SPEC Benchmark Suites, A Summary

by Subra Balan, IBM Corporation

The following is a summary of the SPEC benchmark suites that are available today, what they represent, and their associated performance metrics.

SPEC Release 1

SPEC Release 1, a suite of 10 CPU intensive benchmarks was released in October 1989. Six of these 10 exercise the floating-point and the other four the integer computational capability of the CPU.

These benchmarks were chosen from real end-user applications and represent engineering and scientific environments. They are representative of performance on today's high-performance machines, optimizing compilers, multiprocessing capabilities and large instruction and data caches which have made traditional CPU benchmarks such as Dhrystone and Whetstone obsolete. For a detailed description of each of the 10 benchmarks in the suite, please refer to the the article "The SPEC Benchmark Suite, Release 1 -What Is It?" in the *SPEC Newsletter,* Volume 3, Issue 1,Winter 1991.

The elapsed run time of each of the benchmarks on the SPEC Reference Machine (VAX 11/780) is divided by the elapsed run time on the measured system to yield 10 SPECratios. The geometric mean of these 10 SPECratios is computed as the system's **SPECmark89** and is an overall measure of CPU performance. The geometric mean of the four integer benchmark SPECratios is called **SPECint89** and that for the six floating point benchmarks is called **SPECfp89**. **SPECint89** and **SPECfp89** represent the integer and floating-point performance of the system respectively.

The older metrics **SPECmark**, **SPECint** and **SPECfp** have been renamed with a suffix of "89" to distinguish them from the **SPECint92** and **SPECfp92** metrics from the newer **CINT92** and **CFP92** suites.

To date, results on approximately 100 systems have been published by various vendors in the SPEC newsletters.

SPEC SDM 1

In May 1991, SPEC released its second suite of benchmarks called **SPEC SDM 1**. SDM (System Development Multi-tasking) 1 is a system-level suite comprised of two benchmarks called **057.sdet** and **061.kenbus1**. These benchmarks represent a UNIX- and C- based software development environment. **057.sdet** is patterned after commands found in a commercial software development environment and **061.kenbus1** uses commands found in a research and development environment. For a detailed description of **057.sdet** and **061.kenbus**, please refer to the technical fact sheet and articles in the *SPEC Newsletter*, Volume 3, Issue 2, Spring 1991.

These benchmarks exercise the CPU, memory, disk I/O and operating-system services. They are multi-tasking benchmarks and do not require an external driver. Therefore, true end-user response time is not measured. Results for these benchmarks are expressed in scripts/hour. These benchmarks apply work to the system in units of scripts (sequence of UNIX commands, programs and shell scripts) and the throughput is measured in terms of the number of scripts completed per unit time. Peak system throughput as well as a curve of the workload applied vs. the throughput generated are reported. While **057.sdet** and **061.kenbus1** report throughput in scripts/hour, their results cannot be directly compared, as they each apply a different amount of work to the system being measured.

To date, results on 37 systems have been published by various vendors in the SPEC newsletters.

SPEC CINT92 and CFP92

On January 15, 1992, SPEC released its enhanced CPU suites titled **CINT92** and **CFP92**. These suites build upon the real end-user benchmarks in **SPEC Release 1**. They contain several additional application benchmarks and enhancements to keep pace with the faster hardware and smarter optimizing compilers. **CINT92** is a suite of six integer benchmarks. **CFP92** is a suite of 14 floating-point benchmarks, five of which are single precision. The SPECratios for each of these benchmarks is computed similarly to **SPEC Release 1**. The geometric mean of the six integer benchmark SPECratios is called **SPECint92** and represents the integer performance of the CPU. Similarly, **SPECfp92** represents the floating-point performance of the CPU. An overall measure such as the **SPECmark89** is no longer computed and reported since such a measure masked certain key performance characteristics of the systems being compared. For example, system A and system B could have the same **SPECmark89** with system A exceeding system B in integer performance (**SPECint89**) and system B exceeding system A in floating- point performance (**SPECfp89**).

Depending on the requirements that the end user has, the right solution could have been system A or system B, but this is not conveyed by the **SPECmark89.**

At the product announcement results on 14 systems from HP, IBM, Solbourne and Sun were released. Many more results are published in this SPEC Newsletter.

