

---

## Best Practices for Data Sharing in a Grid Distributed SAS® Environment

Updated July 2010



## Table of Contents

<b>1</b>	<b>Abstract.....</b>	<b>2</b>
1.1	Storage performance is critical .....	2
1.2	Required background knowledge .....	2
1.3	Shared file system experience with SAS .....	2
1.4	NAS Appliance (NFS or CIFS) .....	3
1.5	Clustered and shared file system.....	5
1.6	Hybrid NAS and SAN systems - iSCSI.....	7
1.7	SAS experience with various shared file systems .....	9
1.8	Comparing various storage architectures for SAS .....	10
<b>2</b>	<b>Implementation Guidelines.....</b>	<b>11</b>
2.1	Initial architecture design .....	12
2.2	Throughput requirements for SAS applications.....	12
2.3	Location of SASWORK and Utility Space .....	12
2.4	Keep it simple .....	13
2.5	Design testing.....	13
2.6	Test I/O throughput outside of SAS .....	13
2.7	Example migrations from stand-alone to grid.....	14
	2.7.1 SAS Data Integration example .....	14
	2.7.2 Evaluation process.....	14
	2.7.3 Design approach .....	15
2.8	Multi-user environment example.....	15
	2.8.1 Evaluation process.....	16
	2.8.2 Design approach .....	16
	2.8.3 Sample solution.....	16
<b>3</b>	<b>Conclusion.....</b>	<b>18</b>
3.1	Authors.....	18
	<b>Appendix A: Storage basics and definitions.....</b>	<b>19</b>

# 1 Abstract

Data volumes continue to grow at unprecedented rates. Enterprise solutions must be able to access and process this data efficiently, reliably and quickly in order to produce the results necessary to make solid business decisions. This is true for many SAS customers who look for ways to distribute both their data and or their processing to help them meet ever-shrinking timelines. The performance and success of a solution depends on timely and reliable data access no matter where it resides. Customers are looking to SAS to provide guidance and recommendations on configuration of distributed environments. This paper provides an introduction to basic storage terminology and concerns. The paper describes the best practices used during successful testing with SAS and clustered file systems. This paper can be used as a reference guide when configuring a distributed environment, performing, and scaling to meet the needs of your organization.

## 1.1 Storage performance is critical

Storage performance is the most critical component of implementing SAS in a distributed grid environment. When your storage subsystem is not designed properly your application does not perform well. The performance and architecture of the storage required to support SAS software is critical.

## 1.2 Required background knowledge

There is basic terminology, definitions, and concepts that must be understood when building a grid distributed SAS environment. If you already have a strong background in this area, please read on. If you don't, you can find an overview of critical terms and technologies in the appendix of this document.

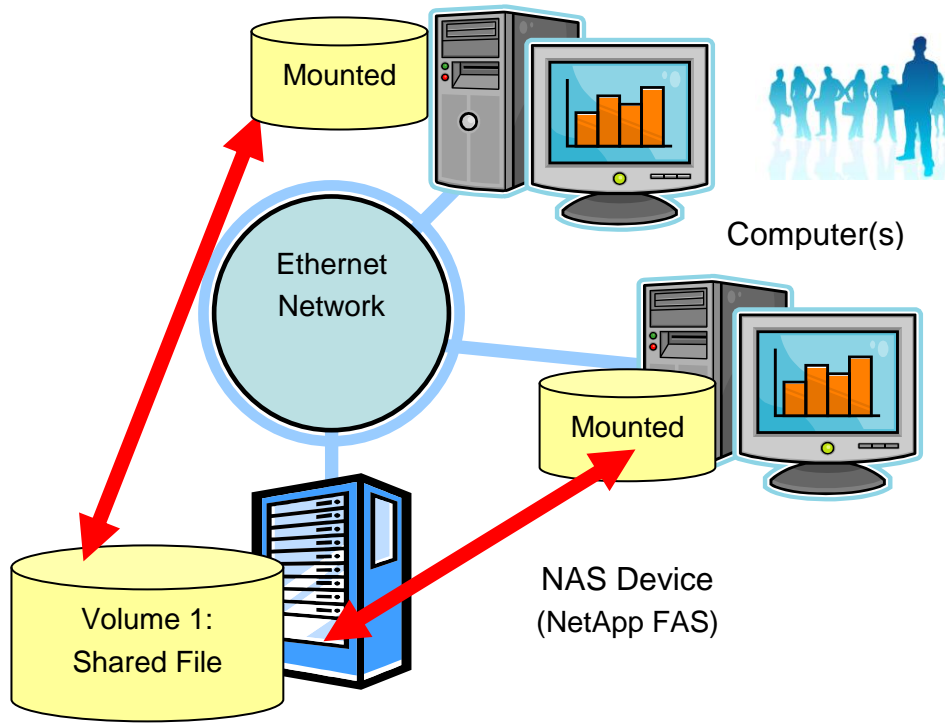
## 1.3 Shared file system experience with SAS

