Updated March 2011

Storage best practices: SAS 9 with IBM System Storage and IBM System p

Considerations for optimal storage layout
Version 3.0

SAS / IBM authors

Margaret Crevar. *SAS Manager, SAS Performance Lab*
Leigh Ihnen, *SAS Solutions Architect, SAS Enterprise Excellence Center*
Harry Seifert, *IBM Senior Certified IT Specialist, Solutions Technical Sales*
Frank Bartucca, *IBM Senior Engineer, Systems and Technology Group*
Vishwanathan Krishnamurthy, *IBM Technical Consultant, Systems and Technology Group*
Frank Battaglia, *IBM Certified IT Specialist, Systems and Technology Group*

Table of Contents

Introduction

Repeatedly IBM® and SAS have successfully teamed for sizing efforts, reference architectures, and customer opportunities. The combined SAS and IBM teams recognized the customer needs to better understand SAS specific I/O characteristics and the related optimal physical and logical IBM System Storage® configurations. This paper gives an overview of SAS specific I/O characteristics and provides IBM System Storage recommendations. The recommendations are specific to SAS 9 on IBM AIX® with members of the IBM System Storage DS® series, IBM System Storage SAN Volume Controller, and IBM XIV® Storage System as noted in the paper.

"SAS is a leader in data mining, predictive technologies and analytic applications."[1] SAS has many product solutions. This paper gives a high-level SAS solution overview but does not attempt to describe I/O characteristics of each solution in detail. Rather, this paper attempts to provide guidance with the identification of SAS I/O characteristics. Some SAS solutions and their specific I/O characteristics with IBM System Storage are used as examples. This paper does not describe installation steps for SAS or for the IBM OS or hardware

Technical team

This paper is an on-going collaborative effort by the SAS Enterprise Excellence Center (EEC), SAS Research and Development, and the IBM Systems and Technology Group. The paper draws from testing results and the knowledge gained from various SAS and IBM sizing and reference architecture projects conducted by SAS and IBM at both the IBM Beaverton Benchmark Lab and at SAS Institute in Cary, North Carolina.

Thanks to the contributing team members:

Leigh Ihnen	SAS Solutions Architect, SAS Enterprise Excellence Center
Margaret Crevar	SAS Manager, SAS Performance Lab
Harry Seifert	IBM Senior Certified IT Specialist, Solutions Technical Sales
Scott Fadden	IBM GPFS Performance Engineer, Storage Group
Brian Porter	IBM IT Specialist, Advanced Technical Skills Solution Team
Frank Bartucca	IBM Senior Engineer, Systems & Technology Group
Vishwanathan Krishnamurthy	IBM Technical Consultant, Systems & Technology Group
Frank Battaglia	IBM Certified IT Specialist, Systems & Technology Group

Also many thanks to the extended SAS Institute Team for environment setup assistance and validation help, as well as for the many hours of running tests and documentation review.

[1] Gartner, December 20, 2010

Components of SAS Business Analytics Framework

The SAS Business Analytics Framework encompasses full range of world-class solutions, software, and services – enabling the strategic business decisions that optimize performance across your organization. The SAS Business Analytics Framework includes an expanding range of industry and line-of-business solutions as well as award-winning technologies you can put to work by themselves or as part of your purpose-built solutions.

Business solutions – Address critical business challenges unique to your industry or across your lines of business, using solutions that incorporate the strengths of SAS technologies and the industry domain expertise. You can find more information at: http://www.sas.com/software/index.html

Reporting – Get answers to more sophisticated questions, format presentation-quality results, and easily share findings across your enterprise. You can find more information at: http://www.sas.com/technologies/bi/entbiserver/index.html

Analytics – Move from reactive to proactive decisions using the predictive analytics capabilities that only SAS provides. You can find more information at: http://www.sas.com/technologies/analytics/index.html for more information

Data integration – Fully leverage all the data flowing into your business to uncover hidden insights and increase your competitive edge. You can find more information at http://www.sas.com/software/data-management/data-integration/index.html

Through the SAS Business Analytics Framework, you can address your most-critical business issues right now and then add new functionality over time, enabling continuous performance improvement (http://www.sas.com/solutions/performance-management/) across your organization – all from one vendor, all through one framework, reducing your total cost of ownership. Through the combined strengths of the platform for SAS Business Analytics (data integration, analytics, and reporting (http://www.sas.com/businessanalytics/index.html) – you can gain ultimate flexibility in responding to changing business needs. It is possible to achieve more rapid results and build a technology platform over time that is custom-tailored to your business.

Planning SAS storage subsystems

The following guidelines are provided to help in the planning of storage subsystems for SAS deployments.

SAS I/O characteristics

SAS has many application solutions. The foundation component, known as base SAS, runs as a collection of processes. This makes SAS different from a traditional relational database management system (RDBMS) where there is only a single process running at any given time. Generally, each SAS user starts their own SAS session (also known as process) for each SAS job or application they are running. With the new SAS 9 Business Intelligence (BI) Architecture, there are also several SAS servers that are started to support the Java™ applications, but the tendency is for each active SAS user to have their own back-end SAS server or process running.

As previously stated, SAS does not behave the same as an RDBMS. Here are some of the characteristics of SAS software. SAS:

- Performs large sequential reads and writes. Some of the new SAS BI applications do some random access of data, but for the most part, the SAS workload can be characterized as predominately large sequential I/O requests with high volumes of data.
- Does not preallocate storage when SAS initializes or when performing writes to a file. When SAS creates a file, it allocates a small amount of storage, but as the file grows during a SAS task, SAS extends the amount of storage needed.
  Note: File extension is limited to the amount of space available within the file system.
- Uses OS file cache for data access. SAS does not do direct I/O by default.
- Creates large number of temporary files during long running SAS jobs in the SAS WORK directory. These files are created, potentially renamed towards the end of the task, deleted, and/or manipulated potentially hundreds of times within a long running SAS ETL job. The size of the files might be very small (less than 100 MB) to larger (in the 10s of GBs).
- Creates standard OS files for its data store.

Storage considerations for base SAS usage

Here are some general tips for setting up the file systems required by many SAS applications. Note that these tips are very general in nature. A specific SAS application or SAS Solution might require more file systems than that are listed in this section. Also, the exact configuration of the various file systems depends on the SAS usage and the underlying data model. But for the purpose of this paper, here are some general guidelines for setting up the file systems required by basic SAS applications.

It is generally recommended that a minimum of three file systems be set up to support SAS. In an ideal world, the SAS file systems would be their own independent set of physical disks. Use the characteristics and locations of the following SAS file systems as a

reference, especially if you need to share the disk. The system administrator or installer has to avoid sharing these heavy I/O file systems with other applications performing heavy I/O or different I/O tasks from how SAS normally works.

SAS files and file systems include:

- **Root Operating System** - location for the operating system and swap files
- **SAS Executables** - these could be placed with the operating system file systems
- **SAS Data** - location for the permanent SAS data files and raw input data. Mostly SAS data mart reads. There will also be writes when the data mart is refreshed, and some occasional writes at the end of some SAS jobs.
- **SAS WORK** - temporary space for SAS sessions. The data here is available only during the duration of a SAS session and is erased when the SAS session terminates normally. This file system gets the majority of the I/O activity as this is where the temporary files are created during a SAS job. **Note:** Some of the I/O pressure for the utility files created by the threaded procedures can be alleviated by pointing the UTILLOC parameter to a different file system.

Recommended Redundant Array of Independent Disks (RAID) configurations for each file system include:

- **Operating System** – Mirror (RAID1) this file system to ensure the high availability of the hardware.
- **SAS Executables** – Mirror (RAID1) this file system, which can be placed with the operating system file systems.
- **SAS Data** - most SAS users want this to be a redundant file system to ensure the availability of the SAS data. RAID10 in general gives the best redundancy and performance, but it doubles the number of disks. Most users go with RAID5 for this reason. Make sure that you mirror the disks before you stripe them for a more-reliable setup.
- **SAS WORK** - In the past, SAS recommended the striping of the SAS WORK file system (RAID0) without redundancy and high availability for the best performance. This was rationalized because the files created in this file system are temporary in nature and cannot be re-accessed if the file system or SAS session crashes. However, many customers now have the requirement of a highly-available SAS WORK file system. Once again RAID10 generally gives the best redundancy and performance for a highly-available storage solution. In addition, RAID5 (especially on the storage arrays that have cache) is also very popular and performs as well as RAID10 configurations.

