Open Forum

"Open Forum" is premiering in this issue of the SPEC Newsletter. *Its purpose is to provide a forum for SPEC members and newsletter subscribers to discuss issues, controversial or otherwise, related to SPEC's purpose of providing performance evaluation benchmarks and information.*

Articles in the "Open Forum" are the opinions of their authors and do not reflect the official position of SPEC, its board of directors, committees, or member companies.

SPEC welcomes articles, letters, commentary, editorials, etc., from anyone who wishes to publicly voice their opinion on computer system performance evaluation topics. See page 2 for information on submitting material.

Point: Why SPEC Should Burn (Almost) All Flags

by Reinhold Weicker
Siemens Nixdorf Information Systems
Munich, Germany

Should burning the U.S. flag be a federal crime? This question has been a hotly debated topic recently in the United States. Fortunately, the flags this article deals with are something different: compiler and linker options applicable to the generation of a specific program are often called "flags"; in UNIX environments they are typically given in the form of some "-O4" or "-optimize_xyz" strings in the command line invoking the compiler or linker.

Even though these flags do not stir up national emotions, they are also a hot topic. There can be, and have been within SPEC different opinions as to whether the use of these flags in benchmarking is useful and which flags should be allowed. This article is therefore not an official SPEC statement, it is intended to provoke thoughts and opinions from the readers. It is admittedly and intentionally subjective; it explains my opinion on why SPEC's customers, i.e., the computer users at large are better off if we really do "burn our flags".

SPEC's Policy on Optimizations

It is evident from the latest results published in the newsletter that the number of flags used for benchmark measurements is increasing. Obviously, this comes from the pressure to beat the last bit of performance out of a machine when it executes a certain benchmark. Apparently vendors (and so far only computer manufacturers are SPEC members) feel that they cannot afford anything but the best possible SPEC numbers.

What are SPEC's rules on flags and optimizations? Based on experiences with earlier benchmarks like Whetstone and Dhrystone, SPEC has made the following two principles part of the Run and Reporting Rules:

See FLAG BURNING, page 6

Counterpoint: Defending the Flag

by Walter Bays
Sun Microsystems
Mountain View, Calif.

In the U.S., flag burning is constitutionally protected as freedom of speech. But, by custom, old worn-out flags which have become unsightly are burned to dispose of them in a dignified manner. Let us similarly dispose of unsightly compiler flags.

We must not get carried away with the burning, but rather find a middle ground that serves the needs of our customers.

Not a "Black and White" Issue

To begin, let's recognize that this is not an easy black and white issue. Even some portability flags, e.g., " -DSYSV" or "-ANSI," do have performance implications, but SPEC members carefully review all benchmarks to make sure the playing field is level. Then, all submitted results are reviewed by a committee to assure compliance with the run and reporting rules, where any questionable flags are discussed at length. We found, however, that while it was easy for us to spot and question an option such as: "-Dfloat=double" (allowed), it was more difficult to notice an option like "-XZ741" (hypothetical example). Thus, with the introduction of the new CINT92 and CFP92 suites we have begun a more detailed review process that includes a full disclosure of the meaning of each option used.

Dr. Weicker has been instrumental in organizing and leading this more rigorous review process, and we look to him for further improvements. However, while I believe that "one flag" is a worthy goal for each SPEC member company, it is not an appropriate goal for SPEC.

See DEFENDING THE FLAG, page 8

FLAG BURNING, *from page 5*

1. **No source code change, except for portability**

   "SPEC recognizes one class of source code change: those for portability. When source code changes are made, a "diff" listing of the changes must be included in the testing report. SPEC encourages performance comparisons be made across systems running the unmodified SPEC code, or if necessary, with the minimum required portability changes."

2. **No benchmark-specific optimizations**
   "Use of software features (in pre-processors, compilers, etc.) which invoke, generate or use software designed specifically for any of the SPEC Benchmark Releases is not allowed."

The question is: how should these rules be interpreted when it comes to specific flags?

