Updated SPEC Benchmarks Released

SPEC92, New Multiprocessor Benchmarks Now Available

By Brian Case

In the late 1980s, a few industry representatives who wished to improve the state of computer system performance measurement and reporting, especially within the workstation industry, banded together to form SPEC (Standard Performance Evaluation Corporation). From its modest beginnings, SPEC has grown tremendously. While the original suite of 10 programs—now called SPEC release 1 and reported as SPECmark89—is still sold and supported, it is considered obsolete. SPEC now offers several standard benchmark suites with more in the planning stages. In this article, we review the evolution of the SPEC benchmarks and explore the more recent suites in detail.

The original SPEC benchmark suite (Release 1) was made available in October 1989. This suite consists of ten benchmarks: four integer-only and six floating-point-intensive programs. A serious effort was made to select nonsynthetic programs that were in everyday use. This led to the inclusion of programs like the GNU C compiler (gcc), a lisp interpreter (li), and a circuit simulator (spice).

The performance metrics reported by the original SPEC suite are the SPEC ReferenceTime, the SPECratio, and the SPECmark. There are ten reference times, where each is the time in seconds taken by a DEC VAX 11/780 to execute the corresponding benchmark. The ten SPECratios are the quotients of the reference time divided by the corresponding run time for the machine under test. The single SPECmark is the geometric mean of the ten ratios. (The geometric mean in this case is the 10th root of the product of the ten ratios.)

Since the SPEC suite was designed to allow workstations to be compared, it assumes the presence of UNIX and is only now beginning to be applied to PCs. In essence, SPECmarks were intended to replace VAX MIPS, which were popular for comparing workstation performance at the time. The problem with VAX MIPS was that the suite of programs used was not standardized and controlled well enough. An independent organization like SPEC solves that problem.

In December 1991, the integer and floating-point areas of the release-1 suite were officially sanctioned as separate metrics and named SPECint and SPECfp. The computation of these metrics is the same as the SPECmark; e.g., SPECint uses four reference times, four ratios, and the fourth root of the product of the four ratios. At an announcement in January 1992, all the original metrics were officially renamed to include the "89" suffix (SPECint89, SPECfp89, and SPECmark89) to make room for newer suites and metrics with similar names.

One of the problems with Release 1 was that one of the floating-point benchmarks, matrix300, was a small but long-running kernel that was sped up by cache blocking algorithms applied by the KAP preprocessor for FORTRAN programs. For those who used KAP, this one benchmark SPECratio could be improved by a factor of five or more, which resulted in a significantly higher SPECmark. While this kind of optmization is legitimate and beneficial in many environments—cache blocking is an important improvement in compiler optimization technology—matrix300 carried too much weight in the SPECmark89 metric, which resulted in an over-estimation of the effect of cache blocking on real world applications. To solve this and other problems, SPEC introduced new benchmark suites to replace Release 1.

An alternative to SPEC for measuring PC performance is offered by the BAPCo (Business Applications Performance Corporation, which is an organization modeled on but separate from SPEC) benchmark. Like SPEC, BAPCo is an attempt to use real, sizeable programs as benchmarks, but unlike SPEC, BAPCo distributes the benchmarks in binary instead of source. Distributing binary is possible in the PC market since all PCs have the same processor architecture. SPEC will become more important for PCs as they become stronger competitors to UNIX workstations.

Price & Availability

The SPEC92 benchmarks are available in source-code form on a QIC-24 tape. The integer suite costs $425, and the floating-point suite costs $575; both together cost $900. The SDM Release 1.1 suite costs $1450, and SPEC89 Release 1.2b costs $300.

To get the official SPEC benchmark results, subscribe to the SPEC newsletter. Published quarterly, the newsletter costs $550 per year, and it includes a one-page summary of the system configuration and benchmark results for each system for which the vendor has submitted results in the preceding quarter. Back issues are available for $25 each through Winter 1991, and $150 each thereafter.

Contact Diane Dean, SPEC Administrator, NCGA, 2722 Merrilee Dr., Suite 200, Fairfax, VA 22031; 703/698-9600. ext. 318; fax 703/560-2752.