Choosing the number of disks in a RAID5 array is a tradeoff between performance and the amount of time it takes to reconstruct a disk after a disk failure. SAS applications can be both bandwidth intensive and require a high number of input/output operations per second (IOPS). Typically a SAS LIBNAME maps to a directory in a single file system. If the I/O bandwidth or I/O per second requirement would necessitate using more disks than is prudent given recovery concerns, use Logical Volume Manager (LVM) striping to create a file system that spans multiple RAID5 volumes. In most SAS configurations, the SASWORK directory is shared by all users on a host system. This can create a need for a file system that has high I/O bandwidth. Some storage arrays have a maximum number of disks that can be configured in a single RAID5 volume. LVM striping can be used to increase I/O bandwidth and IOPS for a file system.

These are general guidelines regarding the I/O needed for SAS. More specific guidelines regarding how to set up the file systems might require a deeper understanding of the specific SAS application and the data model that will be used. **The main concern when setting up a file system is to ensure that SAS gets the I/O throughput bandwidth needed to complete the SAS jobs in the timeframe required by the SAS users.**

SAS and direct I/O

As mentioned earlier, SAS accesses SAS data files through the operating system's file cache. At times, this can cause performance degradation. Starting with SAS 9.2, a user has the option to directly access SAS data files with the new DIRECTIO functionality. To understand how to use this new SAS 9.2 feature, review the *Improving SAS® I/O Throughput by Avoiding the Operating System File Cache* SAS Global Forum 2009 paper at http://support.sas.com/resources/papers/proceedings09/327-2009.pdf

Prior to SAS 9.2, SAS users were able to do direct I/O only if the enhanced journaled file system (JFS2) is mounted with the **dio** option. Every file in the file system will be read using direct I/O regardless of its size or the I/O transfer size (SAS BUFSIZE option). In general given the mixture of bufsizes and dataset sizes in a SAS library using direct I/0 results in poorer performance. The exception case is when the application uses data sets which are too large to fit in file cache.

A best practice for when to use direct I/O to achieve better system-wide performance is:

- The datasets are larger than file cache
- The datasets are in a separate file system without any other SAS data objects
- The datasets are accessed sequentially
- The BUFSIZE of the data sets is a multiple of 128 KB
- The datasets are not concurrently accessed by multiple SAS sessions with the intention of using file cache to minimize physical I/O

Additional key points about SAS I/O

SAS tasks involving large amounts of data can present different processing and access patterns, creating multiple challenges for system designers. However, some generalizations can be made about SAS data management:

1. SAS usually does large, blocked I/O, as compared with a classical online transaction processing (OLTP) environment. In general, SAS I/O measurements typically focus more on megabytes per second (MBps) rather than IOPS.

   When feasible, it is a good idea to match the choice of BUFSIZE for a data set to the underlying I/O architecture. The degree to which sequential access is the predominant I/O pattern determines the benefit of using a large BUFSIZE. If indexes are used to facilitate random access, unless the index has good locality, using a large BUFSIZE might decrease performance as larger amounts of data need to be read per request. Typically, choosing BUFSIZE involves selecting a value that accesses a complete stripe width for the I/O device. In particular, when RAID5 is the data availability strategy,

matching the BUFSIZE with the stripe size minimizes the cost of calculating parity. (Stripe size is the amount of data that is written to a device. Stripe width is the number of devices involved in the stripe set times the stripe size.) Accessing data in stripe width amounts allows all devices in the stripe to be concurrently used, which typically generates the highest I/O bandwidth from the device.

2. SAS I/O buffer sizes are fixed upon creation of the file. Therefore, optimizations to data movement need to be external to the SAS process (for example, OS kernel and file system parameters).

   If the SAS data set has been created, then changing the settings that control the preferred I/O sizes for the file system might provide performance improvement. Typically, this involves setting the degree of aggressiveness for the OS when doing sequential read-ahead/write-behind and caching of data for reuse on the application's behalf. However, you need to consider both total system throughput as well as individual job throughput. Aggressive settings that maximize individual performance might cause system-wide degradation when the system is under full load.

3. SAS data sets and associated files are built within the confines of the underlying OS, and can be managed by file management utilities which are a part of the OS (or might be a part of optional products). This also means that file placement can be determined by the definition of directory structures within the OS.

   For example, in the UNIX® environment, if there is contention on a file system for multiple SAS data sets, because SAS data sets are OS objects, OS commands can be used to move the data set to a different device and create a symbolic link to a new location in the current directory. This is an example of after-the-fact performance tuning that is available. **Experimental note:** On UNIX, after index creation, this method might work to separate index and data, which is a typical RDBMS tuning tip. SAS indexes are opened using the physical path, where the SAS data set was found. As a result, using concatenated libraries to separate indexes and data does not work.

4. As previously stated, different types of SAS tasks have different I/O patterns. Classical Decision Support Systems (**DSS**) and Data Warehousing (**DW**) might be highly sequential, while On Line Analytical Processing (**OLAP**) environments might utilize multiple readers and writers (when accessing a sub-cube that does not exist) in a more random access pattern across multiple file systems.

5. SAS, as a rich discovery environment for BI and analytical solutions, creates many temporary files of unknown size. Both, the creation of many small temporary files and very large sequentially accessed temporary files, need to be supported by the storage subsystem and OS. In both cases, the size of the temporary file is not known at creation time.

Because SAS I/O does not use preallocated storage but is block-based in user specifiable sizes, the SAS application need to create its own extents in a **pay-as-you-go** design. This design differs from the more traditional preallocation strategies employed by some traditional RDBMS applications. As a result, contention for file system metadata (file system block to physical device volume mapping) can become a limiting factor for achievable I/O bandwidth. A possible solution is to use a file system that allows for the configuration of large fixed extents. While this might be wasteful in terms of unused space and when dealing with large objects the percentage of waste might be small.

Note: This is really an explanation of the consequences of 4 and 5, rather than a separate point.

I/O patterns/Usage cases

- Acquisition of source data for the SAS data warehouse is typically sequential and for very granular data such as customer transactions, store inventory, or store sales can be quite large, in 10s of gigabyte range, exceeding the size of the file cache. These are now often stretching into the 100s of GB and TB range as well.

- Data profiling and hygiene operations on source data are typically implemented using sequential I/O access to the source files. For performance, the data in the source files might be sorted to facilitate cross-source validation. For example, a record for an item sold at a store must have a matching store inventory record.

- When loading data from the SAS data warehouse into a dimensional model, the uses of small look-up tables and SAS format catalogs is common and ensures that the tables are in the file cache. This can increase the warehouse load performance.

- Query and reporting against a dimensional model typically involves random access of both, fact and dimensional tables. For the enterprise-scale solution, the likelihood of having significant cache hit rates on accessing fact tables randomly is very low. If the typical query patterns involve dimension or look-up tables, which are small, and if the tables are cached, query extraction performance will be improved.

- Extracting data from the warehouse for an analytic exploitation data mart typically involves the same random access patterns as query and reporting applications.

- Building the exploitation analytic data mart typically involves sorting and sequentially accessing the extracted data from the warehouse and sequentially writing to the analytic data mart. The volumes of data might be small enough that file cache provides significant performance improvements depending upon how the analytic data mart is refreshed and the physical data model.

- Analytic modeling of the data in the exploitation data mart and scoring of models is typically a sequential process.

When to utilize various storage systems

Storage can be provided to the SAS application by many different methods. In this section you can examine a few of those methods, such as internal server storage disk, directly-attached storage area network (SAN) storage, and Fibre Channel (FC) switch attached SAN storage.

Internal disk

Internal storage is traditionally used for operating system files and executables, application executables, and swap/paging space. Data high availability, disk capacity,

scalability, performance, and storage sharing must be considered when determining the placement of SAS data, SAS executables, and SAS temporary work space. Small test or development SAS environments might only warrant the use of internal drives. However, in general, the larger SAS deployments with production mission critical data use the internal drives only for SAS executables, paging, and O/S related file systems. These same environments usually place SAS file systems with SAS data and temporary workspace on SAN-attached storage.

Storage area network

SAN storage provides a highly-available and high-performance storage option for SAS applications. Typically SAN storage is shared between multiple application servers and/or multitier applications. In the case of a single or multitier SAS application, the SAN storage can be Fibre Channel (FC) direct attached to the application server or FC switch attached. An example of this would be SAS deployed on an IBM System p® server with logically partitioned processors, memory, and adapters into multiple logical partitions (LPARs). As previously stated, it is highly recommended that any competing workloads be isolated from the SAS application deployment as this storage contention can severely impact application performance for both SAS and the non-SAS application. This is true for switch-attached as well as directly-attached SAN storage.