Flags for Portability

There is generally no problem with flags that are necessary for portability, e.g., "-DSYSV" or "-ANSI". They are necessary because compilers need to compile programs written in the ANSI standard form of programming languages as well as those written in older language dialects like K&R C. Also, programs may need to be adapted to the different flavors of UNIX.

Flags That Substitute Code From Special Libraries

In principle, it is a violation of the rule "no source code change" if subroutines that occur in the benchmark source code are replaced by precomputed subroutines from special libraries. However, the SPEC Steering Committee has recognized the fact that certain collections of subroutines – like the Fortran "BLAS" subroutines for linear algebra, or an optimized "malloca" for 085.gcc – can be considered de facto language extensions, are frequently used in practice, and has therefore allowed their use. Substitution of other functions would require explicit permission of the Steering Committee. However, since some substitutions are now legal, readers of benchmark results should know that a flag like "-lblas" * means that optimized, typically assembly-language coded routines are running instead of the C or FORTRAN code given in the benchmark itself. For some programs like "093.nasa7" this may result in a substantial performance gain.

Flags That Are Intended for Specific Compilation Units or Subroutines

In the early times of SPEC, typically only a few compiler flags were used and they were applied to all parts of a program. For recent result publications, one can sometimes see how optimizations are chosen for the different compilation units differently. One unit is compiled with one set of flags, another one is compiled with different flags. There are flags (e.g. "-inline=indxx") that perform certain optimizations, like inlining, for specific subroutines only. Experiments have shown that the program runs a bit faster if a certain subroutine is expanded inline. Certainly the decision of which functions should be expanded inline requires a knowledge of the benchmark. Is this optimization now a "benchmark-specific optimization" ruled illegal by SPEC? Proponents of such optimizations argue that for heavily-used programs, programmers do take great care to find the best possible optimizations. However, this will only be in a minority of cases. Should SPEC base its results on such program development style?

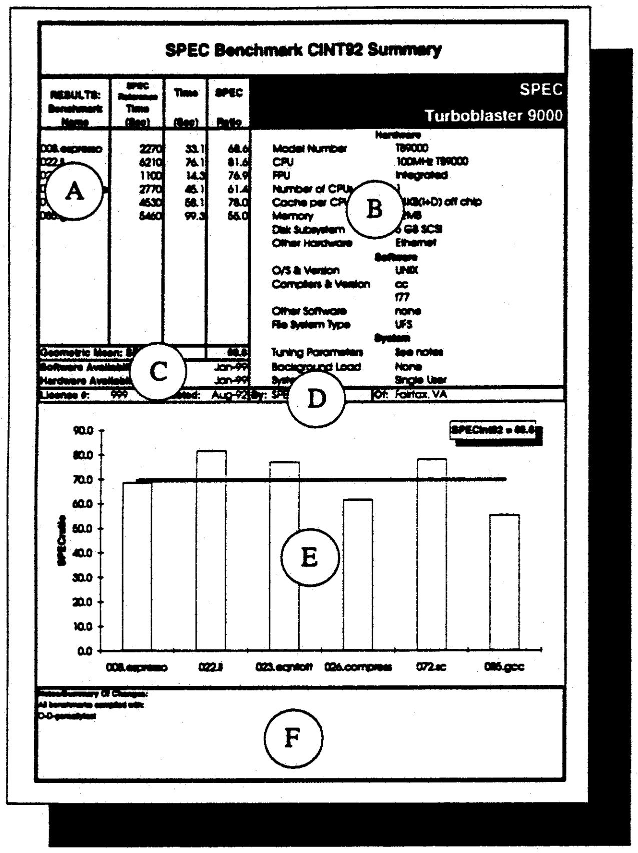
In discussions about such gray areas of benchmarking, I have heard the argument: "Wouldn't it hold up the progress of technology if optimizations, including aggressive optimizations, are outlawed?" But is it really progress of technology if SPEC member companies spend hours of CPU time and months of engineering effort to manually find out the best combination of compiler and linker flags? Compilers that are smart enough to find out themselves, in an automated way, candidates for inlining, loop unrolling parameters, etc., would really be progress of technology. Such a compiler would benefit all programs, not just carefully scrutinized benchmarks. By allowing hand-selected optimizations that name specific subroutines for inlining and other such targeted optimizations, we lead the vendor companies into a direction (hand-tuning) that may result in higher SPEC numbers but not in improvements in compiler technology that benefit all of our customers' programs.