Defined below are three types of shared file systems that SAS has tested in various environments. These types are Network Attached Storage, Clustered / Shared File Systems and Hybrid File Systems.

## 1.4 NAS Appliance (NFS or CIFS)

When you purchase the Network Appliance FAS Storage device, commonly referred to as a NetApp filer, you can leverage standard Ethernet networks to share storage among storage clients using network sharing protocols like NFS or CIFS. There are slight differences between these protocols but typically the technology is the same. The advantage to this storage is that it is usually inexpensive and extremely easy to deploy. It is typically not recommended to use this type of storage for distributed SAS environments that require a large amount of I/O performance. However, as networking throughput and technology improves, the cost and ease of deploying this type of storage makes it a very attractive option.

<b>Advantages</b>	<b>Challenges</b>	<b>Best practices</b>
Easy to deploy and administer.	Less scalability than SAN based CFS.	Use a dedicated Ethernet interface(s) on storage clients for storage traffic.
Low cost shared file system option.	Poorer performance due to packetized network traffic (but newer 10 GB Ethernets could help).	Use more than one Ethernet per storage client for availability.
Can be configured for high availability.	Does not support load balance across multiple Ethernet interfaces easily (extra software and hardware or manual load balancing required)	Use high speed networks like 10 GB Ethernet when possible.
		Ensure NFS server can meet server throughput requirements of storage clients (NFS servers typically don't provide the throughput capability of standard storage devices).



**Figure 1: NAS architecture diagram**

## 1.5 Clustered and shared file system

The most commonly implemented storage architecture for SAS software in a distributed environment is a Clustered or Shared File System (CFS). In this architecture, standard storage devices available from various hardware vendors are shared via SAN architecture (for example: connected together with a fiber channel switch). CFS is software component that is installed on all of the storage clients. Through various processes running on the storage clients, it enables the sharing of metadata and controls simultaneous access. There are many types of CFS software available from both third-party and hardware vendors.

<b>Advantages</b>	<b>Challenges</b>	<b>Best practices</b>
Typically the best performing shared file system architecture	Not all products support multiple operating systems	Understand the metadata requirement for the CFS.
Typically the best performing shared file system architecture	More expensive than NAS	Dedicated server and storage might be required to store metadata. Metadata server response time can be critical to overall CFS performance.
	Harder to administer than NAS	Monitor throughput performance from storage client all the way to the disk volumes in the storage device.
	Requires dedicated storage network (typically fiber channel).	