SAN Volume Controller

An adaptive SAS enterprise computing environment requires a centralized storage solution utilizing a SAN. Typically, SAN storage is dedicated to application servers and multitier applications. The various dedicated SANs isolate the storage by physical server environments, often by mixed vendors, and increase management and maintenance costs of the storage hardware and software.

IBM System Storage virtualization products such as the IBM System Storage SAN Volume Controller achieve the abstraction from the physical volumes of data storage to a logical level. It addresses the increasing complexity of managing storage, while reducing the associated costs. Its main purpose is the full exploitation of the benefits promised by a SAN. Virtualization enables data sharing, ensuring higher availability, providing disaster tolerance, improving performance, allowing for consolidation of resources, providing policy-based automation, in addition to other benefits, which do not automatically result from the implementation of today's SAN hardware components.

Storage virtualization is possible on several levels of the storage network components, meaning that it is not limited to the disk subsystem. Virtualization separates the representation of storage to the operating system and its users from the actual physical components.

Storage virtualization accumulates the storage into storage pools, which are independent of the actual layout of the storage (that is, the overall file system structure). Because of this independence, new disk systems can be added to a storage network and data can be migrated to them, without causing disruption to applications. As the storage is no longer controlled by individual servers, it can be used by any server as needed. In addition, it can allow capacity to be added or removed on demand without affecting the application servers. Storage virtualization will simplify storage management, which has been an escalating expense in the traditional SAN environment.

In the case of a single or multitier SAS application, the storage can be made available to all application and database servers by the IBM System Storage SAN Volume Controller SAN Volume Controller provides a highly-flexible virtual storage option for SAS applications.

The IBM System Storage SAN Volume Controller is an in-band, block-based virtualization product that minimizes the dependency on unique hardware and software, decoupling the storage functions expected in a SAN environment from the storage subsystems and managing storage resources.

In a typical nonvirtualized SAN, shown in the left hand side of Figure 1, servers are mapped to specific devices, and the logical unit numbers (LUNs) defined within the storage subsystem are directly presented to the host or hosts. With the SAN Volume Controller, servers are mapped to virtual disks, thus creating a virtualization layer.

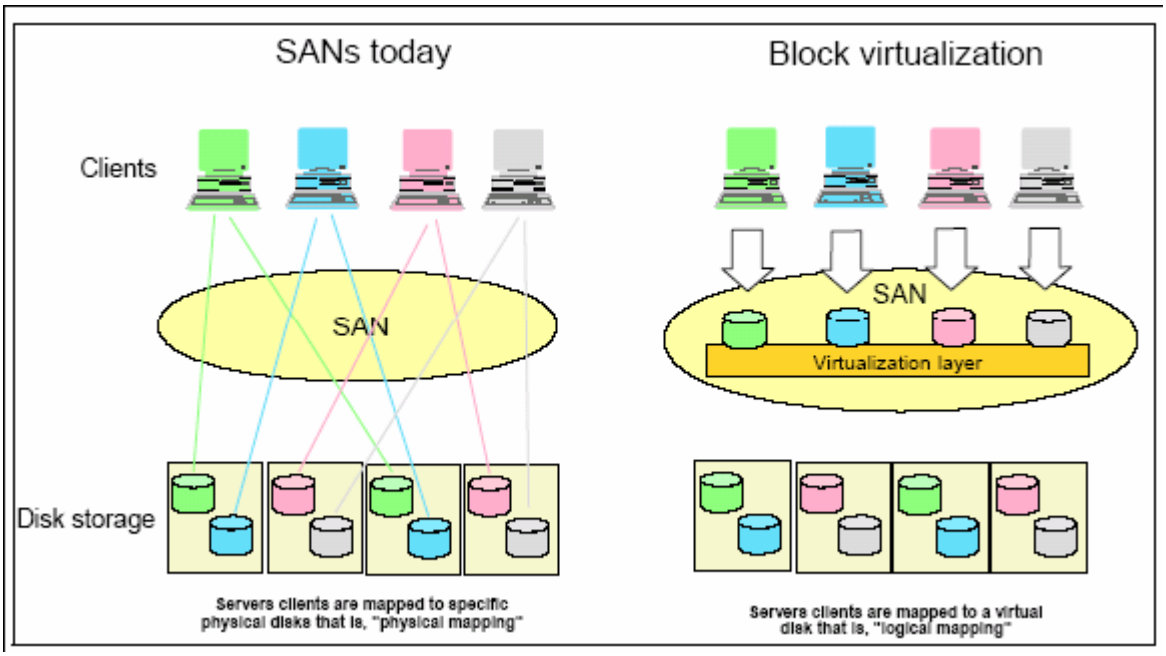

Figure 1: SAN Volume Controller block virtualization

Each SAN Volume Controller consists of one or more pairs of engines, each pair operating as a single controller with failover redundancy. A large read/write cache is mirrored across the pair, and virtual volumes are shared between a pair of nodes. The pool of managed disks is controlled by a cluster of paired nodes.

The SAN Volume Controller is designed to provide complete copy services for data migration and business continuity. As these copy services operate on the virtual volumes, dramatically simpler replication configurations can be created using the SAN Volume Controller, rather than replicating each physical volume in the managed storage pool.

The SAN Volume Controller improves storage administrator productivity, provides a common base for advanced functions, and provides for more efficient use of storage.

The SAN Volume Controller provides block aggregation and volume management for disk storage within the SAN. In simpler terms, this means that the SAN Volume Controller manages a number of back-end storage controllers and maps the physical storage within those controllers to logical disk images that can be seen by application servers and workstations in the SAN. The SAN is zoned in such a way that the application servers cannot see the back-end storage, preventing any possible conflict between SAN Volume Controller and the application servers both trying to manage the back-end storage.

Note: The preferred node by no means signifies absolute ownership. The data is still accessed by the partner node in the I/O group in the event of a failure or if the preferred node workload becomes too high.

Note: Write-through mode is where the data is not cached in the nodes, but written directly to the disk subsystem instead. While operating in this mode, performance is somewhat degraded. However, more importantly, it ensures that the data makes it to its destination without the risk of data loss to which a single copy of data in cache might expose you.

When an application performs I/O to a virtual disk (VDisk) assigned to it by the SAN Volume Controller, it can access that VDisk through either of the nodes in the I/O group. Each node can only be in one I/O group, and as each I/O group has only two nodes, the distributed redundant cache design in the SAN Volume Controller needs to be only two-way.

The SAN Volume Controller I/O groups are connected to the SAN in such a way that all back-end storage and all application servers are visible to all of the I/O groups. The SAN Volume Controller I/O groups see the storage presented to the SAN by the back-end controllers as a number of disks, known as managed disks (MDisks). Because the SAN Volume Controller does not attempt to provide recovery from physical disk failures within the back-end controllers, MDisks are usually, but not necessarily, part of a RAID array. The application servers do not see the MDisks at all. Instead, they see a number of logical disks, known as virtual disks or VDisks, which are presented to the SAN by the SAN Volume Controller.

MDisks are collected in groups, known as MDisk groups. The MDisks that are used in the creation of a particular VDisk must all come from the same MDisk group. Each MDisk is divided into a number of extents (default minimum size of 16 MB and maximum size of 512 MB), which are numbered sequentially from the start to the end of each MDisk.

IBM DS series storage and IBM XIV Storage Systems:

The IBM System Storage Disk Systems products and offerings provide storage solutions with superior value for all levels of business from small and medium business (SMB) to high-end and enterprise systems.

IBM System Storage DS8000® series offers high-performance, high-capacity, secure storage systems that are designed to deliver resiliency and total value for the most demanding, heterogeneous storage environments.
IBM XIV® Storage System is a ground breaking, high-end, open disk system designed to support business requirements for a highly available information infrastructure. The XIV system architecture is a grid of standard Intel® and Linux® components connected in any-to-any topology by means of massively paralleled, non-blocking Gb Ethernet, providing outstanding enterprise-class reliability, performance, and scalability.

IBM System Storage DS6000™ series is designed to provide high availability and high performance in a small, modular package. This series, along with the DS8000 series, offers an enterprise-class continuum of storage systems with shared replication services and common management interfaces.

IBM System Storage DS3000 and DS5000 series are entry and midrange and storage system offering in IBM's family of DS series storage.

For a complete and current list of IBM System Storage offerings, visit IBM System Storage website at **ibm.com**/systems/storage/disk/index.html.