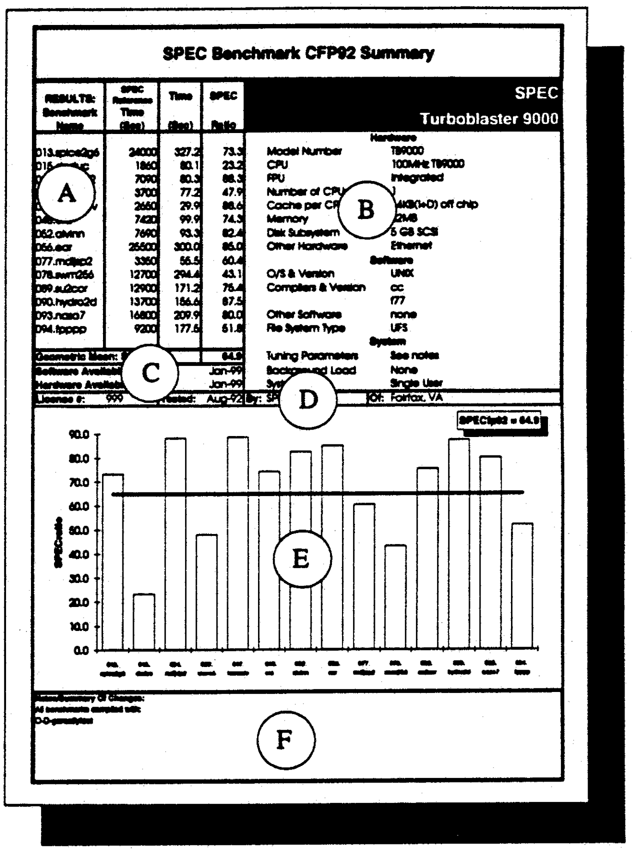
* *All examples are from actual result submissions. Detailed references are intentionally omitted. It is not the purpose of this article to point at a specific vendor, but rather to discuss the fundamental issues.*

See FLAG BURNING, *page 7*

SAMPLE REPORTING PAGES

The following pages contain annotated mock-ups of the various styles of Reporting Pages with an explanation of the materials that go in each section. It is hoped that the more a reader can understand about the Reporting Pages, the more readers will get out of SPEC publications.





SPEC Benchmark CINT92/CFP92 Summary

A: Results Section: A listing of the benchmark names with their reference times, along with the measured elapsed times and the calculated SPECratios. Here are the specific values that can be used for performance comparisons.

B: System Configuration: This section describes the system and its configuration used to achieve the performance presented. The specific packages and the versions utilized for the measured runs are detailed.

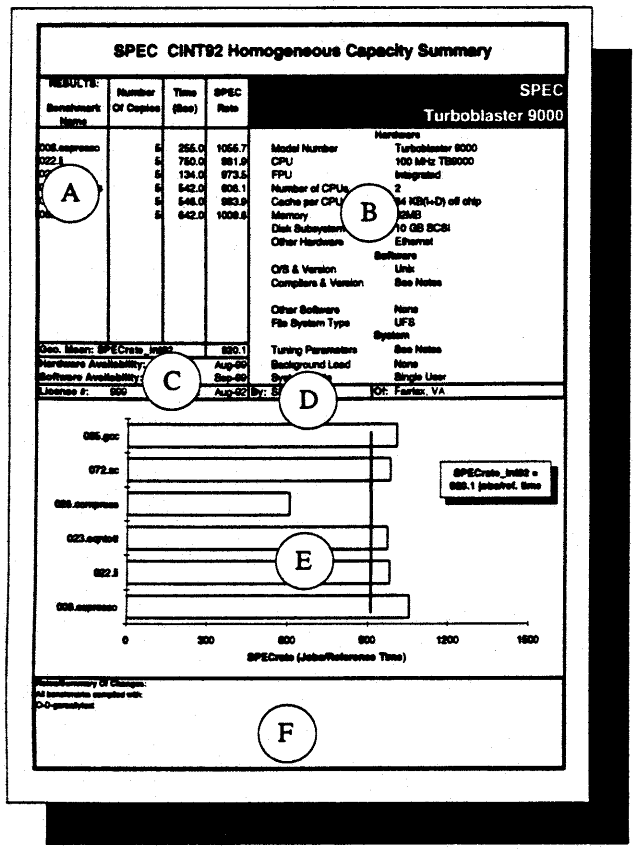
C: Availability: These are the vendor's estimates of when the system reported was or will be available.