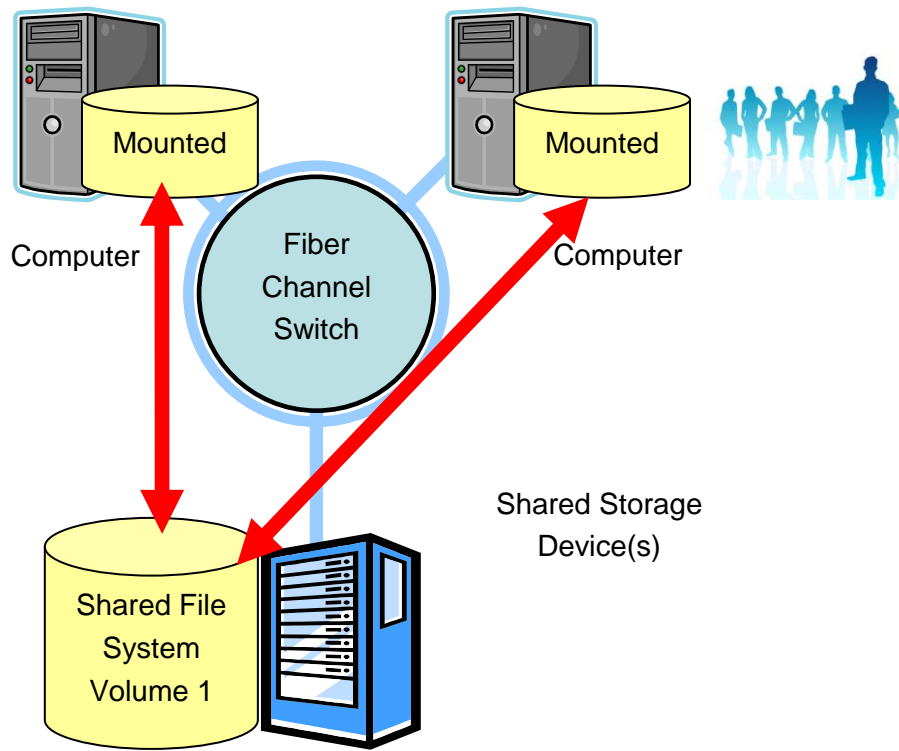


Figure 2: Clustered file system architecture diagram

## 1.6 Hybrid NAS and SAN systems - iSCSI

There are several iSCSI based file systems on the market. EMC's Multi-Path File System (MPFS) is a clustered and shared file system that can run over a pure fiber SAN or be used over iSCSI (MPFSi). SAS has done extensive testing with the MPFSi product depicted in figure 3. Storage clients are connected to a standard Ethernet and specialized equipment converts the network traffic (IP) over to storage protocols. These protocols are sent down to the storage devices by fiber channel. MPFS is based on EMC's previous Highroad product. Many different vendors offer iSCSI storage products.

Advantages	Challenges	Best practices
Faster throughput than pure IP based storage due to use of storage protocols	More complex to administer and monitor	Use a dedicated Ethernet interface(s) on storage clients
Strong failover and load balancing capability with multiple Ethernet interfaces	More expensive than NAS solution	Use more than one Ethernet per storage client for availability and performance
		Monitor throughput capability at all layers of the architecture
		Seek design help from the vendor to ensure throughput requirements can be met