IBM General Parallel File System

IBM General Parallel File System (IBM GPFS™) is a critical component for SAS grid solution deployments. GPFS is a high-performance shared-disk cluster file system that provides file system services to parallel and serial applications. GPFS allows parallel applications to simultaneous access a single file or multiple files, from any node in the GPFS cluster, while managing a high level of control over file system operations. GPFS is particularly appropriate in an environment where the need for data bandwidth exceeds the capability of a distributed file system server. In addition to high-speed parallel file access, GPFS provides fault tolerance, including automatic recovery from disk and node failures.

GPFS provides a flexible virtual storage option for distributed SAS applications that require high-performance data access. GPFS currently powers many of the world's largest scientific supercomputers and commercial applications requiring high-speed access to large volumes of data.

GPFS allows shared file access within a single GPFS cluster and across multiple GPFS clusters for users. A GPFS cluster consists of:

- AIX nodes, Linux nodes, or a combination thereof. A node may be:
  - An individual operating system image on a single computer within a cluster.
  - A system partition containing an operating system.
- One or more shared disks, which are defined in GPFS as Network Shared Disks (NSDs)
- A network for GPFS communications allowing a single network view of the configuration. A single network is used for GPFS communication.

GPFS software can be added to existing analytic servers and coexists with local file systems. This allows migration to GPFS and the use of multiple file system types for SAS solution.

This paper does not cover the installation, implementation, or administration of GPFS, the creation and management of storage pools, or definition of file placement policies. For these topics and tasks refer to the GPFS product documentation at:
http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp?topic=/com.ibm.cluster.gpfs.doc/gpfsbooks.html
or GPFS in IBM Redbooks® at:
ibm.com/redbooks/cgi-bin/searchsite.cgi?query=GPFS

Tuning for different operating system environments

During deployment of SAS specific operating system tunable parameters need to be considered for optimal storage performance in addition to general SAS tuning parameters. In the following sections, IBM AIX® storage tunable parameters are examined.

AIX tuning storage recommendations

This section details SAS 9 storage-specific AIX tunable parameters also found in the *SAS AIX5L, AIX6 and AIX7 Tuning Guide* at:
ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101529

There are many nonstorage related SAS tunable parameters that are not mentioned in this paper that should be examined for optimal SAS application performance. These AIX nonstorage and storage-related tunable parameters can be found by referencing *SAS AIX5L, AIX6, and AIX7 Tuning Guide* in detail. The following section includes storage configuration guidelines.

General storage guidelines

- Use JFS2 on 64-bit AIX kernel.
  - With the introduction of the IBM AIX 5L™ operating system, IBM introduced a new file system referred to as enhanced JFS (JFS2) that provides greater scalability than journaled file system (JFS). JFS2 is designed and optimized for a 64-bit kernel environment taking full advantage of 64-bit functionality. JFS2 is the default file system for a 64-bit kernel.
  - In general, SAS requires large rates of sequential disk I/O. The AIX file system named JFS2 can detect and exploit the read-ahead and write-behind characteristics of the application under normal file caching policy.
  - You can select **3. Enable 64-bit kernel** and **4. Create JFS2 File Systems** on **Install Option** screen during AIX installation (for example, from CD).

- It is strongly recommended that SAS WORK or UTILLOC point to a more robust file system than /var.
  - AIX instructions use /var for various purposes, such as storing temporary files, mail spool files, and all security logging information. Run out of /var space might cause SAS processes to terminate abnormally.

- Configuring paging space at least with the following suggestions:
  - Place paging spaces on dedicated disk or disks to eliminate I/O contention.
  - Use multiple paging spaces spread over multiple disks.

  - Make the primary paging space a little bigger than the secondary paging spaces.
  - Insure that the paging space is sufficient to support the number of concurrent SAS processes as the number of SAS processes can be dynamic depending upon the application workload.
  - Set *nokilluid=10* with *vmo*
  - strict_maxclient is 1 by default. Disabling it by setting to 0 will help remove the hard limit on how much of RAM can be used as a client file cache.
  - Set lru_poll_interval, which determines the interval in milliseconds, at which LRU page replacement daemons poll for off-level interrupts. Default: 10 milliseconds. Possible values: 0 through 60,000 (1 minute).

    An LRU page replacement daemon blocks low priority interrupts while running on a processor. If this option is enabled, LRU page replacement daemons will process pending interrupts at the designated interval. On a heavily-loaded system with large amounts of I/O, enabling this option can improve I/O throughput as I/O interrupts do not have to wait for LRU page replacement daemons to complete their processing.

- Determining the application's I/O access patterns is important for I/O layout and tuning.
  - To achieve the best I/O performance, the access patterns and storage configuration should be compatible. If the application's I/O patterns are not known, then additional data mgiht be gathered to determine dominant patterns. For example, in the experiments carried out by the test team, AIX trace indicates that SAS Revenue Optimization application drives traditional large sequential I/O characteristics but it also contains a fair amount of random I/O. Thus optimization for different I/O access patterns (dominant and nondominant) is recommended.

- Ensure tuning from a system-wide perspective (for example, VMM, LVM, FS, disk storage) for SAS workload.

- Use the appropriate number of host bus adapters from the storage to the host server to provide the required front-end application bandwidth.
  - Many SAS I/O workload patterns can be throughput intensive. However, this is not always the case for all SAS applications or necessarily true while running the entire SAS application.
  - High-performance storage channels, such as Fibre Channel technology need to be considered over slower mediums.
  - Use dynamic multipathing, if possible, to spread the I/O load over multiple adapters. Care needs to be exercised when locating SAS data libraries on mount points.

- Spread the I/O workload across many physical disk spindles rather than fewer larger-capacity disks.
  - Provide better I/O performance by sizing for quantity of disks instead of capacity of disks.
  - Implement storage system RAID striping across multiple physical disks. **Note:** In general testing, it has been observed that there is a slight performance advantage to using RAID10 over RAID5 for SAS temp space file systems. This is not necessarily the case for other SAS file systems. Use RAID10 or RAID5 depending on the level of redundancy and total capacity compared to usable capacity that is required for each type of file system.

  - Use LVM stripping instead of concatenation.

- Minimize disk contention between SAS temporary space and data spaces.
  - Avoid disk contention by placing SAS temp space file systems and SAS data file systems on physically separate disks.
  - Use multiple storage server controllers to further separate and isolate the I/O traffic between SAS temp and data spaces. This also provides a more robust disk back end to handle I/O tasks.
  - Use multiple mount points for SAS file systems. Place system O/S, SAS, user, SAS temp, and SAS data file systems on separate physical disk.
  - Separate the single SAS temp space file system (SAS WORK) into separate SAS temp file systems with physically separate disks **if** multiple users otherwise might have to share the SAS temp space (SAS WORK) **and** sharing the disk or file system increases disk contention beyond acceptable response levels.
  - Create separate JFS2 log files on separate physical disk fors each SAS file system.

- Isolate SAS I/O from non-SAS workloads.
  - In general, SAS applications can be highly sequential large I/O workloads. Disk contention between SAS applications and other non-SAS small I/O random IOPS applications increases service times of all applications and decreases I/O performance.

- Use the AIX Scalable volume group or Big volume group with *mklv –T 0* option to avoid the logical volume control block reserve of the first 4K of space.
  - With the LVCB present, the first data block starts with a 4 K offset.
  - When LVCB's exist on a lv they can cause I/Os to span multiple physical volumes due to this offset.

- Be mindful that AIX file systems are aligned on a 16 K boundary when choosing the disk stripe, segment size, or array stripe size.
  - A strip is the size of data to be written to each physical disk in the array. A stripe is the size of the full write across all the physical disks in the array. Example: strip size x number of disks = stripe size.
  - Note that the AIX LVM stripe size that can be selected from the *smit lv* create panel is actually the single strip size (not stripe) or size of data to be written to each of the array disks and not the full stripe size across all the physical disks.

- Synchronize SAS BUFSIZE with the storage system stripe size and the AIX LVM stripe size (if using LVM striping), and VMM read-ahead increments.
  - Synchronizing I/O sizes results in more efficient I/Os while reducing the total number of I/O requests to the storage subsystem.
  - **Note:** LVM striping may or may not provide better performance depending on the SAS application or the storage subsystem configuration. Testing your specific application is recommended.

AIX JFS and JFS2 tuning

The AIX file system is called Journaled File System (JFS) or enhanced Journaled File System (JFS2). AIX file system presents a logical view of files and directories linked together to form a hierarchical tree structure.