SPEC92 Benchmark Descriptions

Integer Programs

Espresso is an EDA tool that generates and optimizes PLA structures. Espresso is written in C and is relatively small. The execution is characterized by loops, array manipulation, and storage allocation. The benchmark input consists of four different input models. Espresso is sensitive to cache size.

Li is a LISP interpreter written in C. The benchmark is a solution to the nine queens problem. The backtracking algorithm of this problem is recursive, which makes it a challenge for register window architectures (because of the rapidly varying stack depth) and register allocation in non-window architectures (because of the lack of leaf procedures) .

Eqntott translates Boolean equations into truth tables. Written in C, the primary operation performed is sorting on relatively large data sets, which can stress data caches on some machines.

Compress is a data compression application that uses Lempel-Ziv coding. This benchmark compresses and uncompresses a 1 MB file 20 times. While the benchmark is CPU intensive, it also performs measurable amounts of VO.

Sc is a spreadsheet benchmark based on a program originally written by James Gosling. Even though it is a spreadsheet, this C program performs mainly integer calculations. The spreadsheet display depends on the UNIX cursor-control ("curses") package to treat a dumb terminal as an X/Y-addressible character display. Since output is directed to a file, tuning the curses package is one way to improve performance on this benchmark.

Gcc is the GNU C compiler version 1.35 from the Free Software Foundation. Like compress, gcc performs measurable amounts of I/O; about 10% of gcc time is usually spent in I/O. This version of gcc produces 68020 (Sun-3) code, and the benchmark consists of processing 76 input files. Gcc performance is improved by large caches and fast I/O devices.

Floating-Point Programs

Spice is an analog circuit simulator written in FORTRAN with a UNIX interface written in C. Floating-point computations are performed in double precision. The benchmark processes five copies of the gray-code circuit. The benchmark causes high data cache miss rates on most machines .

Doduc, a kernel taken from an original application written by Nhuan Doduc, is a Monte Carlo simulation of the time evolution of a thermo-hydraulical model for a nuclear reactor's component. This FORTRAN program performs double-precision floating-point and is non-vectorizable. Execution is characterized by little I/O, short branches and loops, and many subroutines.

Mdljdp2 "...solves the equations of motion for a model of 500 atoms interacting through the idealized Lennard-Jones potential." Nuff said. This is a double-precision FORTRAN program. An input file provides the density and temperature for the model.

Wave5 solves Maxwell's equations and particle equations of motion. This large FORTRAN program uses single-precision arithmetic, and the input data calls for 500 K particles on a grid with 50 K points for five time steps (SPEC reduced the time steps from 20 to five).

Tomcatv is a highly vectorizable FORTRAN program used to analyze general geometric domains such as airfoils and cars. Tomcatv favors superscalar and vector processors and results in high data-cache miss rates on most machines.

Ora traces rays through optics consisting of spheres and plane surfaces. This FORTRAN program uses double precision arithmetic.

Alvinn is a single-precision C program that trains a neural network called ALVINN using back propagation to keep a vehicle from driving off of a road. The neural net model has 1220 input units for a video camera and a laser range finder, 32 output units representing the direction control from the network, and 30 hidden units. The neural network is fully connected.

Ear is a single-precision C program that uses FFTs and other library routines to simulate the human ear. The original source is from Apple Computer.

Mdljsp2 is the single-precision version of mdljdp2.

Swm256 is a single-precision FORTRAN program that solves a system of shallow water equations using finite difference approximations on a 256×256 grid. This benchmark spends about 48% of its time executing one of its subroutines (calc2).

Su2cor is a vectorizable, double-precision FORTRAN program used in quantum physics. A pre-processor can effect a vectorization factor of 15.0 with a vectorization degree of 98.5%.

Hydro2d is a vectorizable, double-precision FORTRAN program using Navier Stokes equations to compute galactical jets. This program has a vectorization factor of 8.8 and a vectorization degree of 99.5%.