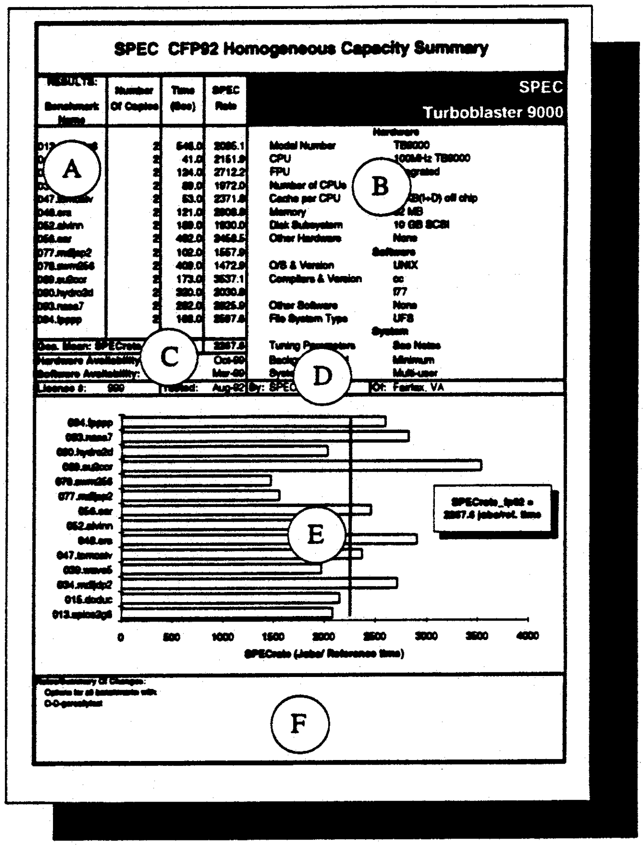
D: Sponsor: This line describes who ran these tests, where, and when. All test sponsors are bound by the license agreement to publish results only in accordance with the run and reporting rules; the holder of the license whose number is listed must make sure that all the rules have been followed.

E: Graph: This is a graphical display of the SPEC metrics listed in the Results Section. The SPECratio for each benchmark is drawn as a bar on a chart — there is no significance to the ordering of the bars, just the display of the varying sizes. The line across all the bars represents the geometric mean of all the ratios and that summary mean value is highlighted in a box placed within the graph. The Y axis limits are in multiples of 30 to facilitate comparisons between similar systems.

F: Notes Section: The details about compiler and operating system options used and other features necessary to replicate the results are described here.

Sample Reporting Pages





SPEC CINT92/CFP92 Homogeneous Capacity Summary

A: Results Section: A listing of the benchmark names, then the number of copies run and the measured elapsed times and the calculated SPECrates are listed here. These are the specific values that can be used for performance comparisons.

B: System Configuration: This section describes the system and its configuration used to achieve the performance presented. The specific packages and the versions utilized for the measured runs are detailed.