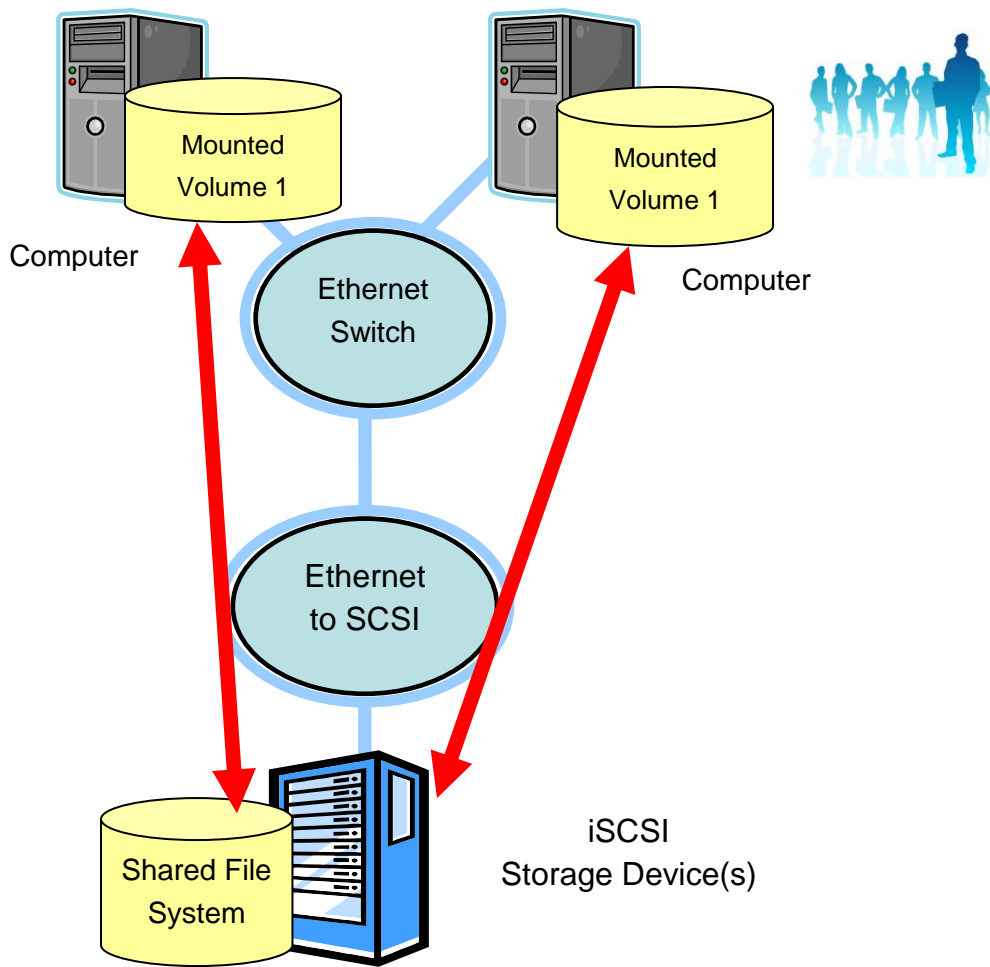


Figure 3: Hybrid NAS and SAN architecture diagram

## 1.7 SAS experience with various shared file systems

Each of these falls into one of three categories: Hybrid, NAS, or CFS. Following this table is an overview of each category including some advantages, challenges, and best practices for implementing them with SAS in a distributed or grid environment.

<b>Operating system used in tests at SAS*</b>	<b>File Sharing Technology</b>	<b>Category</b>
Red Hat Linux (RHEL 4)	EMC Celerra Multi-Path File System on iSCSI (MPFSi)	Hybrid
Red Hat Linux (RHEL 4)	Network Appliance (NFS)	NAS
Sun Solaris 10	Sun StorageTek QFS	CFS
Red Hat Linux (RHEL 4)	Global File System (GFS)	CFS
Microsoft Windows	HP Polyserve Matrix	CFS
Red Hat Linux (RHEL 4)	HP Polyserve Matrix	CFS
IBM AIX	IBM Global Parallel File System (GPFS)	CFS
HP-UX	Veritas Clustered File System (CFS)	CFS
Microsoft Windows	IBM Global Parallel File System (GPFS)	CFS
Microsoft Windows	Microsoft CIFS	NAS

\* Testing level and experience of each file sharing technology varied.

## 1.8 Comparing various storage architectures for SAS

Listed in the table below are the primary storage types along with some of the advantages and challenges of each. Highlighted in bold in the table are some of the most important points to consider when choosing a storage option.