Nasa7 is a double-precision FORTRAN program. Changes have been made from the original SPEC release 1 version: input data comes from a file; the source has been split into two files; output data shows individual results for seven kernels to help the benchmark jockey understand and improve the performance of the individual kernels.

Fpppp is a double-precision, FORTRAN program from quantum chemistry. This program is difficult to vectorize because it contains very large basic blocks. This version solves a larger problem than the SPEC89 fpppp through a change to the input file.

Machine Name CPU External Cache	SPECint92	SPECfp92	SPECint89	SPECfp89	SPECmark89
Silicon Graphics CRIMSON 50 MHz R4000 8K I-cache + 8K D-cache 1M I+D secondary cache	58.3 (-4%)	61.5 (-21%)	60.6	77.8	70.4
HP 9000 Series 700 Model 750 66 MHz PA-RISC 1.1/FPU 256K I -cache + 256K D-cache	48.1 (-6%)	75.0 (-26%)	51.5	101.6	77.5
IBM RS/6000 POWERstation 560 50 MHz POWER 5064 8K I-cache + 64K D-cache	42.1 (-4%)	85.5 (-40%)	43.8	143.5	89.3
HP 9000 Series 700 Model 710 50 MHz PA-RISC 1.1./FPU 32K I-cache + 64K D-cache	31.6 (-11%)	47.6 (-23%)	35.4	62.4	49.7
Intel XPRESS Desktop 486DX50 50 MHz 80486DX, 8K I+D cache 256K I+D secondary cache	30.1 (+6%)	14.0 (-23%)	28.5	18.3	21.9
DECstation 5000 Model 240 40 MHz R3000A/R3010A 64K I-cache + 64K D-cache	27.3 (-2%)	29.9 (-16%)	27.9	35.8	32.4
Sun SPARCstation 2 40 MHz CY7C601/TI390C602A 64K I+D cache	21.8 (+0%)	22.7 (-17%)	21.7	27.4	25.0

Table 1. Comparison of SPEC89 and SPEC92 results (differences from SPEC89 in parentheses).

CINT92, CFP92, SPECint92, and SPECfp92

In January 1992, SPEC released the larger and improved CPU measurement suites CINT92 and CFP92. The corresponding metrics, SPECint92 and SPECfp92, are now the preferred method of reporting raw speed.

The 'C' in CINT and CFP identifies these suites as component benchmarks (processor intensive) as opposed to system benchmarks (processor, VO, bus, memory, etc.). The CINT92 suite contains six benchmark programs—two more than for SPEC release 1—and the CFP92 suite contains fourteen benchmark programs, eight more than in the original release.

The program suites have increased in size to include more application areas. SPEC says CPF92 is larger than CINT92 because floating point programs come in two flavors—single and double precision—and floating-point benchmarks show substantially more variability in performance than do integer benchmarks. A broader spectrum of floating-point benchmarks will better characterize the strengths and weaknesses of each machine. Table 1 shows the SPEC92 and SPEC89 results for selected high-performance machines.

CINT92 contains three of the original integer programs from release 1, and the fourth, gcc, is replaced with a longer-running version. The two new benchmarks in CINT92 are a Lempel-Ziv compression program and a UNIX dumb-terminal-based spreadsheet. All integer programs are written in C.

CFP92 includes five of the six original release-1 floating-point programs plus nine new programs representing new application areas such as quantum chemistry, quantum physics, optics, and weather prediction. Twelve of the benchmarks are written in FORTRAN with the remaining two coded in C. Nine use double-precision arithmetic while five use single-precision. (See the sidebar for more details about the benchmarks.)

Benchmark	Lines	Code KB	Data KB
espresso	14838	208	64
li	7741	176	56
eqntott	3454	80	313
compress	1503	56	450
sc	8485	208	131
gcc	87791	736	144
spice2g6	18912	480	7939
doduc	5334	272	134
mdljdp2	4456	224	230
wave5	15062	344	14288
tomcatv	184	160	3636
ora	533	168	30
alvinn	272	88	490
ear	5237	144	65
mdljsp2	3883	224	191
swm256	487	184	3638
su2cor	2514	248	4117
hydro2d	4448	224	327
nasa7	1177	216	2842
fpppp	2718	232	330

Table 2. Sizes of SPEC92 benchmarks.