It has been SPEC's position that these standard benchmarks are not intended as a replacement for benchmarking the actual customer application. However, they do provide a fair and equitable means of comparing the relative performance of various systems. SPEC also recommends that users avoid comparing using single-number composites such as **SPECmark89, SPECint89, SPECfp89, SPECint92, SPECfp92,** and the **057.sdet** and **061.kenbus1** peak scripts/hour. Comparison of CPU performance is better done by selecting one or more individual benchmarks that more closely represent the customer's application than the composite could. Comparison of **057.sdet** and **061.kenbus** performance is better done by using the entire performance curve. For this reason, the results pages designed by SPEC and published in the SPEC newsletters ensure as complete disclosure as possible to enable detailed comparison and independent reproduction of published vendor results.

Subra Balan works at the IBM Performance Evaluation Center located in Roanoke, Texas.

How To Build Your Own Benchmarks Using SDM Release 1 Suite With Custom UNIX Workloads

S. Krishna Dronamraju & Asif Naseem, AT&T/NCR Corporation, Naperville, Illinois

The **SDM Release 1** suite is ideally suited for inserting custom UNIX workloads and running them using the SDM template. Here the tools available in the **057.sdet** benchmark are explained with enough detail such that informed users can plug in their own custom workloads.

057.sdet benchmark contains several directories, among which bin, output, scripts and masterclone directories need to be changed to incorporate any new workloads. The **057.sdet** benchmark workload is in the " file. X" files under the scripts directory. The UNIX commands that are used as a workload are entered in the scripts directory in files, file.0 ... file.20. The content of these files need not be unique. To increase the workload, they may be duplicated many times over. The number of files is also not unique. If a different set of files is chosen, then the random numbers collected in the randsets have to be regenerated using the shell programs, gen.tscripts and genrands.

When the benchmark is executed, a number of processes (users) are forked off by the benchmark driver program and each such user process runs in a directory of its own called a clone. These clone directories are built by the benchmark preprocessing tools in a disk partition optioned by RUNPLACES variable in the Makefile wrapper provided. These clone directories are built using the masterclone directory. The masterclone directory can be described as a typical user's directory and it is copied as many times as the number of clones optioned by the RUNSEQUENCE make wrapper variable. The benchmark automatically captures the output from each clone run in the mrun.n files in the output directory. At the end of a typical run these generated run (2run.1, 2run.2 for a 2 clone run) files are filtered through the cleanstderr and the output is sorted and compared with the generic file, provided in the same directory. In case there are differences between these files and the generic file, error messages are generated and performance metrics are not calculated.

To introduce a new workload the following changes need to be done:

1. Remove the current workload kept in the scripts directory, in file.0 .. file.20 and incorporate your own workload. These 21 files are at random combined [mixed] to form term scripts, term.1 etc, which are used as workloads by each clone process. In case the total number of basic files are different from the 21 files then the three shell programs, gen.tscripts genrands and asmbl.sh need to be altered appropriately.

2. Remove the current application directories from the masterclone directory. Appropriately choose a clone environment and application files and design them to work with the set of UNIX workload [commands] you have chosen in the scripts directory.

3. Make a trial run of the workload by executing the benchmark. The output generated by each clone process is captured in the mrun.n files and when they are compared with the existing generic file, errors are generated, as expected. Any of these run files can be taken and following the same steps given in the shell programs, compare.sh, errckr and cleanstderr, a new generic file can be generated .

Usually a few iterations of these steps should produce a flawless benchmark with custom designed workload.

S. Krishna Dronamraju and Asif Naseem work at the AT&T/NCR Laboratories in Naperville, Illinois.

LADDIS File Server Benchmark

By Bruce Keith, Digital Equipment Corporation

A vendor-neutral Network File System (NFS) file server benchmark was submitted to SPEC during the August 1991 SPEC Benchathon by the LADDIS Group, an informal organization of leading NFS vendors. Called LADDIS, the benchmark generates a synthetic NFS workload based on a workload abstraction of an NFS operation mix and an NFS operation request rate. The benchmark measures NFS file server performance in terms of NFS operation response time and throughput.

The LADDIS benchmark will be the first benchmark in SPEC's System-level File Server (SFS) Benchmark Suite.