Storage type	Advantages	Challenges
Direct Attached Storage (DAS)	<p>Predictable performance.</p> <p>Easy administration.</p> <p><b>Good for temporary storage in a distributed environment where data doesn't need to be shared.</b></p>	<p><b>File system and data not shareable</b></p> <p>Can be underutilized since it cannot be shared among multiple servers.</p>
Network Attached Storage (NAS) Data shareable	<p><b>Easy to access since Ethernet is a standard across operating systems.</b></p> <p>Typically has shared file system capability built in (NFS or CIFS).</p> <p>Easy administration.</p> <p>Typically lower cost to implement (switches, built in Ethernet cards, cabling.).</p> <p>10 gigabit Ethernet now available to help with performance.</p>	<p><b>Predictable performance and throughput rates harder to achieve due to other network activity.</b></p> <p>Scalability is not linear as you increase infrastructure.</p> <p>Data is packaged into packets and re-assembled at target, which can cause slower throughput rates.</p>
Storage Area Network (SAN) + Clustered File System (CFS) Data sharable	<p>Allows point-to-point communication between storage and host systems.</p> <p>Transfer of data between storage devices does not require host CPU or memory cycles.</p> <p><b>Higher throughput rates than NAS, similar to DAS performance.</b></p> <p>Higher availability can be achieved versus NAS due to architecture.</p>	<p><b>Typically more complex to administer</b></p> <p>Requires specialized network infrastructure.</p> <p>More expensive due to extra infrastructure and extra cost of clustered file system (CFS).</p> <p>Often involves multiple manufacturers to create optimal infrastructure.</p>
Hybrid (iSCSI) Data sharable	<p>Uses existing Ethernet for clients.</p> <p>File system sharing part of the solution.</p> <p>Easy administration on the client side.</p> <p><b>Faster than NAS solution.</b></p>	<p>Can be as expensive as SAN and CFS solution.</p> <p>More complex to administer on the server and network side, more physical components to monitor and administer</p> <p><b>Not as fast as a SAN based systems.</b></p>

## 2 Implementation Guidelines

The following lists of items (pre-implementation, test, and design considerations) have been assembled from SAS experience with distributed environments and storage. These guidelines can help you plan your migration of SAS applications to a distributed environment.

### Is my application or environment a good candidate for distribution?

Use the guidelines defined in the first section of this paper on when to or when not to move your application to a distributed environment.

### Draw a data flow diagram

Draw a diagram of the environment that helps you understand how data moves through your application. Include the amounts of data and throughput requirements. Figure 4 outlines network, storage, and server requirements. Below is an example of a data flow diagram for a grid enabled ETL process. Each phase of the process is shown with details of the volumes of data that is accessed during the phase. Understanding throughput requirements during execution is critical to the storage architecture design. Use the process attached in the appendix of this paper to help analyze your applications data flow requirements.