The matrix300 benchmark is conspicuous by its absence from CFP92, and SPEC gives several reasons for dropping it. First, hardware and software (compilers) have improved so much that matrix300 ran in less than six seconds on some machines. The primary culprit is the KAP FORTRAN preprocessor from Kuck & Associates. KAP has the ability to change the way a program accesses arrays and matrices: computations are restructured to improve cache access patterns. Matrix300 also had the problem that input data came from constants in the program text; consequently, an aggressive compiler could propagate constants and eliminate most of the work in the benchmark, which is unrealistic.

The application area addressed by matrix300 is still represented since some of the other floating-point benchmarks perform matrix multiplication. To address the over-optimization problem, input data for all programs now comes from disk files.

As shown in Table 1, the differences in the integer performance ratings between the SPEC89 and SPEC92 suites is relatively small when compared to the differences in floating-point performance. Sun appears to benefit the most from the new suites. The SPARCstation2 lost no ground in integer performance and only 17% in floating-point. Intel says the SPECint92 for the Intel machine is higher than the its SPECint89 because the two new benchmarks, sc and compress, perform well with the Metaware compiler and x86 architecture (the same compiler, Metaware 2.4b, was used in each case). IBM's POWERstation 560 fared the worst in floating-point, with its SPECfp92 being 40% lower than its SPECfp89, while the HP9000 fared the worst in integer, with a SPECint92 11% lower than its SPECint89.

Despite the improvements incorporated into these two new suites, it is interesting to note that the benchmark programs still do not stress instruction caches. According to an analysis performed at HP, all of these programs exhibited small miss rates with a 32 KB instruction cache, and all have miss rates of less than 1%

with a 128 KB instruction cache.

One interpretation of this information is that these programs contain undersized "hotspot" kernels, but another is that the old "80/20" rule (programs spend 80% of their time in 20% of their code) is accurate for real programs. Table 2 shows the number of lines of source code, object code sizes, and static data sizes for all 20 SPEC92 benchmarks. As is clear from the table, twenty percent of most of the benchmarks will nearly fit in a 32 KB I-cache.

SPEC92 appears to be an improvement over SPEC89. The increase in the number of benchmarks and the lower overall sensitivity to a single optimization technique make SPECint92 and SPECfp92 more indicative of realizable performance. As always, SPEC encourages the use of the individual SPECratios when making comparisons, and the larger number of benchmarks gives each machine type more of an opportunity to shine in one particular application area.

While SPEC92 results are frequently quoted now, SPEC89 numbers are still very popular among vendors because they are usually higher. SPEC92 has been available long enough that vendors should be publishing these numbers exclusively, but until SPEC92 completely replaces SPEC89, the phrase "buyer beware" applies.

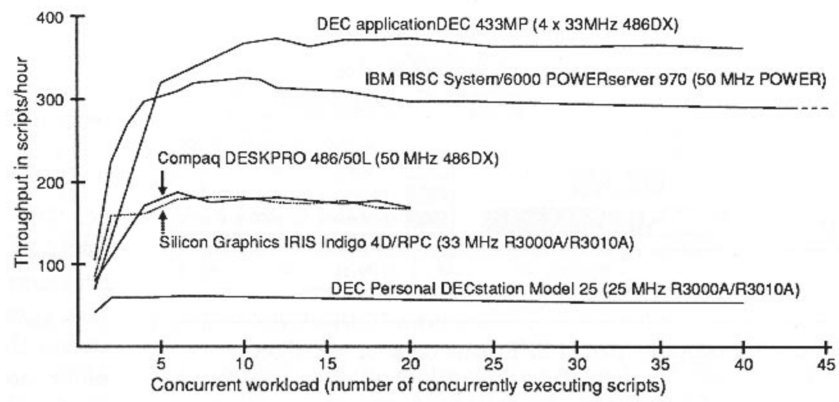

Figure 1. Plot of sdet throughput as a function of workload.