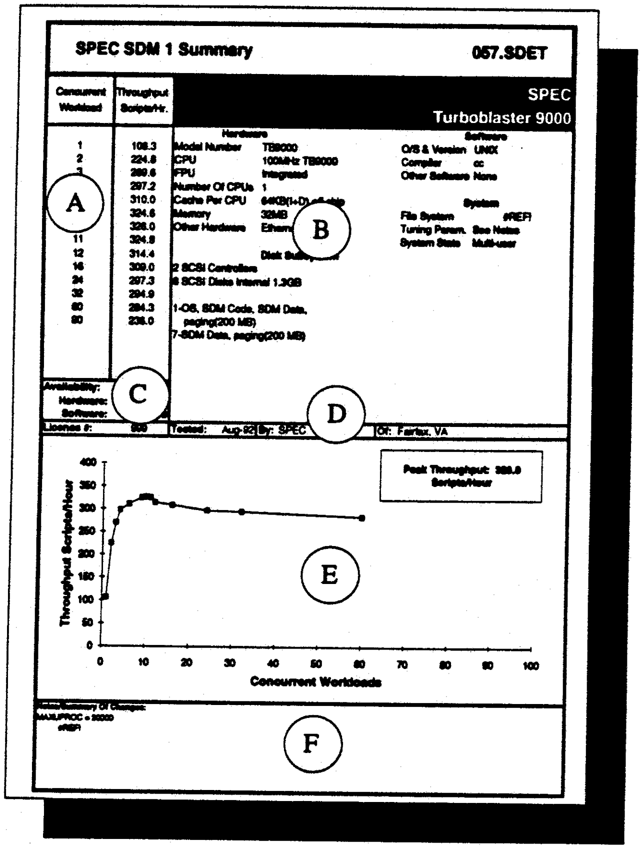
C: Availability: These are the vendor's estimates of when the system reported was or will be available.

D: Sponsor: This line describes who ran these tests, where, and when. All test sponsors are bound by the license agreement to publish results only in accordance with the run and reporting rules; the holder of the license whose number is listed is the one with the responsibility that all the rules have been followed.

E: Graph: This is a graphical display of the SPEC metrics listed in the Results Section. The SPECrate for each benchmark is drawn as a bar on a chart — there is no significance to the ordering of the bars, just the display of the varying sizes. The line across all the bars represents the geometric mean of all the ratios and that summary mean value is highlighted in a box placed within the graph. The X axis limits are in multiples of 500 to facilitate comparisons between similar systems.

F: Notes Section: The details about options used and other features necessary to replicate the results are described here.

Sample Reporting Pages



SPEC SDM 1 Summary 057.SDET

SPEC
Turboblaster 9000

Concurrent Workload	Throughput Scripts/Hr.
1	108.3
2	224.8
3	269.6
	297.2
	310.0
	324.6
	326.0
11	324.8
12	314.4
16	309.0
24	297.3
32	294.9
60	284.3
80	238.0

Hardware

Model Number	TB9000
CPU	100MHz TB9000
FPU	Integrated
Number Of CPUs	1
Cache Per CPU	64KB(I+D) off chip
Memory	32MB
Other Hardware	Ethern

Disk Subsystem

2 SCSI Controllers
8 SCSI Disks Internal 1.3GB

1-OS, SDM Code, SDM Data, paging(200 MB)
7-SDM Data, paging(200 MB)

Software

O/S & Version	UNIX
Compiler	cc
Other Software	None

System

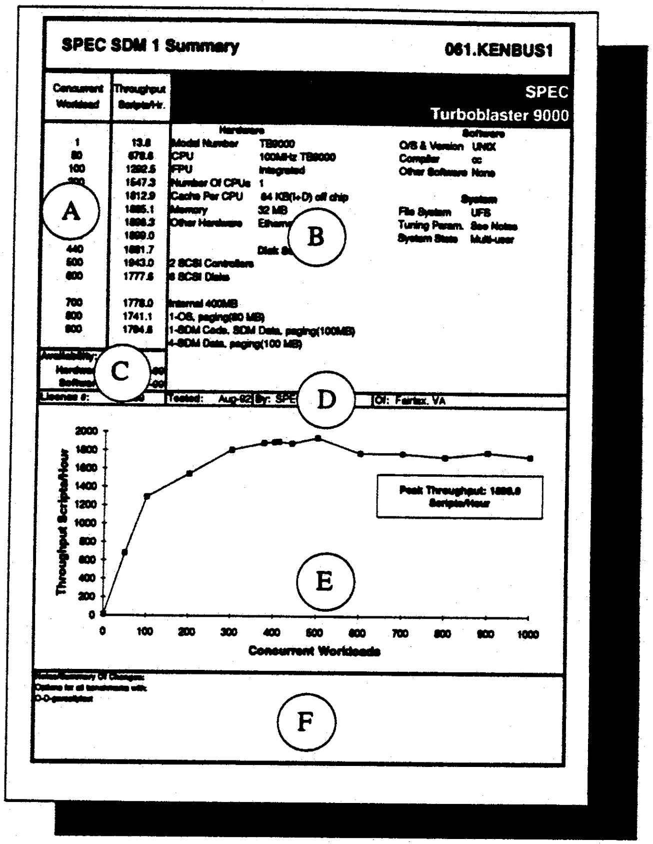
File System	#REF!
Tuning Param.	See Notes
System State	Multi-user