In general, SAS applications have a great deal of large sequential read and write disk I/O. If the workload has many large I/Os to a file system (for example, large sequential I/O to JFS2), the I/Os might be bottlenecked at the file system level while waiting for a construct called bufstructs. The bufstructs for JFS2 is dynamic and the number of bufstructs per file system can be increased. The file system must be remounted for the new value to take effect.

The I/O characteristics of SAS usually create the situation where VMM read-ahead, and write-behind algorithm can be used to improve the performance of sequential file access. The parameters listed in ***Table 1*** can be tuned using the *ioo* command.

Parameter	Description
j2_dynamicBufferPreallocation	This tunable (16 by default) specifies the number of 16 k chunks to preallocate when the file system is running low of bufstructs.
j2_nBufferPerPagerDevice	This tunable (512 by default) specifies the number of bufstructs that start on the paging device. JFS2 allocates more dynamically. It may be appropriate to change this value if j2_dynamicBufferPreallocation tuning has already been attempted and the number of external pager file system I/O requests blocked due to no fsbuf increases rapidly.
j2_maxPageReadAhead	This tunable (128 by default) specifies the upper limit for AIX JFS2 prefetching. It affects efficiently when doing large I/O.
j2_minPageReadAhead	This tunable (2 by default) determines the number of pages ahead when VMM initially detects a sequential pattern.
j2_nPagesPerWriteBehindCluster	Controls the gathering I/Os for sequential write behind. The default is 32.

Table 1: Most frequently used AIX file system tuning parameters

Release-behind mechanism for JFS and enhanced JFS

Release-behind mechanish is another suggested tuning for SAS. This feature allows the file system to release the file pages from file system buffer cache as soon as an application has read or written the file pages. This feature helps the performance when an application performs a great deal of sequential reads or writes and most often, once accessed, these file pages will not be accessed again in the near future.

If release-behind is not used, it might cause threads to wait on page replacement to supply enough free frames to handle file reads or writes. In the worst case, the page replacement activity might cause paging. When writing a large file without using release-behind, writes will go very fast whenever there are available pages on the free list. When the number of pages drops to minfree, VMM uses the least recently used (LRU) algorithm to find candidate pages for eviction.

A trade-off of using the release-behind mechanism is that application can experience an increase in processor utilization for the same read or write throughput rate (as compared to not using release-behind). This is because of the work required to free pages, which is

normally handled at a later time by the LRU daemon. Also note that all file page accesses result in disk I/O as file data is not cached by VMM. However, applications (especially long-running applications) with the release-behind mechanism applied still perform more optimally and with more stability.

This feature can be configured on a file system basis. When using the *mount* command, enable release-behind by specifying one of the following three flags:

- Release-behind sequential read flag (-rbr)
- Release-behind sequential write flag (-rbw)
- Release-behind sequential read and write flag (-rbrw)

AIX LVM tuning

The LVM provides an abstract logical view of the underlying physical disk devices. Logical volumes are employed to contain paging spaces and dump areas, but most often they underlie file systems. LVM uses a construct called pbuf to control a pending disk I/O. A single pbuf is used for each I/O request. The application generating large amount of I/Os or striping and mirroring environment usually requires more pbufs to satisfy the system needs. Running out of pbufs can degrade the performance as the I/O initiating process is suspended until pbufs are available again.

- The parameter *pv_pbuf_count*, used to control the number of pbufs available to the LVM device driver, can be set for each logical volume using the **lvmo** command.

Disk storage tuning

From a high level, the AIX I/O stack contains several layers that an I/O must traverse. At each layer AIX keeps track of the I/O. Some of the layers have specific queues that are useful to consider tuning. The I/O stack layers are:

- Application
- File system (optional)
- LVM (optional)
- Subsystem Device Driver (SDD) or SDDPCM (if used)
- hdisk device driver
- Adapter device driver
- Interconnect to the disk
- Disk subsystem
- Disk

In this section the focus is on tuning the middle layers consisting of SDD, hdisk and adapter device drivers. The goal is to improve simultaneous I/O capability and realize efficient queue handling. Refer to ***Table 2*** for some of the parameters that can affect disk and adapter performance. In general, SAS applications will benefit from careful consideration and tuning of these parameters.

Both the disk and adapter have maximum transfer parameters that can be adjusted to handle larger I/O, reduce I/O splitting, and coalesce I/O as it moves up and down the stack. In addition, both have I/O queues that can be adjusted to accept additional I/Os.

If SDD is used (IBM System Storage DS6000 or DS8000 systems) the data path optimizer (dpo) device I/O queue should be evaluated. SDD provides a virtual path to the storage

subsystem LUN or logical disk and provides several hdisk devices through the physical paths (such as FC adapters). So, with SDD one can issue queue_depth x number of paths to LUN.

However, when the dpo device queue is enabled (default is yes) any excess I/Os that can not be serviced in the disk queues go into the single wait queue of the dpo device. The benefit of this is the dpo device is designed to provide fault-tolerant error handling. This might be desirable for high-availability applications but for other applications there are advantages to disabling the dpo device queue and utilizing multiple hdisk wait queues for each SDD vpath device. Note that this is not an exhaustive discussion and does not detail any possible AIX limitations for total number of I/Os. Also the queue parameters need to be carefully evaluated before implementing any changes. For tuning guides specific to a particular IBM storage system, such as the IBM System Storage DS8000, DS6000, or DS4000 systems, refer to the “Resources” section.

Parameters	Description
max_xfer_size	• FC adapter maximum I/O that will be issued.
max_transfer	• Disk maximum I/O that will be issued.
queue_depth	• Disk maximum number of simultaneous I/Os. • The default is 20 but can be set as high as 256 for IBM Enterprise Storage Server® (ESS), DS8000, and DS6000 systems.
num_cmd_elems	• FC adapter maximum number of simultaneous I/Os. The default is 200 per adapter but can be set up to 2048.
qdepth_enable	• Subsystem Device Driver (SDD) data path optimizer (dpo) device queueing parameter. • The default is yes. A setting of **no** disables SDD queueing. • Use this with ESS, DS6000, and DS8000 systems.
lg_term_dma	• Long term DMA - Memory area the FC adapter uses to store I/O commands and data.
LTG	• AIX volume group Logical Track Group parameter. • LTG specifies the largest I/O the LVM will issue to the device driver. • In AIX 5.3, the LTG dynamically matches the disk maximum transfer parameter.

Table 2: Disk and adapter I/O tuning parameters

Note: It is important to understand the I/O characteristics of the application in order to properly tune within the I/O stack layers. If the SAS application has predominantly large I/Os then the application performance can benefit from adjusting maximum transfer sizes, long term DMA, and the LTG. The recommended starting values for a large I/O highly sequential workload are *lg_term_dma=0x800000*, and *max_xfer_size=0x200000*.

Queue information can be monitored in AIX 5.3 and higher versions with the *iostat –D* command. For AIX 5.1 and 5.2 SAR can be used. It is recommended that *qdepth_enable=no* to use the hdisk wait queue rather than the dpo device wait queue.

It is recommended to increase the *num_cmd_elems* for the FC adapter from the default (initially start at 400). Some of these parameters require a system reboot to take effect. For additional guidelines, see the tuning guide links found in "Resources" section.

Use the following commands to display and modify disk and adapter parameters and settings.

Disk – max_transfer, queue_depth

- **lquerypv –M hdisk#** displays maximum I/O size a disk supports.
- **lsattr –El hdisk#** displays current disk values.
- **lsattr –R -l hdisk# -a max_transfer hdisk#** displays allowable values.
- **chdev –l hdisk# -a max_transfer=value –P** modifies current disk values

Note: The device should be in an offline state or disabled state before changing any parameters. Then **cfgmgr** will need to be issued.

Adapter – max_xfer_size, lg_term_DMA, num_cmd_elems

- **lsattr –El fcs#** displays current value.
- **chdev –l fcs# -a max_xfer_size=value -P** modifies current value.

Note: The device should be in an offline state or disabled state before changing any parameters. Then **cfgmgr** will need to be issued.

SDD/DPO – qdepth_enable

- **lsattr –El dpo** displays current value.
- Use **datapath** command to change if at SDD 1.6 or higher. Otherwise the **chdev** command can be used. Example: **datapath set qdepth disable**

Application tuning

You can often improve system performance by tuning your applications to make the best use of the system resources. You can make use of the following tuning tips.