BACKGROUND

Motivated by the need to provide customers with a unified, standard method of comparing NFS file server performance, the LADDIS Group was formed in February 1990. The name LADDIS is an acronym of the founding companies. The group presently consists of technical representatives from Legato Systems (Robert Lyon), Auspex Systems (Bruce Nelson), Data General (Mark Wittle), Digital Equipment (Bruce Keith), Interphase (Vincent Lefebvre), and Sun Microsystems (John Corbin). The LADDIS Group now works within the SPEC organization in the System-level File Server (SFS) Subcommittee.

Written in C, the LADDIS benchmark is loosely based on Legato's 1989 nhfsstone benchmark program. LADDIS maintains nhfsstone's NFS workload abstraction of an NFS operation mix and an NFS operation request rate. However, LADDIS overcomes limitations in nhfsstone, specifically, its NFS client platform sensitivity, the accuracy of generated load, and single NFS client operation.

LADDIS minimizes NFS client platform sensitivity and offers improved accuracy of its generated load by executing the NFS protocol directly within the benchmark. Thus, NFS protocol implementation issues such as file attribute and data caching are normalized across different vendors' platforms running the LADDIS benchmark. This normalization allows LADDIS to measure file server performance independent of the NFS client platform used to execute the benchmark. Hence, LADDIS is a measure of NFS file server performance and is not a measure of NFS client performance.

The multiple-client capability of LADDIS allows collections of NFS clients to be used to generate an aggregate load on an NFS server using one or more TCP/IP networks. This is important since a single client or network may saturate before a given server saturates. Further, the use of multiple NFS clients provides a more realistic environment with respect to network contention.

OPERATION

To benchmark an NFS server, the server and two or more NFS client systems are configured on one or more isolated networks. One of the NFS clients is designated the "LADDIS Prime Client." The prime client controls the execution of the LADDIS NFS load-generating code on the remaining NFS clients, which are called "LADDIS Clients." Figure 1 *(see page 12)* illustrates a LADDIS testbed comprised of an NFS server, the LADDIS Prime Client and several LADDIS Clients.

The prime client typically executes the NFS load-generating code in addition to controlling the benchmark. However, the prime client is not required to execute the NFS load-generating code. This permits an external "test manager" to control the benchmark and coordinate other distributed system-performance monitoring processes outside the scope of the LADDIS benchmark. This allows the benchmark to be used as a server characterization tool as well as a benchmark.

The individual conducting the test uses a single user interface on the LADDIS prime client to control the benchmark. All test parameters are defined in a single control file, called laddis_rc. To start the test, the laddis_mgr control script is invoked. The laddis_mgr control script executes successive runs of the benchmark at increasingly higher NFS load levels while holding the NFS operation mix constant.

METRICS

The LADDIS benchmark measures NFS server response at the NFS protocol level on the LADDIS Clients for each NFS load level. For each run of the benchmark, the server's average NFS operation response time is measured for the applied NFS load.