Multiprocessor Benchmarks

In 1990, a new metric, SPECthruput, was announced as part of SPEC release 1.2b. This metric was an attempt to measure the available throughput of a system, instead of just the raw CPU speed. In particular, this metric was intended to be useful for multiprocessors and was supposed to measure per-processor speed.

This metric is computed in the same way as the SPECmark except that the reference time and machine-under-test times were measured for the completion of two concurrently running copies per processor of each program. The SPECthruput was then reported as "number of cpus@geomean." For example, if the test machine had four CPUs and the geometric mean of its thruput ratios was 10, the machine's SPECthruput was 4@10. Its Aggregate Thruput metric was 40.

SPECthruput was confusing and invited erroneous comparisons between SPECthruputs and SPECmarks. SPEC committee discussions eventually led to the creation of the Homogeneous Capacity Method to replace SPECthruput and System Development Multitasking to provide another angle for evaluating system performance.

Essentially, there are two environments that could benefit from the advantages of multiprocessors: interactive and CPU-bound. System Development Multitasking measures how well a multiprocessor system handles interactive task loads, while the Homogeneous Capacity Method rates how much benefit a multiprocessor system provides to CPU-bound task loads.

Interactive Response

In May 1991, SPEC released the System Development Multitasking (SDM) benchmark suite to replace its SPEC Thruput Method A (an updated SDM 1.1 was released in 1992). SDM has the major advantage of applying a variable load to a system. It produces a response curve that can be used to draw conclusions about system bottlenecks.

The SDM 1 suite contains two benchmarks, sdet and kenbusl, that model activity in development environments. Sdet and kenbus1 are essentially scripts that invoke sequences of UNlX commands. Sdet appears to be more of a commercial, text-processing simulator as it includes 26 different UNIX commands including spell and nroff. Kenbus1 is more of a software-development simulator since it excludes some of the text processing commands. Kenbus1 has the additional characteristic of simulated keyboard entry: commands are entered from the script at a three-character-per-second rate.

Throughput is the performance metric for each of these benchmarks and is defined as scripts/hour. A complete characterization of a system's performance is shown by the throughput response curve, which is plotted as scripts/hour vs. number of concurrently executing scripts.

The number of concurrent scripts is increased until at least one of two termination conditions is reached: (1) system throughput degrades to 70% of the peak thruput value, or (2) the system is subjected to two times the workload that produced the peak throughput value. Note that a vendor is free to run the benchmark beyond these termination conditions and provide the additional data for publication. Since kenbus1 employs a "think-time" of 1/3-second per character, kenbus1 throughputs are much higher than those for sdet.

For the purposes of analysis, the SDM 1 curves can

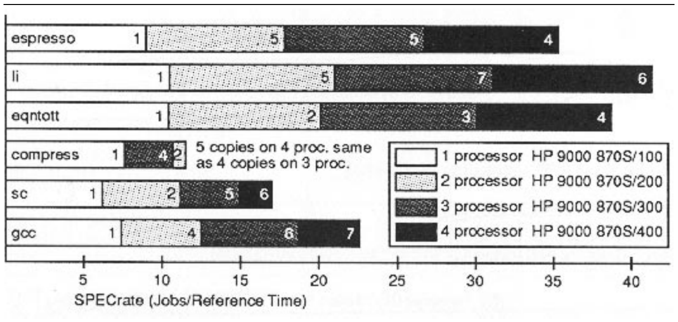

Figure 2. Histogram plot of SPECrate data for HP 870S.

be broken into three regions: rising, flat, and declining. A steep slope in the rising region with a peak at low loads indicates either that the machine has a very fast CPU and disk combination or that no actual disk I/O is being done. RAM disks and other caching techniques can cause low I/O overhead.

A long flat region indicates that the system is able to sustain increased loads gracefully. A lot of main memory helps to create a long flat region since more jobs can be kept in memory, leaving all disk bandwidth available for actual data I/O requests.

