

Disk I/O Considerations for SAS® Processing

John Hall, Hewlett-Packard Company, Cary, NC

Abstract

An understanding of the operation of Disk Input/Output (I/O) can be fundamental to a successful SAS processing environment. This paper explores various configuration options that can be used when designing a SAS processing server. The particular implementation discussed is a Hewlett-Packard server using the HP-UX operating system. We begin with a discussion of file systems implementations and explore attributes and benefits of each. We next focus on considerations for SAS server processing, including partitioning and striping. Real world examples are referenced to demonstrate our experiences in these areas.

General Concepts

It is necessary to understand some fundamental concepts to effectively use the various options available in creating and using file systems in HP-UX. Because the SAS system utilizes the underlying file system of the host OS, an understanding of the file system implementation on HP-UX becomes important in configuring a SAS server environment. The file system shipped with HP-UX has a default set of file and kernel parameters, which are generally acceptable for typical OLTP applications. Performance of Data Warehousing (DW) and Decision Support applications can usually be improved by modification of these defaults. Understanding the I/O needs of the particular SAS processing environment and properly configuring the system to meet those needs can be critical to a successful SAS implementation.

Logical Volume Manager (LVM) is a set of commands and underlying software that allows partitioning of disks in a more flexible and efficient manner than traditional disk sections. By using LVM, disk space may be allocated according to current anticipated needs, rather than the structural features built into disks. Should needs change, LVM allows logical reconfiguration of disks to meet changing space requirements.

At the heart of LVM are logical volumes, which are collections of pieces of disk space from one or more physical disks. Each collection, or logical volume, is constructed so that it appears to HP-UX as a single disk. Logical volumes also support increased data and system

availability options through a process called mirroring. Disk mirroring allows identical copies of important data to be maintained on the system that can be used in the event the original data becomes inaccessible. Logical volumes also allow disk striping, a process that can increase I/O performance by spreading data across multiple disks.

The file system buffer cache is used by HP-UX to manage the flow of data between main memory and secondary memory (disks). This is accomplished by setting aside a certain amount of system memory into a reserved area for use as a buffer cache. The buffer cache allows the system to optimize disk performance in two ways: it allows the I/O subsystem to make large disk transfers to and from the disk, and it allows applications to avoid reading or writing to disk if the data they want happened to already be in the buffer cache. On HP-UX, the amount of memory used by the files system buffer cache is dynamic; that is, it can change in size depending on system demand for virtual memory vs. buffer cache.

File System Types

HFS

HP-UX uses a file system known as the High Performance File System (HFS), which is also known as the McKusick file system. HFS is based on the BSD UNIX file system. HFS uses a system of cylinders divided into blocks and fragments to logically organize data on the physical disk. Information about location of data and other metadata is placed into superblocks, which are duplicated and spaced throughout the logical structures to provide fast access as well as redundancy of critical information.

VxFS

VxFS (also called JFS) is an extent-based journaling file system that offers fast file system recovery (as compared to HFS) and online features such as online backup, online resizing, and online reorganization. The implementation of VxFS on HP-UX is based on the version from Veritas Software Inc.

NFS

NFS services allow HP-UX to share files and directories with other systems over a local area network in a client/server approach. Through the use of configuration files, hosts and users may be granted access to files and directories on systems that have NFS capability.

Because of the performance implications and stateless nature of NFS, this type of file system has limited usage within the SAS processing environment.

CDFS

CD-ROM File Systems are used to read a CD-ROM with a file system on it. A CDFS must be mounted as a read-only file system, and is typically used for retrieval of data files that have been placed in secondary storage.

File System Attributes

Blocks and Fragments

Data to be transferred to or from the disk is organized in blocks. Blocks are further divided in smaller divisions called fragments. The size of blocks and fragments may be specified at the time of HFS file systems creation. Selection of the size of blocks and fragments represent a compromise between access speed and efficient use of disk space. Larger blocks enable greater access speed, but waste disk space. Conversely, smaller block sizes make more efficient use of disk space but break data into more pieces, requiring additional disk access. In a SAS processing environment, large files are quite the norm; therefore, building file systems with large (64k block/8k fragment) file system parameters can result in increased I/O performance.

Striping

Disk striping is a technique for increasing per-process disk performance. A striped file system is created in HP-UX by using LVM commands, either through the System Administration Manager (SAM) or through LVM commands from the shell interface. A striped file system is composed of two or more physical disk drives which have been configured into a volume group using LVM.

When using disk striping, data is set up such that what would otherwise be consecutive blocks of data on a single disk are intentionally and systematically placed on different physical disks. Thus, in a three-way striped logical volume, data will be spread across three disks, with each disk storing every third block of data (block 1 goes to

disk 1, block 2 goes to disk 2, block 3 goes to disk 3, block 4 goes to disk 1, etc.). The size of each of the blocks used in striping is referred to as the stripe size, and can be specified at the creation of the striped logical volume.

Disk striping using LVM can yield a performance increase to applications which read and write large, sequentially accessed files. Data access is performed in a parallel fashion over the multiple disks simultaneously, resulting in a decreased amount of time to perform an I/O operation. Disks which are combined in a stripe set should be of the same size and type if possible to reduce latency and obtain maximum benefit.

While disk striping does not require more disk space, it does require multiple disks. Because data is being distributed across multiple disks, the loss of a single disk in the striped volume set may result in the loss of portions of files in the striped logical volume. Attention should be given to location of various permanent and temporary storage areas when designing disk layouts.

Interfaces

Multiple disk interfaces can be successfully used to increase I/O efficiencies within a SAS server environment. By using multiple Fast/Wide (F/W) SCSI disk interfaces within a server, not only can file system capacity be increased, but performance can also be increased by more fully utilizing the total available bandwidth of the SCSI interface.