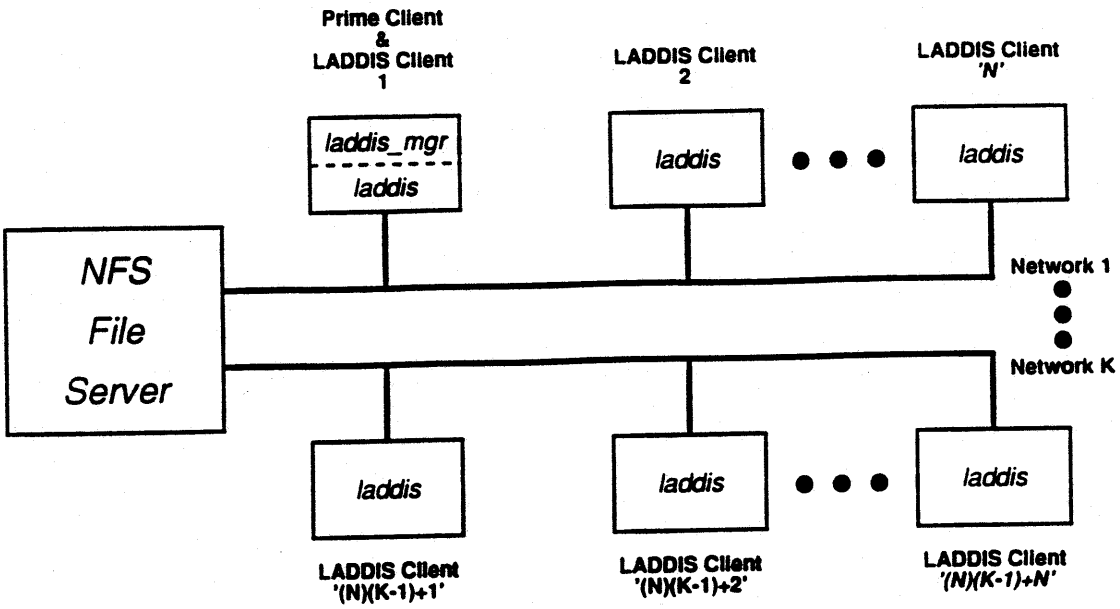

Figure 1: LADDIS Testbed Scenario

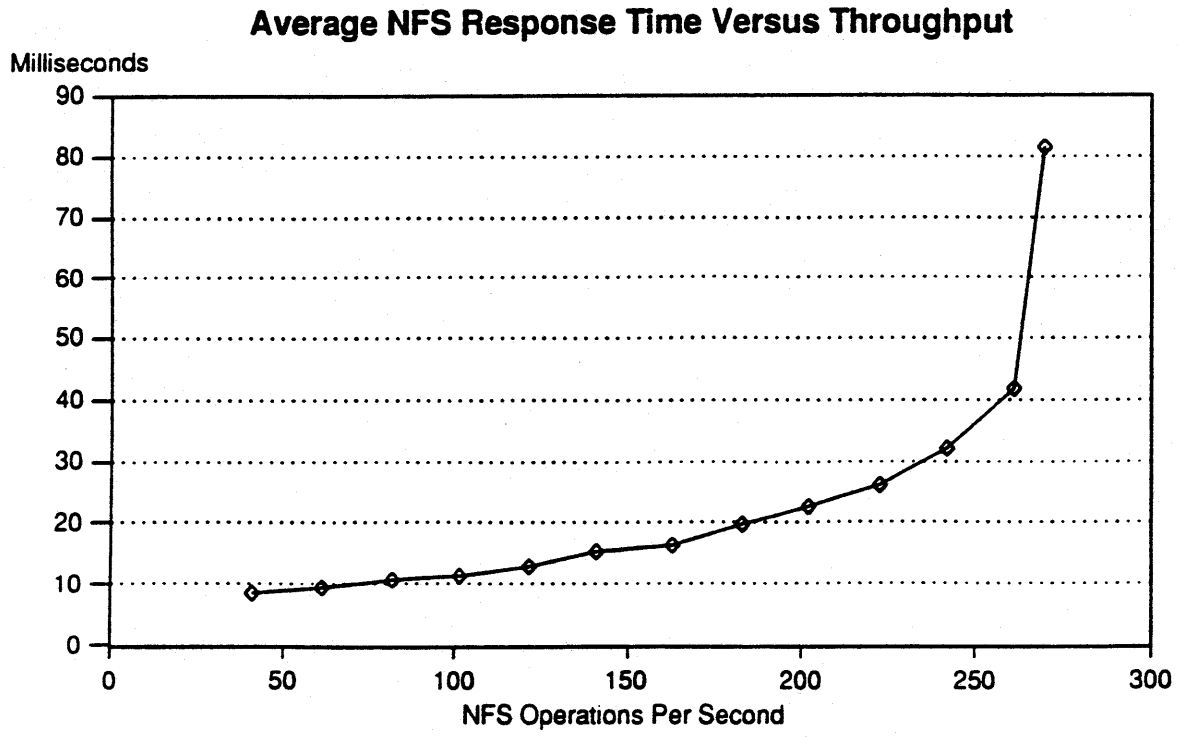

Figure 2: NFS Server Response Time versus Throughput Graph

At the conclusion of a series of runs, a graph of NFS server response time versus NFS load (throughput) for the particular mix of NFS operations is constructed as shown in Figure 2 *(see page 12)*. Average NFS operation response time in units of milliseconds is plotted on the y-axis while average NFS throughput in terms of operations per second is plotted on the x-axis.

Different servers or different configurations of a given server can be compared by plotting their LADDIS NFS response time versus NFS throughput curves on a single graph.

NFS server performance is most accurately depicted by the NFS response time versus NFS throughput curve. The curve clearly illustrates the rate of server response degradation for increased server load. Thus, the LADDIS Group has proposed that SPEC report LADDIS results in a format that is similar to that used in the SPEC SDM Release 1 Summary.