Availability:
Hardware:
Software:

License #: 999 | Tested: Aug-92 | By: SPEC | Of: Fairfax, VA

Peak Throughput: 326.0 Scripts/Hour

Notes/Summary Of Changes:
MAXUPROC = 30000
#REF!



SPEC SDM 1 Summary 061.KENBUS1

SPEC
Turboblaster 9000

Concurrent Workload	Throughput Scripts/Hr.
1	13.8
50	678.6
100	1292.5
200	1547.3
	1812.9
	1885.1
	1898.3
	1899.0
440	1881.7
500	1943.0
600	1777.6
700	1778.0
800	1741.1
900	1784.6

Hardware

Model Number	TB9000
CPU	100MHz TB9000
FPU	Integrated
Number Of CPUs	1
Cache Per CPU	64 KB(I+D) off chip
Memory	32 MB
Other Hardware	Ethern

Disk Su

2 SCSI Controllers
6 SCSI Disks
Internal 400MB
1-OS, paging(50 MB)
1-SDM Code, SDM Data, paging(100MB)
4-SDM Data, paging(100 MB)

Software

O/S & Version	UNIX
Compiler	cc
Other Software	None

System

File System	UFS
Tuning Param.	See Notes
System State	Multi-user

Availability:
Hardwa
Softwa

License #: | Tested: Aug-92 | By: SPE | Of: Fairfax, VA

Peak Throughput: 1899.0 Scripts/Hour

Notes/Summary Of Changes:
Options for all benchmarks with:
O-O-guesslptest

SPEC SDM 1 Summary

A: Results Section: The concurrent workloads reported and the resulting throughput are listed here with specific values that can be used for performance comparisons.

B: System Configuration: This section describes the system and its configuration used to achieve the performance presented. The specific packages and the versions utilized for the measured runs are detailed. The disk subsystem and the partition and location information might be very helpful in understanding how to configure such a system to handle a significant amount of I/O.

C: Availability: These are the vendor's estimates of when the system will be available.

D: Sponsor: This line describes who ran these tests, where, and when. All test sponsors are bound by the license agreement to publish results only in accordance with the run and reporting rules; the holder of the license whose number is listed is the one with the responsibility to make sure that all the rules have been followed.

E: Graph: This is a graphical display of the curve which comes from plotting the points listed in the Results Section. The 057.SDET graph should rise sharply from near the origin, reach a plateau near the peak performance, and then gradually tail off. The 061.Kenbus1 graph will rise steadily out from the origin and gradually crest over the peak performance. It may plateau for a while, then it will tail off. On these benchmarks, thoughput rises as the workload increases, until the system begins to bottleneck on resources at or near its peak, then performance will tail off as the workload increases beyond the available resources and the system begins to suffer the overhead of being overdriven. The X axis limits in these graphs are chosen to facilitate comparisons between similar systems. For 057.SDET the limit is one from the series 50, 100, 200, 400,... For 061.Kenbus1, the limit is a multiple of 500.

F: Notes Section: The details about kernel parameters and system options necessary to replicate the results are described here.

