SPEC95 Retires SPEC92

Test Suite Has New Benchmarks and Faster Reference Machine

by Brian Case

The System Performance Evaluation Corporation (SPEC) has announced the new SPEC95 benchmark suites to replace the previous suites, SPEC92, which were released three years ago. SPEC95 encompasses a sweeping set of changes that address concerns about compiler attacks on benchmarks; small, cache-resident programs; and an out-of-date reference machine. The SPEC95 suite will be available in October.

In the late 1980s, SPEC grew out of the efforts of a few Unix system vendors that saw the need to improve the state of system performance measurement, particularly in the realm of Unix systems and workstations. The first SPEC benchmark suite was released in October 1989; known initially as "Release 1," it was later renamed SPEC89, and the single performance metric was called SPECmark89.

Three years later, in January 1992, as the original benchmarks began to show signs of age, a major update of the SPEC suite was released as SPEC92 *(see **061203.PDF**)*. The integer benchmarks remained largely the same, but the floating-point suite was expanded from 6 programs to 14 in an effort to represent both C and FORTRAN languages and both single- and double-precision computations.

Along the way, the suites of integer and floating-point benchmarks were formally named CINT92 and CFP92 (the "C" in their names comes from component benchmarks vs. system benchmarks), respectively, with corresponding performance metrics named SPECint92 and SPECfp92.

Motivation for SPEC95

A replacement for SPEC92 is needed because software and hardware technology have advanced significantly over the past three years. In 1992, using the VAX 11/780 as the reference machine seemed reasonable if perhaps stubborn. At the time, what is now a midrange 486 ruled the high end of the PC world, the first Pentium chips were still prototypes in Intel's labs, and the fastest RISC workstations outclassed the VAX by a factor of "only" 40 to 60 (SGI's Crimson R4000 machine scored a 58.3 on SPECint92). Today's top machines are 300–500 times faster than the VAX.

In addition, compiler technology has brought some kernel-based benchmarks (benchmarks that run one or a few small loops for a long period of time) under "attack." Compiler writers are crafting more clever compilers, and faster machines have allowed those compilers to do more analysis while maintaining acceptable compile times. SPEC had to grapple with this problem for the first time when the Kuck & Associates FORTRAN preprocessor (KAP) was used by some companies to drive matrix300 performance through the roof, thereby improving SPECfp92 results for those with access to KAP *(see **061203.PDF**)*. The KAP-enhanced results were actually relevant to some sites that do lots of matrix300-like work, but SPEC's emphasis on common application performance led it to abandon matrix300.

Problem benchmarks in the CINT92 suite were eqntott, sc, and espresso. On the fastest current machines, the run times for these benchmarks are under 10 seconds. For eqntott, with a run time of three seconds, for example, a small measurement error of 0.25 or 0.5 seconds can alter the SPECratio dramatically. Furthermore, these benchmarks came under compiler attack. When SPEC was unable to find input models to increase run times to reasonable levels, these benchmarks were abandoned. Simply running these benchmarks 100 times does not make them usable because a program run repeatedly with a small input model usually exhibits atypical cache behavior.

A New Reference Machine for a New Era

SPEC95 results are normalized to a new reference machine—a SparcStation 10/40 with a 40-MHz SuperSparc, no level-2 cache, and 64M of main memory (C compilation for reference times uses SC3.0.1 with compiler flags "–fast –xO4"). SPEC says it decided to choose a reference machine without level-2 cache because not all machines come so equipped. It is also best to have a reference machine that will beat few, if any, of the machines being benchmarked.

A complete SPEC95 trial on the reference SparcStation—which requires at least three iterations of each benchmark (see below)—takes 48 hours. A second trial is needed to generate baseline results, and attempts to discover the optimum configuration of compiler flags require even more runs.