The SDM Release 1 results reporting format contains a graph, a list of data points, and a summary of the relevant hardware, software, and system parameters used to conduct the test. The testbed component summary would be extended to include appropriate LADDIS-specific items such as the NFS operation mix and network components used in the test.

STATUS

The work to port the LADDIS NFS server benchmark to all of the SPEC members' platforms began at the August 1991 SPEC Benchathon and continued at the December 1991 Benchathon. At the conclusion of the December 1991 Benchathon, PRE-LADDIS Version 0.0.16 was running on 10 of the 14 SPEC members' platforms present and compiling on 13 out of the 14 platforms spanning AIX, BSD, SVR3, and SVR4 environments.

Further, interoperability testing began during the December 1991 Benchathon in which 10 different SPEC members' platforms successfully participated in a multiple-client run of the LADDIS benchmark targeting an NFS server.

Interoperability testing and porting efforts will continue at the March 1992 and May 1992 SPEC Benchathons. In addition, SPEC Run and Reporting Rules for the LADDIS benchmark are currently being formulated during regular SPEC meetings and Benchathons.

SPEC's first beta test of a candidate benchmark began in February 1992 when a beta test version of the LADDIS benchmark was made available to the computer industry. The beta test version, called PRE-LADDIS Version 0.1.0, is licensed by SPEC. The beta test is intended to capture the porting and execution experiences of those beyond the SPEC organization to broaden the base of experience of those running the PRE-LADDIS benchmark. The feedback from the beta test process will be incorporated into the development effort for the final SPEC version of LADDIS.

The SPEC PRE-LADDIS beta test license can be obtained by sending an electronic mail message to "spec-preladdis-beta-test@riscee.pko.dec.com" with a subject line of "Request SPEC PRE-LADDIS Beta Test License". The SPEC PRE-LADDIS beta test kit is sent by electronic mail to the requestor upon SPEC's receipt of the requestor's signed beta test license.

FURTHER READING

Keith, Bruce, "Perspectives on NFS File Server Performance Characterization," *USENIX Summer Conference Proceedings*, pp. 267-277, June 1990.

LADDIS Group, "LADDIS - A Vendor-Neutral Standard NFS Benchmark," *INTEROP 1991 Fall Conference Proceedings*, October 9, 1991.

Sandberg, Russel, et al., "Design and Implementation of the Sun Network Filesystem," *USENIX Summer Conference Proceedings*, pp. 119-130, June 1985.

Bruce Keith is the NFS performance project leader at Digital Equipment Corporation in Maynard, Mass. He is the release manager for SPEC's System-level File Server (SFS) Benchmark Suite and is the SPEC Project Leader for the LADDIS benchmark. Bruce is also Digital's representative in the LADDIS Group.

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 Release 1

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. For the throughput measure (SPECthruput), which requires two copies per processor, the reference time is doubled.

SPECratio: The SPECratio for a benchmark is the quotient 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.

SPECthruput89: SPECthruput89 is a measure of the amount of work (throughput), given a particular workload, that a system under test can perform relevant to a reference system. It is derived by comparing the speed of a machine under test running a scaled load against a SPEC reference machine running two copies of the Release 1 benchmarks. To calculate SPECthruput89, first calculate the Thruput Ratio for each benchmark. Divide the reference time (in this case, the time posted by a VAX 11/780 concurrently running two copies of each Release 1 benchmark) by the run time of the machine under test concurrently running two copies of the same benchmark per processor. Therefore, if the reference for running two copies of the gcc benchmark was 100 seconds, and a single-processor machine under test ran the two copies of gcc in 10 seconds, the Thruput Ratio for gcc would be 10 and notated as "1@10." Second, take the geometric mean of all the individual Thruput Ratios and use the notation "#cpu@geomean" where "#cpu" is the number of processors in the system under test and "geomean" is the geometric mean of theThruput Ratios. The Aggregate Thruput can be calculated by multiplying the geometric mean by the number of processors. For example, if the geometric mean is 6, and the number of processors is 3, SPECthruput89 is 3@6 and the Aggregate Thruput is 18.

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.

SPECintThruput89: SPECintThruput89 is the geometric mean of the Thruput Ratios from the four integer benchmarks only.

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.

SPECfpThruput89: SPECfpThruput89 is the geometric mean of the Thruput Ratios from the six floating-point benchmarks only.

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 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, to three significant digits.

SPECratio: The SPECratio for each benchmark is the quotient 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.

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.