Benchmark Results Reporting Format Terminology

Portability Changes: Minimum performance neutral changes to the source code necessary to make the benchmark run correctly. These changes are reviewed and approved by other SPEC Steering Committee members before being published in the SPEC Newsletter.

SPEC CINT92 and CFP92

SPEC Reference Time: The time in seconds it takes a DEC VAX 11/780 to run each individual benchmark in the suites, accurate to three significant digits.
SPECratio: The SPECratio for each benchmark is the quotient to the nearest tenth, derived by dividing the benchmark's SPEC Reference Time by the elapsed run time on a particular system to run the same benchmark. For example, the SPEC Reference Time for 008.espresso is 2270 seconds. If a particular system ran 008.espresso in 227 seconds, the system's SPECratio for 008.espresso is computed as 2270/227 and equals 10.0.
SPECint92: Geometric mean of the SPECratios from the six benchmarks in CINT92 (008.espresso, 022.li, 023.eqntott, 026.compress, 072.sc and 085.gcc). This is obtained by taking the sixth root of the product to the six SPECratios.
SPECfp92: Geometric mean of the SPECratios from the fourteen benchmarks in CFP92 (013.spice2g6, 015.doduc, 034.mdljdp2, 039.wave5, 047.tomcatv, 048.ora, 052.alvinn, 056.ear, 077.mdljsp2, 078.swm256, 089.su2cor, 090.hydro2d, 093.nasa7 and 094.fpppp). This is obtained by taking the fourteenth root of the product of the fourteen SPECratios.

HOMOGENOUS CAPACITY METHOD based on SPEC CINT92 and CFP92

SPECrate = #CopiesRun * ReferenceFactor
* UnitTime / ElapsedExecutionTime

where #CopiesRun is the number of concurrent copies of the benchmark run, ReferenceFactor is the reference time for the benchmark (see definition above) divided by 25,500, UnitTime is the number of seconds in a week, and ElapsedExecutionTime is the time from starting the concurrent copies of the benchmark until all copies have completed.

SPECrate_int92: Geometric mean of the SPECrates from the six benchmarks in CINT92.
SPECrate_fp92: Geometric mean of the SPECrates from the fourteen benchmarks in CFP92.

SPEC SDM 1

SDET Peak Throughput: The maximum value of the throughput reached, by monotonically increasing the offered workload on the system in terms of the number of concurrently executed SDET scripts, while running the SDET benchmark.
KENBUS1 Peak Throughput: The maximum value of the throughput reached, by monotonically increasing the offered workload on the system in terms of the number of concurrently executed KENBUS1 scripts, while running the KENBUS1 benchmark.
Concurrent Workload: The number of scripts executed concurrently during a benchmark run is termed the concurrent workload. The scripts are either SDET or KENBUS1 scripts as per context.

SPEC Suite1

SPEC Reference Time: For Release 1, the SPEC Reference Time is the time (in seconds) that it takes a Digital Equipment Corporation VAX 11/780 machine to run each particular benchmark in the suite. Consequently, the reference time differs with each benchmark.
SPECratio: The SPECratio for a benchmark is the quotient to the nearest tenth, derived from dividing the SPEC Reference Time by a particular machine's corresponding run time. For example, if the SPEC Reference Time was 1501 seconds, and machine X's run time was 521 seconds, the calculation would be: 1501/521 = 2.9 SPECratio.
SPECmark89: Geometric mean of the 10 SPECratios. Compared with the arithmetic mean (average), the geometric mean is a fairer way of reporting suite results because it compensates for isolated SPECratio extremes while giving each program equal importance.
SPECint89: Geometric mean of the results of the four integer benchmarks: 001.gcc, 008.espresso, 022.li, and 023.eqntott. This is obtained by taking the fourth root of the product of the four SPECratios of the above mentioned benchmarks.
SPECfp89: Geometric mean of the results of the six floating-point benchmarks: 013.spice2g6, 015.doduc, 020.nasa7, 030.matrix300, 042.fpppp, and 047.tomcatv. This is obtained by taking the sixth root of the product of the six SPECratios of the above mentioned benchmarks.