- A new SAS 9 feature will allow you to direct the utility files created by multithreaded SAS procedures to a different location from the SAS WORK area. The SAS parameter is called –UTILLOC. With this parameter and the -WORK parameter, you can direct temporary files created by a SAS session to different file systems (I/O paths). Both, the –UTILLOC and the –WORK parameters, must be set at the invocation of the SAS session. This feature can greatly help the performance of SAS applications/programs that are I/O bound on the directory where SAS WORK is currently pointing.

- Each SAS session consists of multiple processes. If the SAS session does not have an attached X-display then there will be two processes **sas** and data path optimizer. If an X-display is attached, a third process **motifxsassm** is started.

  The SAS Memory Utilization option **MEMSIZE** specifies the total amount of memory available to each **sas** process. As SAS dynamically allocates and frees memory, the amount of memory used by a sas process is usually less than MEMSIZE.

  svmon shows for a V9.1 sp 2 batch sas process after initialization 11,683 4 k pages of nonsystem memory and 10,592 of the 4 k pages are shared. The **elssrv** process after initialization uses 11,093 4 k pages of which 10,592 are shared with the sas process. Therefore, the first SAS batch session at initialization uses approximately 49 MB of memory and each subsequent SAS session at initialization uses 7 MB.

For GUI-based sessions, the sas process after initialization uses 12,403 4 k pages of nonsystem memory and 10,592 of the 4 k pages are shared. The elssrv process after initialization uses 11,093 4 k pages of which 10,592 are shared with the sas process. The motifxsassm process uses 11,371 of which 10,592 are shared with the sas and elssrv processes. Therefore, the first SAS batch session at initialization uses approximately 55 MB of memory and each subsequent SAS session at initialization uses 13 MB.

Performance monitoring methodology

This section outlines a methodology for performance monitoring.

Goals

Here is an example of some common goals for performance monitoring.

- Help to identify the performance bottleneck and tune the performance.
- Help to build an ISV workload characteristics profile in IBM System p environment. It can be served as a baseline profile for future ISV and IBM collaborations. This data can potentially be used to create an ISV sizing estimator.

Monitoring scope

Monitoring scope includes processor and memory utilization, disk I/O, network I/O, chip and memory subsystem, application, logical partition/logical processor, and system environment/configuration. It monitors each of those areas from overall to detailed perspectives.

When to monitor

You can monitor:

- When the run reflects a representative time slice of application workload
- When facing a performance bottleneck
- Anytime you prefer to view the details of insight

Suggested performance tools for monitoring

Tool	Description
vmstat	▪ Monitor overall system performance in the areas, such as processor, virtual memory manager (VMM) activity, and I/O.
tprof	▪ A global and micro-profiling tool. It is used to check the hot spots.
curt	▪ Produce a detailed processor utilization for process/thread/pthread activity.
trace	▪ The trace can be post-processed to check the events, such as krlock contention, workload access pattern, inode contention, and so on.
svmon	▪ Monitor the detailed memory consumption on real and virtual memory.
ps	▪ Monitor process/thread status and memory consumption as well.
iostat	▪ Monitor overall I/O stats including disks loads or adapters, and system throughput.
sar	▪ Report the per-processor, disk, run queue statistics.
filemon	▪ A magnifying glass tool. Used for detailed file I/O activity (for example, hot lv, pv).
netstat	▪ Report network and adapter statistics.

netpmon	▪ Report detailed statistics on network I/O and network-related processor usage, data rates, and response time.
hpmcount	▪ A tool that programs the on-chip and memory subsystem's performance monitor facilities to count a set of events.
lparstat	▪ Report logical partition-related information. For example, partition configuration, hypervisor call, and processor utilization statistics.
mpstat	▪ Report logical processor information in logical partition. For example, simultaneous multithreading (SMT) utilization, detailed interrupts, detailed memory affinity, and migration statistics for AIX threads, and dispatching statistics for logical processors.
topas	▪ Report the local system's statistics, including: processor, network, I/O, processes, and workload management classes utilization.
nmon	▪ A commonly used freeware tool for capturing AIX performance data. ▪ Use this tool together with nmon analyzer which loads the nmon output file and automatically creates dozens of graphs reflecting key system performance characteristics. ▪ Refer to the *SAS Performance Monitoring – A Deeper Discussion* paper at http://www2.sas.com/proceedings/forum2008/387-2008.pdf This is an SAS Global Forum 2008 paper for procedure of collecting nmon trace.

Table 3: Suggested performance tools for monitoring

Monitoring example – Processor utilization monitoring

Here are some examples:

- Suggested monitoring tools: *vmstat, iostat, ps, sar, tprof*
- Overall processor utilization monitor
  A system is probably processor-bound if the system processor utilization (usr+sys) is always greater than 80 percent. *iostat*, *vmstat*, and *sar* can help in determining whether a system is processor bound or not. Here, *vmstat* is used to demonstrate the processor monitoring methodology.

  The four processor utilization group, such as us, sys, wa, idle in *vmstat* report indicates processor time spent in user mode, system mode, idle or I/O wait. The first group **kernel thread** of two columns **r** and **b** represents statistics about thread queues. It is suggested to check these two columns first.

  % us: Percentage of processor time spent in the application code (that is SAS). In order to maximize the throughput, ideally this value should be as high as possible.
  % sys: Percentage of processor time spent in the system calls and kernel code. Ideally system time should be as low as possible. High percentage in system time needs to be investigated.
  % wa: Percentage of processor time spent waiting for an I/O (disk read/write, network and so on) to be completed. Ideally, this value should be zero. If not, it means there is some opportunity to improve system throughput by either tuning disk or network or memory configuration.
  "r": Average number of runable kernel threads during the sampling interval. The run queue is used to display the number of active tasks that are currently waiting for processor resources. The higher the value in **r**, the more processor work there is to do, which is an indication of processor bottleneck.

"b": Average number of kernel threads in the wait queue during the sampling interval. If threads are consistently being forced to wait, processor performance will get degraded.

- Detailed processor utilization analysis
  If you decide that the system is processor bound, *tprof* can be used to check which process or program is dominating the processor usage. *ps* can also be used but profiler is a better method. After identifying the high-utilization process, you can decide if this behavior is normal and then tune as needed. Conducting further analysis before just adding more processing power to the server is always recommended.

  - **Step 1- Profiling entire system**: In order to have a better understanding of SAS workload characteristics on IBM Power processor-based server, establishing a baseline profile is the first step. You need to get familiar with the workload pattern by checking if there are outstanding routines based on the profiling data. An outstanding routine means that the processor spends more time in this routine compared to others (for example, 25 percent compared to 3 percent). Further evaluation is required for the outstanding routine.

  - **Step 2- Micro-profiling SAS user application**: Micro-profiling can focus on where processor spent the most time in the application. For instance, if vmstat reports that processor utilization was spent in user mode, micro-profiling of the application is reasonable.

Understanding processor utilization on IBM Power Systems™ - AIX

Traditionally, you have been accustomed to use processor utilization as the primary metric to understand the performance of a system running a workload, to do capacity planning, and to do charge back. Processor technology has undergone tremendous changes in the past decade, which has called for a change in the way that processor utilization is computed and correctly interpreted. To better understand how processor utilization is computed in AIX and what changes it has undergone in the past decade in synchronization with the IBM POWER processor-based technology changes, refer to the IBM article, Understanding Processor Utilization on POWER Systems – AIX at: **ibm.com**/developerworks/wikis/display/WikiPtype/Understanding+Processor+Utilization+on+POWER+Systems+-+AIX

GPFS tuning for distributed SAS workloads

GPFS tuning is specific to the workload type. Here are some GPFS tuning recommendations in a distributed SAS 9 environment. These are in addition to the recommendations provided in the "Disk storage tuning" section.

Parameter	Description
File system blocksize -B *blocksize*	The GPFS file system blocksize should be set based on the specific SAS solution I/O requirements. Example: For a mixed workload of sequential and random I/O in a specific SAS Markdown Optimization test, setting the GPFS blocksize to 256 KB rather than 2 MB improved the overall file system performance by 22 percent. **Note**: Real-world performance might vary and is dependant on specific test environments.
pagepool	Set pagepool to 8 GB if there is available memory and the workload includes random file access. More cache will help the random portion of SAS workloads. Currently 8 GB is the maximum size allowed for pagepool (GPFS 3.1). The next version of GPFS will support a larger value of pagepool.
maxFilesToCache	The default value for maxFilesToCache is 1000. Increasing this value to 20,000 or greater, in general, improves the performance for user interactive tasks. Example: The SAS MO solution tested uses thousands of files and benefited from increasing this value.
Disk bandwidth	In a distributed environment, the amount of available memory might be smaller on each system. In this case, to further improve the performance of the solution, add additional disk bandwidth to GPFS. This means adding additional physical disks to the GPFS storage pools.