To get a ballpark estimate—really only a guess—of SPEC95 results for a given machine that already has SPEC92 results, the SPEC92 values for the reference machine shown in Table 1 can be used as a scaling factor—divide current SPEC92 results for your machine by the numbers in Table 1. Of course, since the programs and data sets in the suites are different, there is no simple way to convert between SPEC92 and SPEC95

Suite	Benchmark	Elapsed Time (sec.)	SPECratio
CINT92	008.espresso	64.1	35.41
	022.li	170.0	36.53
	023.eqntott	17.6	62.50
	026.compress	124.1	22.32
	072.sc	53.5	84.67
	085.gcc	169.2	32.27
SPECint92			41.26
CFP92	013.spice2g6	1053.4	22.78
	015.doduc	61.8	30.10
	034.mdljdp2	144.9	48.93
	039.wave5	197.5	18.73
	047.tomcatv	69.5	38.13
	048.ora	62.4	118.91
	052.alvinn	92.9	82.78
	056.ear	446.9	57.06
	077.mdljsp2	122.9	27.26
	078.swm256	1149.4	11.05
	089.su2cor	495.1	26.06
	090.hydro2d	295.2	46.41
	093.nasa7	712.5	23.58
	094.fpppp	334.0	27.54
SPECfp92			34.35

Table 1. SPEC92 results for the SPEC95 reference machine, a SparcStation 10/40 (40-MHz SuperSparc microprocessor with no level-2 cache). (Source: SPEC)

metrics; each machine/compiler combination has a unique performance profile.

Using Table 1's metrics as scaling factors can predict ballpark SPEC95 numbers, though. For example, assuming a machine has a SPECint92 of 100, we could expect to obtain a SPECint95 for this machine somewhere in the neighborhood of 2.0 to 3.0. For those used to dealing with high SPEC92 scores, SPEC95 results are going to seem disappointingly small at first—SPEC95 is no longer an approximation of VAX MIPS.

Criterion	Desired Characteristics
Memory activity	Should have more memory activity than SPEC92
Running time	Should take into account future performance
Ease of attack by compiler	Should resist special optimization tricks
Working set size	Should be big but not too big (64 MB memory assumed)
Portability	Easy portability required
Programming style	Should be representative of typical style (e.g., modular)
Programming language	C for integer, FORTRAN for floating-point
Application vs. kernel	Reject kernels in favor of complete applications
Robustness	Prefer debugged code
Input workloads	Must be able to construct test, train, and reference workloads

Table 2. Selection criteria for SPEC95 benchmarks (Source: SPEC).

The benchmark programs in SPEC95 were chosen to "fit" in 64M or less of main memory, so as to hit the common denominator of machines likely to be benchmarked with SPEC. In the next three years, common systems will have 64M of memory, although SPEC acknowledges that servers and high-end workstations will have much more (in fact, they already do). With benchmarks and input workloads that fit in 64M of memory, the effects of I/O performance due to swapping and paging are less likely to affect benchmark performance.

SPEC95 Characteristics

As before, the SPEC95 CPU benchmarks consist of two suites, CINT95 and CFP95, and the corresponding performance metrics are SPECint95, SPECbase_int95, SPECfp95, and SPECbase_fp95. The two groups of programs are referred to as component-level benchmark suites because they test the CPU, caches, memory, and compiler (but not the I/O system). In contrast to a couple of the earlier benchmark programs, an explicit design goal for the new suites is to make the effects of I/O, network, display, and operating-system performance negligible on SPEC95 benchmark results. The SPEC benchmarks are for evaluating the core of a system; other elements are deliberately not measured, even though they are extremely important in many applications.

The SPEC committee used objective evaluations of program characteristics and data gathered from hardware performance monitors to evaluate candidate benchmarks for SPEC95. Experience with the deficiencies of past versions of the suites led the SPEC committee to develop the list of criteria shown in Table 2. The initial offering from SPEC will be for Unix systems, but one of the goals for SPEC95 is increased portability. SPEC member companies have indicated that the benchmark programs are portable to various flavors of Unix, Windows NT, and OpenVMS. Releases for these other OS's will follow when sufficient demand materializes. The use of SPEC95 for comparing PC systems might increase dramatically if a Windows NT version were available.

Table 3 lists the individual benchmark programs in the two SPEC95 suites. The new integer suite is 33% larger than CINT92. Three benchmarks from CINT92 are part of the new suite, but they have been significantly modified in both source code and workload to increase their running time.

The new floating-point suite is smaller than CFP92 by four benchmarks. Five of the CFP92 benchmarks are included in CFP95, but, as with the integer suite carryovers, all were significantly modified in both source code and workload. The changes in suite sizes—more programs in the integer suite, fewer in the FP suite—reflect a greater emphasis on measuring integer performance.