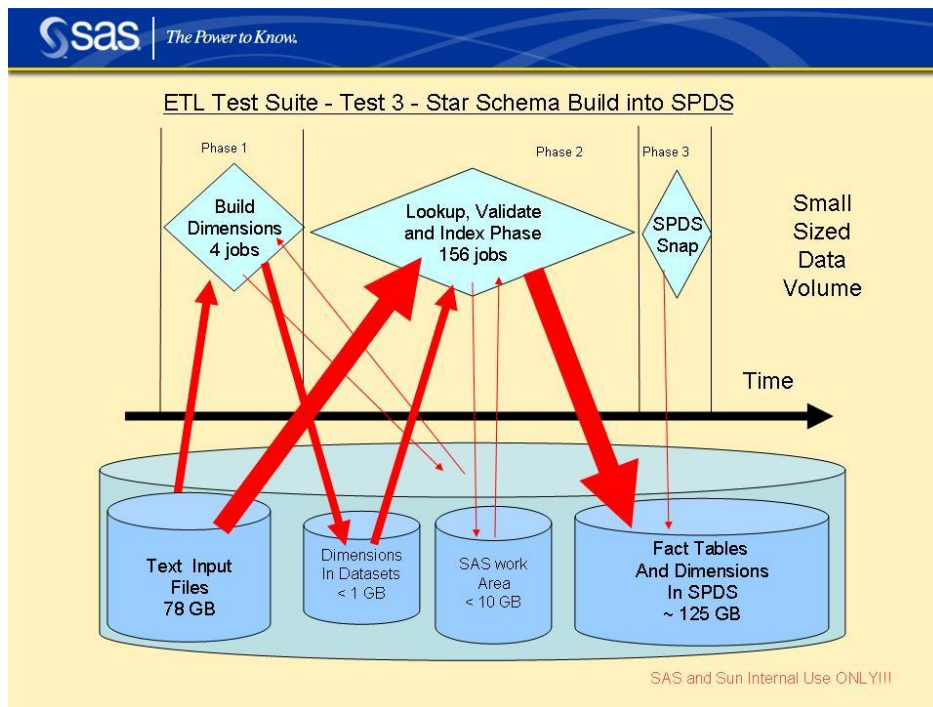


Figure 4: Sample data flow diagram

## 2.1 Initial architecture design

Draw a detailed diagram of the distributed architecture. This helps identify required hardware, network, and software that you need. Be sure to label all your storage and network connections (type, sustained throughput capability, protocol).

## 2.2 Throughput requirements for SAS applications

It is always best to use measured I/O throughput statistics from your application when building out your data architecture. However, when there are no real numbers the following can be used as general estimates for calculating throughput requirements for SAS:

<b>Ad hoc SAS Power Users (heavy)</b>	50-75 MB per second simultaneous process
<b>SAS Enterprise Guide / Microsoft Add-In (medium)</b>	15 MB per second simultaneous user
<b>Business intelligence users (light)</b>	1 MB per second simultaneous user

**Note:** Many processes or applications running in a distributed fashion are classified in the first category as heavy.

## 2.3 Location of SASWORK and Utility Space

A common question asked, is where to locate SAS temporary storage in a distributed environment. Should the temporary storage be local to the grid node, or on the network shared storage? This chart highlights the advantages and challenges of each choice. Which one you choose is dependant on your application's requirements.

Where to put SAS temporary space?	Advantages	Challenges
Local or DAS	<p>Predictable Performance</p> <p>Might be faster performance, especially if you have a heavily utilized shared storage environment.</p> <p>Less SAN / CFS traffic, therefore better overall performance.</p> <p>File system failure for SAS temporary space takes one node down (however with RAID, this is less of an issue).</p>	<p>Not shareable disk space, so it might sit idle.</p> <p>Takes up expensive real estate, maybe you want to use smaller blade servers.</p> <p>What do you do if you want to expand temporary storage and don't have extra disk slots.</p>
On the shared storage	<p>More disk spindles available for temporary storage, so performance could be better.</p> <p>No wasted disk space since all storage is shareable.</p> <p>One place to administer storage.</p>	<p>Shared resources increase likelihood of unpredictable performance</p> <p>Shared storage is probably more expensive than local attached</p> <p>More storage area bandwidth is required due to more traffic.</p>

A typical SAS environment has separate file systems for the following items:

- SAS Data**      Could be multiple file systems of varying sizes.
- SASWORK**     Temporary storage for SAS heavily accessed in data intensive applications.
- SAS Utility**    Heavily used for things like SAS sort, by default included with SASWORK.
- SAS Binaries**   Location of SAS and third-party application smaller read only file system.

How many you actually deploy can vary. This is based on the ability of the file system to handle certain sized disk volumes and how easily as well as the ease of growing their size in the future. The number and size of file systems is much less important than the performance of these file systems for your application. Focus on performance first.

In the past the RAID level of choice for SAS was mostly RAID 0, which is great for performance but provides no redundancy. Today, most customers are implementing RAID 5 file systems for almost everything. Dedicated hardware on the storage devices to generate RAID redundancy information (parity) has improved in performance and increased RAID 5 performance. The performance is equal to mirroring and striping combined (RAID 1+0, RAID 10). The SAS temporary space should be implemented with redundancy; otherwise any disk failure would take down all SAS processes running across your system.

## 2.4 Keep it simple

Distributing an application across a grid requires a different type of thinking than that of a single server implementation. It is important not to over design and make your architecture so complex that it is difficult to monitor, expand, diagnose, tune, and monitor. Be sure to consider if the performance improvement is worth the additional cost for administration and usability of a distributed environment.