Table 4: GPFS parameters

IBM System Storage DS4000 options

System Storage DS4000 system options:

- Enable read and write cache as well as write cache with mirroring for each DS4000 system logical drive
- Disable the write cache without batteries option
- Enable the cache read ahead multiplier with at least a value of 1
- Enable Microsoft® Windows® RDAC driver write caching and advanced performance for each drive. Refer to Figure 2 and Figure 3.

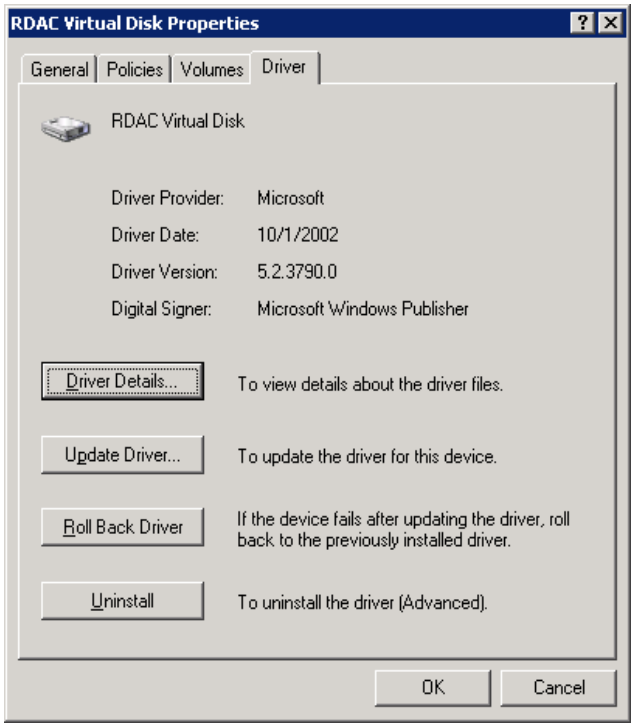

Figure 2: General RDAC driver properties

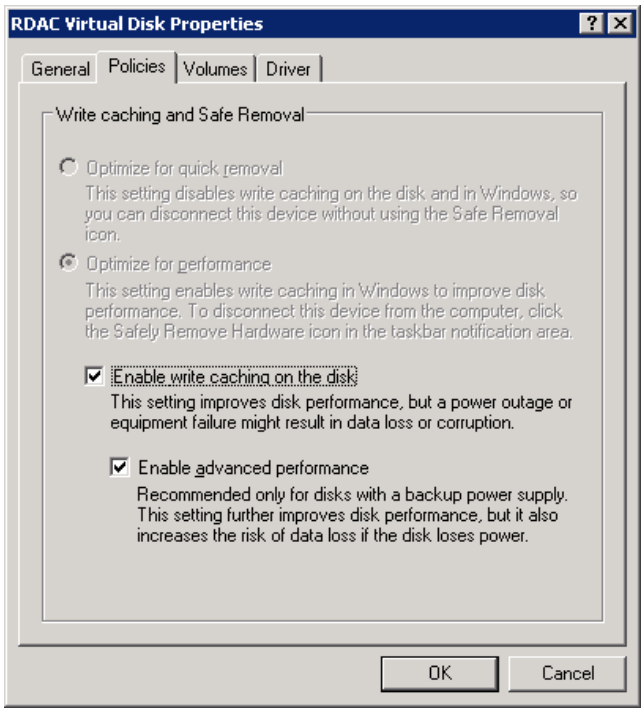

Figure 3: RDAC Windows write caching and advanced performance

Storage monitoring tools and recommendations

This section provides a summary of the IBM Tivoli storage monitoring tools.

IBM Tivoli Storage Productivity Center

Storage monitoring of individual storage subsystems can be dependent upon the storage platform and the operating system as shown in the above O/S storage recommendations sections. Some storage platforms provide more robust monitoring tools while the O/S specific monitoring tools vary widely depending on the server platform. The IBM Tivoli® Storage Productivity Center is designed to provide the ability to monitor disparate storage platforms. Tivoli Storage Productivity Center can be used for monitoring of IBM XIV Storage System, IBM System Storage DS8000 series, DS5000 series, as well as the IBM System Storage SAN Volume Controller among other storage platforms.

More about Tivoli Storage Productivity Center: As the growth of data storage continues to explode, there is an increasing need for businesses to find ways to control the cost of storage. Managing storage infrastructure has grown in complexity as customers acquire new storage infrastructure that is heterogeneous. And businesses need to identify, evaluate, control and predict the growth of data through its lifecycle in order to address storage service levels in accordance with IT Information Library (ITIL) and data retention needs.
You can visit **ibm.com**/systems/storage/software/center/index.html to find more information on IBM Tivoli Storage Productivity Center.

The IBM Tivoli Storage Productivity Center is an open storage infrastructure management solution designed to help:

- Reduce the effort of managing complex, heterogeneous storage infrastructures
- Improve storage capacity utilization
- Improve administrative efficiency

Tivoli Storage Productivity Center is designed to provide reporting capabilities, identifying data usage and its location, and provisioning. It is also designed to provide a central point of control to move the data based on business needs to more appropriate online or offline storage, and centralize the management of storage infrastructure capacity, performance, and availability. These key capabilities support information on demand, enabling the easy management of data through its lifecycle while helping to address ITIL and data retention needs and reduce the cost of storage.

The IBM Tivoli Storage Productivity Center family of products includes:

- IBM Tivoli Storage Productivity Center for Basic Edition:
  ibm.com/systems/storage/software/center/basic/
- IBM Tivoli Storage Productivity Center for Data:
  ibm.com/systems/storage/software/center/data/index.html
- IBM Tivoli Storage Productivity Center for Disk:
  ibm.com/systems/storage/software/center/disk/
- IBM Tivoli Storage Productivity Center for Disk Midrange Edition:
  ibm.com/systems/storage/software/center/disk/me/index.html

- IBM Tivoli Storage Productivity Center for Replication:
  ibm.com/systems/storage/software/center/replication/index.html
- IBM Tivoli Storage Productivity Center Standard Edition:
  ibm.com/systems/storage/software/center/standard/

The new IBM Tivoli Storage Productivity Center is designed to provide a central console to manage storage infrastructure to:

- Plan storage and database growth
- Be able to view end-to-end storage infrastructure, pinpoint performance and availability issues, and assess business impact
- Configure heterogeneous storage infrastructure
- Report on business storage capacity, backup and archive operations, storage infrastructure performance, and availability
- Do problem determination within SANs and storage systems

The IBM Tivoli Storage Productivity helps you:

- Monitor and track the performance of SAN attached SMI-S compliant storage devices
- Manage the capacity utilization and availability of file systems, databases, and IBM Tivoli Storage 3584 tape libraries
- Monitor, manage, and control (zone) SAN fabric components
- Manage advanced storage replication services (Peer-to-Peer Remote Copy and IBM Tivoli Storage FlashCopy® manager)
- Automate capacity provisioning to help improve application availability
- Centralize the management of your storage infrastructure from a single interface using role-based administration and single sign-on
- Provide a single management application with modular integrated components that are easy to install, configure, and operate
- Manage performance and connectivity from the host file system to the physical disk, including in-depth performance monitoring and analysis on SAN fabric performance