Combining multiple F/W SCSI interface adapters with disk striping can provide additional benefits, as disk access can be spread over multiple disk interfaces throughout the striped logical volume. This is a good strategy for SASwork areas, which are prone to considerable I/O pressure during DSS and DW projects.

Partitioning

Logically separating SASwork areas from permanent SAS dataset storage allow for different I/O systems to be utilized throughout SAS processing. Typically, there is a mix of I/O including access of permanent SAS datasets as well as the temporary SASwork areas throughout the execution of a SAS application. By separating the location of these data locations, the available I/O bandwidth is more fully utilized by the entire system. The execution of multiple SAS jobs on the server further extends this notion, with I/O being spread over the available disks and interfaces on the server.

Dynamic Buffer Cache

The buffer cache as implemented on current versions of HP-UX is dynamic in nature. It can grow and shrink in size, depending on operating system and virtual memory needs. Both the file system buffer cache and virtual memory subsystem access the same body of RAM in main memory. The ability to dynamically alter the amount of memory used by the buffer cache and virtual memory subsystem provides for the optimization of available system RAM. The dynamic buffer cache provides HP-UX the flexibility to accommodate both smaller application programs that do a lot of I/O as well as larger programs that do limited I/O but have requirements for large amounts of memory.

Limits on how far the file system buffer cache may be allowed to grow and shrink are available through HP-UX kernel parameters `DBC_MAX_PCT` and `DBC_MIN_PCT`. By default, these parameters are set to 5% and 50%, respectively. The file system buffer cache may be configured at a specific value by setting the kernel parameters `NBUF` and `BUFPAGES` to the desired fixed values.

SAS System Example

Hewlett-Packard has set up a Competency Center (CC) within the SAS Customer Technology Center (CTC) to assist in system sizing and configuration discussions as well as perform proof-of-concept and software evaluations using customer application code. These exercises have led to "real-world" examples of how modification of default file system parameters can have a positive impact on SAS processing. Here is an example of a recent evaluation which illustrates the effect of changing the default block and fragment sizes of a HFS file system.

The test was set up to evaluate the effect of increasing the default block and fragment size of the directory containing the SASwork directory. A SAS program was selected which performs various operations on an approximately 250 MB SAS dataset. The SASwork directory was configured for all testing over a striped volume set with 10 physical disks which is across three F/W SCSI interfaces.

The first run (Case A) was executed with the default 8K block / 1K fragment file system. The second run (Case B) changed to a 64K block / 8K fragment for the SASwork directory. Testing was performed under controlled conditions to assure the only changes in system configuration were the block/fragment file system parameters. The test case was submitted in batch and only

a single instance of each job was submitted at any one time.

The actual performance increase by making only this change was 12.64% in overall job execution time. Additional gains in overall SAS processing time were achieved by combining this modification with other changes to file system parameters. These changes included using JFS file systems and modification of other parameters, which will be discussed in future papers.

Tricks and Tips

When configuring a SAS server where I/O is a primary consideration, here are a few tips that may be useful in configuring and maintaining a robust SAS processing environment:

- Keep pathnames and filenames as short as practical. File paths are cached within HP-UX and the smaller the pathname, the more entries fit within the cache. Don't get carried away, however - a directory or file name can still be descriptive and relatively short.
- Use performance tools to establish a 'baseline' of system performance, paying particular attention to I/O related parameters such as file system buffer cache utilization and I/O by file system. HP performance tools such as GlancePlus/UX provide real-time monitoring capability over a wide range of system performance areas.
- Configure specific directories as SASwork areas that are optimized for performance. Separate SASwork areas from permanent SAS dataset storage areas and operating system file systems.
- In a multi-user environment, consider segmenting users into groups based on their I/O processing needs. For example, if there are just a few SAS users who do heavy I/O processing, break them into a group and provide them their own SASwork area. This can be accomplished by specifying a specific location for the `-saswork` parameter in a `config.sas612` file for these users and placing the file in the user's home directory. This strategy can help distribute the I/O load on the system and allow system administrators to optimize particular disk configurations best suited for the environment.

Summary

There are many advantages that can be realized by modification of the default file system parameters in a SAS processing environment. Optimization of the I/O systems within the server can provide benefits to the SAS user community (faster turnaround of SAS jobs) as well as system administrators (better system performance).

Planning and implementation of file systems should be a part of an overall strategy for system administration of a SAS processing environment. The benefits derived by optimization of file systems will be amplified as the processing requirements increase in the SAS application environment.

Trademarks

SAS is a registered trademark of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

HP-UX is a registered trademark of Hewlett-Packard Company.

Other brand or product names are registered trademarks or trademarks of their respective companies.

References

Hewlett-Packard Company (1995), *HP-UX System Administration Tasks*, Palo Alto, CA: Hewlett-Packard Company

Hewlett-Packard Company (1992), *How HP-UX Works: Concepts for the System Administrator*, Palo, CA: Hewlett-Packard Company

Loukides, M. (1991), *System Performance Tuning*, Sebastopol, CA: O'Reilly & Associates

Madell, T. (1997), *Disk and File Management Taskson HP-UX*, Upper Saddle River, NJ: Prentice Hall.

Vahalia, U. (1996), *UNIX Internals The New Frontiers*, Upper Saddle River, NJ: Prentice Hall.

Author Information

John Hall
Technical Consultant
Hewlett-Packard Company
P.O. Box 37000
Raleigh, NC 27627
(919) 460-2390
j_hall@hp.com

Copyright 1997, Hewlett-Packard Company