## 2.5 Design testing

It is a common mistake to minimize the data volume during testing of any solution or product because of a lack of available resources or time. This results in an unrealistic test as the hardware (both storage and servers) caches data. A test that “cheats” and uses cache for most of the data access does not truly test the entire system and sets false expectations. Be sure you use enough data in your tests to saturate hardware cache for storage and system memory.

## 2.6 Test I/O throughput outside of SAS

When you build out your test environment or the final distributed architecture, it is critical to test the I/O throughput capability before you install or test SAS. Listed in the reference section of this paper are some links to tools that can be used to test I/O throughput outside of SAS. Getting a baseline of

the hardware before executing your SAS application will help you isolate performance issues after you deploy your application. If your baseline is not what you expect the system to be capable of, this allows you to tune and fix the architecture before you implement SAS. Doing I/O tests early saves you time and frustration in the deployment phase.

## **2.7 Example migrations from stand-alone to grid**

Here are two possible examples of applications that are good candidates to consider moving to a distributed or grid environment (there are many others). The first example is a custom SAS<sup>®</sup> Data Integration or ETL application. The second is a large multi-user environment that grew beyond the ability that a single server can handle. These examples give you a high-level idea of the process of you when determining if and when it is beneficial to move to a distributed environment. determine whether. Financial considerations, although important, are not included in this technically focused discussion.

### **2.7.1 SAS Data Integration example**

**Problem:** Customer has multiple Terabytes of data that they need to process into a central Teradata warehouse on a nightly basis. The processing window is shrinking and the data volumes are growing. The data comes into the SAS server through hundreds of individual input streams that are stored in separate ASCII text files. The customer is not able to complete the ETL jobs in the required window. The current 8 CPU server is 100 percent busy. A single serial SAS program was written to handle the ETL process of the incoming files.

### **2.7.2 Evaluation process**

From first glance, this example appears to be a good candidate for grid. The technical questions to ask are the following:

- What is the I/O requirement per process?
- Are you sure CPU bound?
- What is the ETL processing window?
- How long does it take to process the largest input stream? Average runtime?
- How many input streams to you need to process?
- If applicable, what is the bandwidth of the connection to the Database and or other data resources?

### 2.7.3 Design approach

- Make sure the SAS program can be split into separate programs that can execute simultaneously across multiple grid nodes.
- Determine how many simultaneous SAS threads or processes are needed to complete the ETL task in the required window of time.
- When larger SAS processes and tasks take longer to run than the execution window allows, look at implementing parallelization inside the SAS process. Break the processing of larger files into a parallel process. In order to break apart your data, you need to look for processing dependencies. If there are any, you can not be able to parallelize within the program.
- Draw a data flow diagram for the problem: include throughput, runtime, and data volume requirements.
- Analyze your data flow diagram and begin to design your hardware architecture.
- Test your design in a lab or validate it by comparing to existing reference architectures.

#### Sample solution

This problem is an ideal candidate for a SAS grid implementation using CFS or hybrid storage architecture. Although this is a simplified example, the data volumes listed here are better suited for these two types of storage (higher throughput requirements). When the data volumes were lower and there was more emphasis on CPU utilization, then a NAS solution could be used.

An example solution for this would be multiple 4-way servers attached via a fiber channel switch to a large storage array(s). A CFS would be used to provide for file sharing across the grid nodes. SAS Grid Manager would be used in combination with SAS Data Integration Server to deploy and manage SAS across the grid.

## 2.8 Multi-user environment example

Problem: Customer has many concurrent SAS users sharing a single very large multiple CPU server (SMP system). The system response is unpredictable and administrators notice lengthy periods where CPU is at or close to 100 percent utilization (sometimes causing a server crash). There is no room in the current hardware to add CPU or I/O resources and the user base and demands on the server continue to increase. The customer uses SAS Business Intelligence, Microsoft Add-In, SAS Enterprise Guide and also has many ad hoc SAS users. There are many ETL processes and many SAS stored processes are spawned by the SAS applications. The data warehouse is a combination of SAS data and remote access to an RDBMS.