In contrast to the previous floating-point suite, CFP95 consists exclusively of FORTRAN programs. The

SPEC committee decided to stick with FORTRAN for several reasons. First, work on optimizing FORTRAN compilers for FP calculations has been in progress for 40 years; C compiler writers have only recently begun paying serious attention to the quality of FP code. Second, since many sites are interested only in either integer or FP performance, SPEC wished to avoid requiring such sites to have the highest-quality versions of both compilers. Also, the baseline results require the same compiler flags for all benchmarks (see below); since a significant number of programs are required to make the baseline measurements meaningful for a single compiler configuration, the size of the FP suite would have been considerably larger. A larger suite would have only made the long run time even longer. When a need for C-language FP benchmarks is demonstrated, SPEC will consider defining a separate suite.

New Run Rules

For both CINT95 and CFP95, a valid performance result requires running each benchmark a minimum of three times. The median time of all runs of a benchmark is the time for that benchmark. SPEC says this method of gathering run times ensures that a "typical" value for run time is used instead of a "guaranteed-not-to-exceed" or an unusually slow time. Also, to prevent the impression that results have more precision than is actually measurable, the reporting rules require that only three significant digits be reported.

For SPEC95, the reference times generated by a SparcStation 10/40 for the 18 benchmarks are listed in Table 3. The reference times have been deliberately rounded to the nearest 100 seconds. Note that even the fastest-running benchmark takes over 20 minutes to complete on the reference machine. These run times preclude running SPEC on architectural simulators.

As before, the official running time for a benchmark on a test machine (the median of at least three runs) is divided into the SparcStation reference time to arrive at a SPECratio for that benchmark. Also as before, the overall performance metrics are computed as the geometric mean of the ratios for the benchmarks in a suite. For example, the SPECint95 metric is computed as the eighth root of the product of the eight ratios for the benchmarks in the integer suite.

To further enhance the validity of the SPEC metrics and highlight any atypical results, SPEC requires that a full disclosure include "baseline" metrics as well as the standard SPECint95 and SPECfp95 metrics. The baseline metrics—SPECbase_int95 and SPECbase_fp95—are produced by the standard benchmarking procedure, except that the compilation of the benchmarks is allowed to use no more than four optimization flags, and the same four flags—in the same order on the compiler command line—must be used for all compilations in a suite. Assertion flags are not allowed. SPEC prefers these flags to be the ones recommended by the compiler vendor. Profile-driven recompilation using the training input model (see below) is allowed as long as it is fully automated and the four-flags rule is obeyed.

One of the chief goals of SPEC is to model real-world application performance. Unfortunately, there is significant variability in the efforts by developers to optimize code for speed. Some use every tool available and

Suite	Benchmark	SPEC92 Name	Lines Of Code	SPEC95 Reference Time (seconds)	Comments
CINT95	099.go	—	29246	4600	Artificial intelligence; plays a game of go
	124.m88ksim	—	19915	1900	Motorola 88000 CPU simulator
	126.gcc	085.gcc	205085	1700	New version of gcc; builds SPARC code
	129.compress	026.compress	1934	1800	Compresses and uncompresses file in memory
	130.li	022.li	7597	1900	LISP interpreter with new workload
	132.ijpeg	—	31249	2400	Graphic JPEG compression and decompression
	134.perl	—	26871	1900	Perl language interpreter; solves a puzzle
	147.vortex	—	67202	2700	Object-oriented database management system
CFP95	101.tomcat	047.tomcatv	190	3700	Mesh-generation program
	102.swim	078.swm256	429	8600	Shallow water model with 1024 × 1024 grid (single precision)
	103.su2cor	—	2332	1400	Quantum physics; Monte Carlo simulation
	104.hydro2	090.hydro2d	4292	2400	Astrophysics; hydrodynamical Navier-Stokes equations
	107.mgrid	—	666	2500	Multigrid solver in 3D potential field
	110.applu	—	3868	2200	Parabolic/elliptic partial differential equations
	125.turb3d	—	2100	4100	Simulate isotropic, homogeneous turbulence in a cube
	141.aspi	—	7361	2100	Solve temperature and wind velocity and distribution of pollutants
	145.fpppp	094.fpppp	2784	9600	Quantum chemistry
	146.wave5	039.wave5	7764	3000	Plasma physics; electromagnetic particle simulation

Table 3. Composition of the SPEC95 benchmark suites. All integer benchmarks are C-language programs; all floating-point benchmarks are FORTRAN programs. CINT95 grew to eight benchmarks from six in CINT92. CFP95 has four fewer benchmarks than CFP92. (Source: SPEC)

find the optimum combination of compiler flags, others—perhaps concerned more with multiplatform support—compile with "-O" and leave it at that. So far, the SPEC committee's best resolution of the arguments on both sides of the issue is to make both SPEC and SPECbase metrics available and require reporting SPECbase.