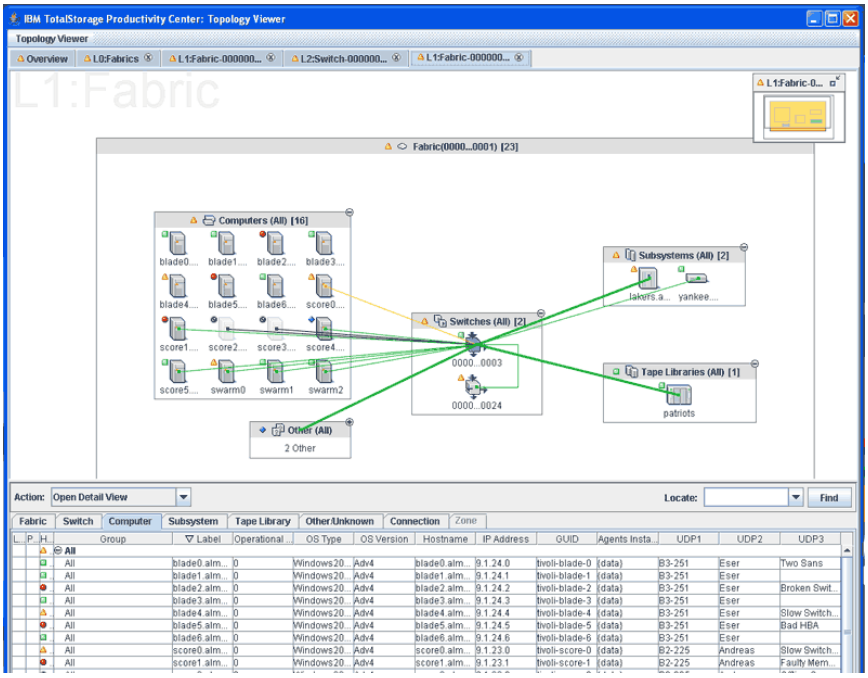

Figure 4: Tivoli Storage Productivity Center

IBM XIV Storage System GUI

The IBM XIV Storage System software includes features that allow you to monitor the system. You can review or request the current system status and performance statistics at any time. The IBM Tivoli Storage Productivity Center can also be configured to communicate and manage the IBM XIV Storage System. The monitoring functions available from the IBM XIV Storage System GUI allow the user to easily display and review the overall system status, events, and various statistics. Figure 5 shows the look and feel of the XIV system GUI and a sample report of the type of data that can be displayed.

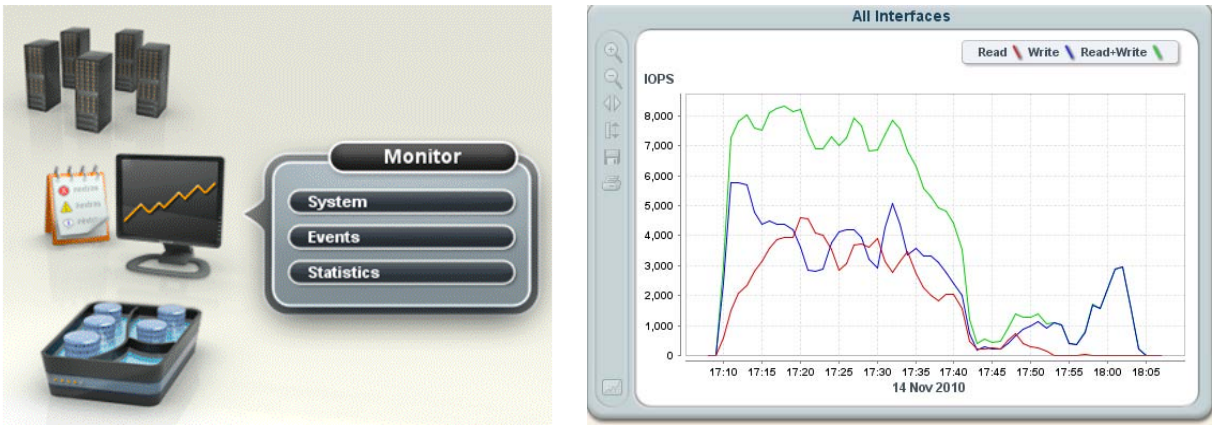

Figure 5: IBM XIV Storage System GUI and a sample report

Conclusions

SAS 9 is a robust and rich solution with specific I/O patterns dependent upon specific application and environment variables as well as the associated server and storage hardware. The desired I/O throughput or IOPS or both can be achieved by carefully factoring in the application I/O variables and requirements into each server or storage hardware design, selecting the appropriate high-performing IBM server and storage systems for the performance requirements, and adapting the provided I/O thumb rules thoughtfully into the specific application environment through OS, application, and hardware tuning.

Resources

- SAS 9 on AIX5L and AIX6 Tuning Guide (I/O specific recommendations and additional SAS tuning tips)
  ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101529

- IBM XIV Storage System: Architecture, Implementation, and Usage
  ibm.com/redbooks/abstracts/sg247659.html

- General Parallel File System - Document Library:
  http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp?topic=/com.ibm.cluster.gpfs.doc/gpfsbooks.html

- General Parallel File System FAQs (GPFS FAQs):
  http://publib.boulder.ibm.com/infocenter/clresctr/topic/com.ibm.cluster.gpfs.doc/gpfs_faqs/gpfs_faqs.html

- IBM Tivoli Storage Productivity Center V3 – Performance Management Best Practices
  ibm.com/support/docview.wss?uid=ssg1S7001493&rs=1133

- IBM System Storage SAN Volume Controller:
  ibm.com/systems/storage/software/virtualization/svc/

Redbooks

- IBM System Storage Solutions Handbook, SG24-5250
  ibm.com/redbooks/abstracts/sg245250.html?Open

- IBM Midrange System Storage Implementation and Best Practices Guide, SG24-6363
  ibm.com/redbooks/abstracts/sg246363.html

- IBM System Storage DS6000 Series: Performance Monitoring and Tuning, SG24-7145
  ibm.com/redbooks/abstracts/sg247145.html

- IBM System Storage DS8000 Series: Performance Monitoring and Tuning, SG24-7146
  ibm.com/redbooks/abstracts/sg247145.html

- Implementing an IBM b-type SAN with 8 Gbps Directors and Switches, SG24-6116
  ibm.com/redbooks/redpieces/abstracts/sg246116.html

- Implementing the IBM System Storage SAN Volume Controller V5.1, SG24-6423
  ibm.com/redbooks/abstracts/sg246423.html

- IBM Tivoli Storage Productivity Center V3.1: The Next Generation, SG24-7194
  ibm.com/redbooks/abstracts/sg247194.html?Open

- Deployment Guide Series: Tivoli Storage Productivity Center for Data, SG24-7140
  ibm.com/redbooks/redpieces/abstracts/sg247140.html?Open

- Using IBM Tivoli Storage Productivity Center for Disk to Monitor the SVC, REDP-3961
  ibm.com/redbooks/abstracts/redp3961.html?Open

Support

- IBM Tivoli Storage Productivity Center, IBM System Storage SAN Volume Controller, and other IBM Storage products:
  ibm.com/support/entry/portal/!ut/p/c5/04_SB8K8xLLM9MSSzPy8xBz9CP0os3hjAwMjA4Ng01A_jwBLA6OgkABTPx9fQwM3E6B8pFm8D1Da3dHdwNLCwtXFwNMxMCDQwCTQwCDMjIDucJB9-PWD5A1wAEcDfT-P_NxU_YLcCIMsE0dFAP-Sm9Y!/dl3/d3/L0lJSkIna2lpbEEhIS9JRGpBQUF5QUJFU293Q0tpaGlZIS80Qm40dFdBeUlJaVFCOElRLzZfNDY1SVFJNDIwT0Q4MDAyUkNGS0NKSjIwMDAvN180NjVJUUk0MjA4TDQzMDJSUTVJTEpEMzAwNi9Ba0FIVDg5MTgwMDgyL21heGltaXplZA!!/#7_465IQI4208L4302RQ5ILJD3006?date=1298579369737

- IBM Tivoli Storage Productivity Center V3.3.2/4.1 Hints and Tips (Updated)
  ibm.com/support/docview.wss?rs=1133&context=SS8JB5&context=SSWQP2&dc=DA4A10&uid=swg27008254&loc=en_US&cs=utf-8&lang=en

For additional background information about IBM SAS I/O performance considerations and setup concepts refer to the IBM SAS ICC Information Brief entitled *Installing SAS Software on an IBM eServer pSeries Running AIX 5L* found at
http://www.sas.com/partners/directory/ibm/papers.html

Additional SAS information can be obtained at http://www.sas.com

For additional white papers on IBM storage solutions, go to: **ibm**.com/storage/

Trademarks and special notices

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

SET and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

Other company, product, or service names may be trademarks or service marks of others.
Information is provided "AS IS" without warranty of any kind.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Information concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.
All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Contact your local IBM office or IBM authorized reseller for the full text of the specific Statement of Direction.
Some information addresses anticipated future capabilities. Such information is not intended as a definitive statement of a commitment to specific levels of performance, function or delivery schedules with respect to any future products. Such commitments are only made in IBM product announcements. The information is presented here to communicate IBM's current investment and development activities as a good faith effort to help with our customers' future planning.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Photographs shown are of engineering prototypes. Changes may be incorporated in production models.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.