As the number of concurrently executing scripts increases, the CPU spends an increasing percentage of its time simply scheduling processes and allocating resources instead of performing useful work. The declining section of the curve is indicative of this overhead. If the workload is increased far enough, the system starts paging or swapping heavily and throughput drops precipitously.

Figure 1 shows a plot of sdet curves for a few machines. The highest curve on this graph is for a four-processor 486-based machine from DEC. The difference between the initial slope and peak for this machine and the initial slope and peak for the single-processor 486 machine proves that sdet shows off the capabilities of multiprocessors. The DEC multiprocessor has the added advantage of three disk controllers, while the Compaq has only one. This helps keep the multiple processors busy.

These curves also show that system components other than the processor and cache affect throughput. Compared to the Deskpro, the IRIS Indigo curve has a very steep initial slope, which indicates a faster processor. When more than two jobs are running concurrently, however, some other system resource, presumably disk I/O, becomes the bottleneck. Also, note that the IRIS has three times the thruput of the DECstation Model 25 even though the processor is only 33% faster. The lesson is a simple one: in certain environments, more memory and a faster disk system can be a better investment than a faster processor.

Most SDM curves do not have a declining section because the two termination rules end the benchmark before most multiprocessor systems become really overloaded. Most machines in this class have very large main memories (some machines for which SDET results have been reported have as little as 16 MB but most have at least 64 MB, and some have hundreds of MB).

Raw Multiprocessor Horsepower

The component measures—CINT92 and CFP92—measure how fast a system can execute a single copy of a program. SDM 1 is a throughput benchmark designed to test system performance in the wholistic sense. In between these two characteristics is room for at least one more: how fast a system can complete a few CPU-bound jobs, i.e., the total CPU capacity of a system.

The idea is to answer the question "How much CPU horsepower is available?" While the SDM benchmark measures how well a multiprocessor system manages an increasing load of interactive jobs, the Homogeneous Capacity Method (HCM) measures how well a multiprocessor handles a few compute-intensive jobs. HCM uses the programs in CINT92 and CFP92 and provides a meaningful way to compare systems with different numbers of processors for compute-bound jobs. The basic metric for the HCM is the SPECrate, and the final summary metrics are SPECrate_int92 and SPECrate_fp92.

The benchmarking procedure is to run a number of copies of each benchmark concurrently on a system; the number of copies to run is chosen by the benchmarker in a deliberate attempt to maximize the SPECrate for that benchmark. For a single processor system, the number of copies is often, but not always, one, while for multiprocessor systems, the number of copies depends on the benchmark and the characteristics of the system.

Once the optimal number of copies is determined, the time taken to complete all copies is measured. These two numbers, copies run and elapsed time, are used in the following equation:

$$\text{SPECrate} = \frac{\text{\#copies\_run} \times \text{reference\_factor} \times \text{unit\_time}}{\text{elapsed\_execution\_time}}$$

The reference_factor is a normalization term. On the VAX 11/780, the longest-running-benchmark is the floating-point benchmark ear. The reference_factor for a particular benchmark is its VAX running time divided by the VAX running time for "ear." Thus, reference_factor compensates for fast-running benchmarks and makes all the SPECrates fall into a narrower range than they would without reference_factor.

The unit_time term is used to scale the results to a longer time interval, i.e., it makes the SPECrates appear larger. The time interval is arbitrarily chosen to be one week since this is "...the smallest time interval within which the SPEC reference machine (the venerable VAX

11/780) can complete a significant number of jobs." Thus, unit time is 604800 seconds (one week).

Once SPECrates are computed for the individual benchmarks, the geometric mean is taken to arrive at the overall SPECrate_int92 or SPECrate_fp92.

Table 3 shows some SPECrate_int92 data for the HP 870S series of PA-RISC machines. Each CPU is a 50-MHz PA-RISC CPU and FPU with 512 KB of I-cache and 512 KB of D-cache. Figure 2 shows a histogram plot of this data. The shading of the bar indicates the number of processors, while the number in the bar indicates the number of copies used to achieve the corresponding SPECrate.

