable standard for processor performance, including compilers, memory subsystems, etc. Not a perfect standard, not a universally accepted standard, ..., but a reasonable one. However, real usage of computer systems depends on many other performance factors: operating system overhead, storage subsystem bandwidth and latency, terminal capabilities, network subsystem characteristic, etc. Though there are many proposed synthetic benchmarks for most of these, there are very few real applications used as benchmarks of these factors, and there is relatively little consensus or consistency in the use of synthetic benchmarks.

There are many aspects of operating system overhead that could be measured, but some of the most commonly considered are: cost of a system call, cost of context-switching, and throughput of interprocess communication. Just as instruction mixes and synthetic processor benchmarks are of limited value in determining end user performance, these are not directly indicative of system performance. But these are of interest in assessing the efficiency of a particular implementation.

A common benchmark of UNIX system-call overhead is the time to run getpid(), since the usual implementation requires only a lookup in the "u block" and is dominated by the time to enter/exit kernel mode. Similarly, context switch time can be estimated by using a pair of processes which do nothing but interprocess communication (e.g., via a pipe) with minimal data passed. Each process of the pair sends a byte, say, to the other and then waits to receive a byte from the other.

Disk Subsystems

For systems with local disks, probably the most important performance factor after processor performance, or even the most important, period, is the performance of the disk subsystem (transfer rates, latency for random access, etc). There are numerous variables which can dominate performance: the disk drive(s) itself, the disk controller, the I/O bus (if there is one), device driver code, file system layout, and file system code. Partly as a result of these variables, there is no generally accepted benchmark of disk subsystem performance.

There are numerous simple synthetic benchmarks which create files, read files sequentially, read files randomly, etc. By varying the file sizes, using very large files, being careful to write distinct data to different parts of the files, reading alternate files to flush caches, etc., one can get a feel for disk subsystem performance. Some of the more ambitious synthetic disk benchmarks have been incorporated in commercial benchmarking packages or posted to Usenet. However, none of the popular benchmarks are based on real applications.

Multiuser Benchmarking

The status of network benchmarking is similar to that of disk benchmarking, and essentially the same approaches are often used. In some cases, exactly the same approaches are used; e.g., disk subsystem benchmarks are often used as benchmarks of remote file systems. Conversely, benchmarks originally designed for remote file systems are often used to assess performance of local disk subsystems.

There are two basic approaches to multiuser benchmarking. One is to attempt to create a synthetic workload of collections of processes, each collection attempting to represent one user. For example, such a collection might be a shell script which copies a file, uses grep or sed to scan the file (in lieu of an interactive editor), runs a compiler (or other utility) against the file, etc., all in a repeating loop. This is the easier approach to implement, but it is not fully convincing without rigorous arguments.

The other approach is to use one or more secondary computers as terminal emulators. For example, both the computer to be measured and the secondary computer are configured with multiport controllers connected to each other. The secondary machine runs a workload script against each port, while the computer being measured reacts to the workload as if it were driven by a live user. This approach has the advantage of being more intuitively representative of a real system, but is likely to be more expensive to implement.

In either case, the most important question is that of the workloads being used as the benchmark. The workloads can be real applications or synthetic applications. Unfortunately, the typical workloads are synthetic, there are no commonly accepted real workloads for these purposes.

Commercial Benchmark Packages and Services

There are a number of commercial enterprises which produce benchmark suites, collect results across different platforms, run customized benchmarks, etc. These include A&T Systems, AIM Technology, Neal Nelson and Associates, Performance Awareness, and ARS/Workstation Labs.

AIM Technology

AIM has produced several suites of synthetic benchmarks which represent the performance of various subsystems (processor, disk, operating system, ...), characterize multiuser systems, represent workstation applications, represent UNIX utility performance, etc. AIM also produces reports of results of their benchmarks on a variety of manufacturers' platforms.

Neal Nelson and Associates

Neal Nelson is probably best known for a series of "Business Benchmarks" which assess performance of systems on synthetic benchmarks oriented toward multiuser commercial applications. Neal Nelson is now emphasizing services based on terminal emulation equipment and associated benchmarks.

ARS/Workstation Labs

Workstation Labs emphasizes a benchmark suite known as "Khornerstone" which summarizes several aspects of system performance. The Khornerstone suite is based in part on well known benchmarks such as Dhrystone and Whetstone, but is also based on additional benchmarks assessing disk subsystems and multitasking capabilities. Workstation Labs also publishes monthly reports of results for a variety of manufacturers.

Next Steps

Benchmarking is inherently controversial because of the effects on buying decisions and competition amongst manufacturers. The best benchmarks are thus the ones that serve to minimize the controversy; e.g., by providing a direct link to real applications as in SPEC Release 1.0. However, in areas outside of processor performance, there is little agreement and much controversy remaining. Future articles will cover these subjects in more detail. ✦

Resources for Assessing System Performance

J.L. Hennessy and D.A. Patterson, *Computer Architecture: A Quantitative Approach*, Morgan Kaufman (1990).

H.J. Curnow and B.A. Wichman, "*A Synthetic Benchmark*," The Computer Journal 19, 1 (1976).

R.P.Walker, "*Dhrystone: A Synthetic Systems Programming Benchmark*," Communications of the ACM 27, 10 (October 1974).

J.Dongarra, "*Performance of Various Computers Using Standard Linear Equations in a Fortran Environment*," Argonne National Laboratories (1989).

J.Mashey, "*Performance Brief: CPU Benchmarks*," Issue 3.8 (June 1989).

SPEC Newsletter 2, 1 (Winter 1990).

J.K. Ousterhout, "*Why Aren't Operating Systems Getting Faster as Fast as Hardware?*", Usenix Summer Conference Proceedings, June 1990, pp.247-256.

A.Southerton, "*The Performance Measurement Contest*." Unix World (March 1990).

Benchmark Resources

Neal Nelson Business Benchmark, Multiple Language Business benchmark, and others.
Neal Nelson & Associates, 35 E. Wacker Drive, Suite 1510, Chicago, IL 60601 (312) 332-1462

AIM Application Benchmark, Multiuser Benchmark, Benchmark Suite 1. AIM Technology, 4699 Old Ironsides Dr., #150, Santa Clara, CA 95054 (408) 748-8649

System V Verification Suite, Release 3.0,
Contact Neal Kane, AT&T, 55 Corporate Drive, Room C02-24A14, Bridgewater, NJ 08807-6991 (201) 658-7695

C Test Suite (C Language Test Management Software), Contact Barb McLatchie, SCO, Canada, 130 Bloor St., West, 10th Floor, Toronto, Canada M5S 1M5 (416) 922-1937

Empower Remote Terminal Emulation Benchmark, Performix, Inc., 7927 Jones Branch Dr., #400, McLean, VA 22102 (703) 749-1452