SPEC95 still includes the system-level benchmark suites SDM (System Development Multitasking) and SFS (System File Sever); they are not affected by the new definition of SPEC95. In addition, the SPECrate methodology for testing the effectiveness of multiprocessor systems is still supported. The SPECrate methodology is now applicable to uniprocessor, SMP, and cluster-based systems.

Three Input Models Ease Testing Burden

Since a complete trial of the SPEC95 benchmarks can take several days on a mainstream machine, users of the benchmark suite could benefit from the ability to increase the likelihood that a complete trial will produce valid, usable results. To this end, the SPEC95 distribution comes with three sets of input model data for the benchmarks, which are named small, train, and reference. All tools for running the benchmarks and reporting results are written in Perl (which is a popular, powerful Unix scripting language).

The small input model runs quickly—typically one-tenth the time of the full reference model—and is designed simply to check that the benchmarks all compile and run correctly. With the small model, a benchmarker can detect erroneous behavior due to incompatible compiler switches, for example.

The train input model also produces a run of the benchmarks in a fairly short period of time—typically one-tenth to one-eighth of the run time for the reference model. The train input data is specifically for use on machines that have a profile-driven compiler. For these machines, the accepted benchmarking method is to run the benchmark with the training data, use the generated profile to drive the recompilation of the benchmark, then run the profile-optimized version of the benchmark program (at least three times, of course) with the reference model input data. Running a benchmark compiled with profile information gathered from a run on the reference data model would prove interesting, but such results are not acceptable for producing valid, reportable SPEC metrics.

SPEC Prepared for Unforeseen Problems

Though the SPEC committee has attempted to exclude benchmarks that will succumb to special "SPEC-oriented" compiler optimizations, SPEC is prepared to deal with situations of successful compiler attack on susceptible benchmarks. Results submitted for publication in the official newsletter are first reviewed by SPEC for conformance with the run rules. SPEC has reserved the right to alter the suite if necessary to bring results back in line with the goals and criteria of the committee.

Price And Availability

The SPEC95 suite will be distributed only on CD-ROM, and it will be available only in a version containing both suites. The price of $600 for new customers is well below the $995 price charged for the SPEC92 suite to encourage the transition to the new benchmarks. The table below lists complete price information.

Product	New Commercial License	New University License	Current SPEC92 Licensee	Current University Licensee
SPEC95 CD-ROM	$600	$300	$300	$150

To order or get more information, contact Dianne Rice, SPEC, c/o NCGA, Suite 200, 2722 Merrilee Dr., Fairfax, VA 22031; 703.698.960; fax 703.560.2752.

SPEC Hopes SPEC95 Supplants SPEC92

Now that the SPEC92 benchmarks have reached the end of their useful life, SPEC95 brings some much-needed change to processor performance benchmarking. A faster reference machine, much longer run times, resistance to compiler attacks, and a stated policy for dealing with problems as they arise will keep SPEC benchmarking relevant in the face of advancing software and hardware technology.

The committee debated the use of a scaling factor to bring SPEC95 metrics into the same range as existing SPEC92 metrics but decided against it. By choosing not to scale SPEC95 metrics, SPEC has made it trivial to spot mislabeled results: a reasonable 100-MHz Pentium box cannot have a SPECint95 of 100 nor can it have a SPECint92 of 2.5.

The SPEC92 results published in the SPEC newsletter will be annotated as obsolete beginning with the March 1996 issue. After September 1996, SPEC92 results will no longer be published. SPEC wishes to retire SPEC92 as quickly as possible, both to guarantee the success of SPEC95 and to prevent users and the press from passively accepting and printing potentially misleading SPE92 results. The committee emphasizes the point that SPEC95 results may change the relative performance of machines as measured by SPEC92.

SPEC believes everyone involved benefits from the quickest, smoothest possible transition to SPEC95. To this end, SPEC intends to immediately publish a large number of SPEC95 results for existing platforms that have already published SPEC92 results. ♦