Benchmark	1 Processor			2 Processors			3 Processors			4 Processors		
	#copies	Time (sec)	SPEC rate	#copies	Time (sec)	SPEC rate	#copies	Time (sec)	SPEC rate	#copies	Time (sec)	SPEC rate
espresso	1	60.0	897.3	5	151.0	1782.8	5	101.0	2665.3	4	61.0	3530.4
li	1	140.0	1052.0	5	351.0	2098.1	7	331.0	3114.8	6	214.0	4129.5
eqntott	1	25.0	1043.6	2	26.0	2006.9	3	26.0	3010.3	4	27.0	3865.1
compress	1	87.0	755.1	2	114.0	1152.6	4	245.0	1072.6	5	306.0	1073.5
sc	1	173.0	621.0	2	192.0	1119.2	5	361.0	1488.1	6	380.0	1696.4
gcc	1	176.0	735.8	4	415.0	1248.2	6	416.0	1867.8	7	401.0	2260.6
SPECrate_int92			835.4			1514.9			2051.3			2478.8

Table 3. SPECrate data for the HP 870S PA-RISC machine.

The espresso, li, and eqntott benchmarks scale almost linearly with the number of processors, but compress, sc, and gcc show less scalability. As a result, the overall SPECrate_int92 values scale well, indicating that computing capacity does increase with each added processor, but the compress benchmark prevents the overall rating from scaling linearly.

The peculiarity of the data for compress results from its poor cache behavior. In a condition of high cache-miss rates, a multiprocessor will perform poorly because the memory-to-cache bus becomes completely saturated with cache line traffic. Even a single processor and cache can saturate a bus if the miss rate is high enough.

SPEC Reporting Rules

SPEC uses a standard layout when publishing its benchmark results. Each one-page summary includes a table listing all relevant data for the particular benchmark suite; in particular, the individual SPECratios or SPECrates are always listed so that any anomalies, such as the high matrix300 numbers, are clearly visible instead of being buried in a composite result (i.e., SPECmark). SPEC always recommends looking at the individual benchmark results rather than the single geometric mean.

Next to the table is a hardware description that attempts to reveal the performance-influencing characteristics of each machine. These characteristics include processor model, clock rate, cache sizes (but not associativity), main memory size, disk subsystem characteristics including number and kind of controllers and number and kind of drives, OS and compiler types and versions, other important software (e.g., vectorizing preprocessors), tuning parameters, background load, and system state (e.g., number of active logins).

The bottom half of the page contains a graphic display of the performance data contained in the table. A histogram is used for SPECint92, SPECfp92, and the SPECrates, while a line-segment curve is used for SDM data. Finally, a list of notes discloses specific compiler flags used to compile the benchmarks and tuning parameters used to tweak the operating system.

Futures

SPEC and the LADDIS (Legato Systems, Auspex, Data General, DEC, Interphase, and Sun) group are actively working toward a standard network benchmark. With PRE-LADDIS 0.1.0 released in February 1992, LADDIS (formerly known as NHFStone) will be SPEC's first System-level File Server (SFS) benchmark suite. Siemens-Nixdorf has offered SPEC a network benchmark program, which will have the advantage of DIN-standard conformance. For the longer term, SPEC is always looking for new benchmarks and welcomes submissions and comments from the field.

One problem for SPEC is the complex nature of the multiprocessor benchmarks. In the past, SPEC has had the advantage of being simple and easy to understand. While quoting the composite SPECmark89 created problems (e.g., hiding the effect of matrix300), it at least provided a clear starting point for comparing systems. With the sdet and HCM suites, a multiprocessor vendor is hard pressed to find one or two numbers to include in press releases and product literature. Instead, unintuitive tables or complex graphs are required.

Clearly, SPEC has become an important, respected force in the workstation industry. Where the first question about a new workstation used to be "How many Dhrystones does it do?," now the question is 'What is its SPECmark?"

As SPEC suites improve and encompass new areas of measurement, users and vendors alike will benefit from the ability to make informed decisions. Users can make better-educated purchasing decisions, while microprocessor designers can more objectively evaluate proposed architectures and implementation techniques by using simulations of the SPEC benchmarks to measure their effects on performance. ♦

