The Mass Storage Testing Laboratory at GSFC

Ravi Venkataraman, Joel Williams, David Michaud, Heng Gu, Atri Kalluri, P C Hariharan
Systems Engineering and Security, Inc.
7474 Greenway Center Dr., Suite 700
Greenbelt MD 20770-3523
e-mail: {ravi, joelw, michaud, hengg, atri, hari)@ses-inc.com
Phone: +1-301-441-3694
FAX: +1-301-441-3697

Ben Kobler, Jeanne Behnke, Bernard Peavey
NASA Goddard Space Flight Center
Greenbelt MD 20771-1000
e-mail: {ben.kobler, jeanne.behnke, bernie.peavey}@gsfc.nasa.gov
Phone: +1-301-614-{5231, 5326, 5279}
FAX: +1-301-614-5267

Introduction

Industry-wide benchmarks exist for measuring the performance of processors (SPECmarks), and of database systems (Transaction Processing Council). Despite storage having become the dominant item in computing and IT (Information Technology) budgets, no such common benchmark is available in the mass storage field. Vendors and consultants provide services and tools for capacity planning and sizing, but these do not account for the complete set of metrics needed in today's archives.

The availability of automated tape libraries, high-capacity RAID systems, and high-bandwidth interconnectivity between processor and peripherals has led to demands for services which traditional file systems cannot provide. File Storage and Management Systems (FSMS), which began to be marketed in the late 80's, have helped to some extent with large tape libraries, but their use has introduced additional parameters affecting performance. The aim of the Mass Storage Test Laboratory (MSTL) at Goddard Space Flight Center is to develop a test suite that includes not only a comprehensive check list to document a mass storage environment but also benchmark code. Benchmark code is being tested which will provide measurements for both baseline systems, that is applications interacting with peripherals through the operating system services, and for combinations involving an FSMS.

The benchmarks are written in C, and are easily portable. They are initially being aimed at the UNIX Open Systems world. Measurements are being made using a Sun Ultra 170 Sparc with 256K memory running Solaris 2.5.1 with the following configuration:

- 4mm tape stacker on SCSI 2 Fast/Wide
- 4GB disk device on SCSI 2 Fast/Wide
- Sony PetaStore on Fast/Wide differential SCSI 2

Description of the Benchmark Code

The program exercises the I/O system of the machine on which it is running by processing (writing, reading, and copying) a specified number of files of specified file size and block

size combinations. The code can exercise just the disk or it can exercise the disk and one tape drive.

A system load that is typical for the user may be evaluated by choosing files of different file and block size combinations. The tests are run on a stand-alone system to isolate user, network, and other overhead that may affect the measurement. The result thus obtained would be the ideal (maximum) throughput or the baseline throughput. By running the tests over the network the network overhead may be computed. If required, external loads may be applied

The same benchmark could be run with the system subject to other load on CPU and I/O or run under Client/Server mode to measure the response times under typical user operating conditions.

In order to exercise a Hierarchical Storage Manager (HSM), it is only necessary to write a number of files to the fill the (staging) disk that forces HSM to migrate the files out to tape. Any attempt to read the files back will force the HSM to migrate (in) the files to the staging disk. HSM migration parameters are measured by special tests that force migration of files from the staging disks to the tape using the vendor provided APIs.

The disk test is performed in the following way:

(1) Write a file to disk
(2) Read that file from disk
(3) Copy that file from disk to disk
(4) Read the copy

The tape test is performed in the following way:

(1) Write a file to tape
(2) Read the file from tape
(3) Copy the file from tape to disk
(4) Copy the file from disk to tape
(5) Read the copy from tape

Whenever a file is written, there are three words at the end of the file that are used to mark the file. These words contain a checksum generated when the file is first written, a file marker, and the length of the file. Whenever a file is read, except in the case of the copy operation, these three words are checked to make sure that the file has been properly identified and read.

When benchmark tests were first run, the effect of the file system memory cache was immediately evident. Small files remained in memory, and hence were "read" back very quickly. In order to minimize this effect, the benchmark was modified to do a pre-test preparation. This simply reads and writes a sufficient number of files to fill up the file system memory cache.

The benchmark may be run in one of two modes: rotational or sequential. In the first mode, a file is created and written to the disk. Then that file is read back in and checked. Next the file is copied, and afterwards the copy read and checked. This is repeated for the specified number of files, and then for each block size and file size combination specified. In sequential mode, however, all of the writes are done in succession for the specified number of files. Then they are all read back in and checked, next copied, and finally the copies read and checked. As expected, the effects of the file system cache are more evident

in rotational mode than in sequential mode, since in rotational mode there is no processing intervening between the write (or copy) and the subsequent read of a given file.

As the program executes, it collects statistics and writes them into a comma-separated-variable file that may be easily read by a spreadsheet program for further analysis.

Some Test Results

A fairly lengthy series of tests is required to properly exercise the HSM. Such a series of tests produces much more data than can be presented in this paper. For this reason, only a small, but representative, sample will be given here. The following charts show results for files of size 7,962,624 bytes, using a block size of 66,536 bytes. Both Rotational and Sequential mode test results are presented.

The short downward spikes on write, read, or copy operations are attributed to the file system cache. Periodically it fills up and some delay occurs while it is cleared. The upward spikes on the subsequent read operations may also be attributed to file system cache. The long downward spikes are attributed to the action of the HSM. Note that these spikes occur at regular intervals.

In Rotational mode, a file is written, then read, then copied to a new location, and then the copy read in again. This is repeated for the specified number of files. The HSM comes into play whenever a write or a copy causes the disk usage to reach the high water mark. Hence there are downward spikes either on the write or the copy. The reads, occurring immediately after the write or copy, are fairly fast, and the HSM does not come into play directly. However, the HSM may still be making more space on the disk at that time, and thus cause overall throughput to slow because of contention for the disk.

In Sequential mode, all of the files are first written. Hence the downward spikes occur whenever the disk usage reaches the high watermark. By the time a file has to be read, it is certain that it has been rolled out to tape. Hence the HSM is active in moving the file back to disk on every read. The same comment holds for the copy operation and the following read.

Additional tests will be run to determine the effect of changing the high and low watermarks.

Future Directions

Following the initial test of the benchmarks on PetaStore, we will be testing the code on other environments to test for portability. We will also begin including code for testing client-server connections to the mass storage system.

Acknowledgments

We thank Sony Corporation for the loan of the library unit and Petaserv; in particular, we are grateful to Frank Jones, Stu Reeder and Madhu Reddy for sharing their expertise, and for being available to answer our numerous questions.