### **2.8.1 Evaluation process**

This example also appears to be a good candidate for a grid or distributed environment since growth appears to be an issue and there are many simultaneous, but different users. The technical questions to ask are:

- What is the I/O requirement per process? (Are you sure you are CPU bound?)
- How many users in each class and what are the response time requirements?
- What are the data volumes that each user needs to access and what are the typical throughput requirements in MB per second? (When possible draw a data flow diagram.)
- If applicable, what is the bandwidth of the connection to the database and or other data resources?
- Are there user groups that could be isolated to different grid nodes in order to improve response time?
- Are there priority users? (Grid scheduling can help with this.)

### **2.8.2 Design approach**

- Make sure that all the SAS applications can be executed on a grid. Implementing the SAS applications on the grid should not impact the way the users access SAS today.
- Estimate the number of CPUs and I/O storage that need (estimate from data you can get from the existing environment – be sure to factor in growth).
- Are there tasks that can't be run inside of current runtime requirements? If so, can these tasks be parallelized to execute across multiple nodes (parallelism inside the process itself).
- Draw a data flow diagram for the problem: include user types, throughput, runtime, and data volume requirements.
- Analyze your data flow diagram and begin to design your hardware architecture.
- Test your design in a lab or validate it by comparing to existing reference architectures.

### **2.8.3 Sample solution**

This problem is an ideal candidate for a SAS grid implementation and probably best suited for CFS or hybrid storage architecture. Although this is a simplified example, the data volumes listed here are better suited for these two types of storage (higher throughput requirements). However, if the data volumes are lower for some applications / user types, it could be possible to use NAS for some of the user groups. This could help reduce overall cost, but would also increase complexity if you use multiple storage types.

An example solution for this would be multiple grid nodes attached via a fiber channel switch to a large storage array(s). A CFS or hybrid shared file system would be used to provide for file sharing across the grid nodes. SAS<sup>®</sup> Grid Manager would be used in combination with the SAS products to deploy and manage the users across the grid.

## 3 Conclusion

Grid distributed systems can be implemented on many different storage technologies. It is important to always spend extra time when designing and configuring your storage architecture to meet your applications requirements. Improper storage configuration is the number one reason for poor performing SAS grid distributed environments. The guidelines outlined in this paper can help you design a successful solution.

### 3.1 Authors

Cheryl Doninger      Research and Development      [Cheryl.doninger@sas.com](mailto:Cheryl.doninger@sas.com)

Tom Keefer      Enterprise Excellence Center      [Tom.keefer@sas.com](mailto:Tom.keefer@sas.com)

# Appendix A: Storage basics and definitions

Storage systems are a complex and detailed part of today's information technology systems. This section defines some of the background and definitions that are required to understand the concepts for setting up a distributed environment.

**Storage device** – A device where data is physically stored. Data remains stored on a storage device even when the power is removed. This could also be referred to as an array, cabinet, or storage appliance.

**File system** – A component that manages the data stored on a storage device. The file system handles things like storing, retrieving, caching, and buffering data.

**Storage Client** – A computer or server that accesses storage over an Ethernet or storage area network. In a grid environment this would be referred to as a grid node.

**DAS** – This stands for Direct Attached Storage. Direct attached storage is one or more disk drives located within a computer cabinet and connected directly to the CPU (this also includes internal storage).

Examples file systems for direct attached storage: AIX JFS, AIX JFS2, Veritas VXFS, and Sun Microsystems ZFS.

Note: File systems all have support matrices. Refer to the vendor support matrix of a particular file-system to determine whether your environment is supported.

**JBOD** – This stands for Just a Bunch of Disks. This is an older acronym used when talking about a group of hard drives attached directly to a computer via a fiber channel, IDE, or SCSI connection. These disks are connected without the benefit of features typically found in advanced storage subsystems like file cache and or RAID controllers. These types of hard disks are managed directly by operating system tools and require CPU resources from the host computer system to manage.

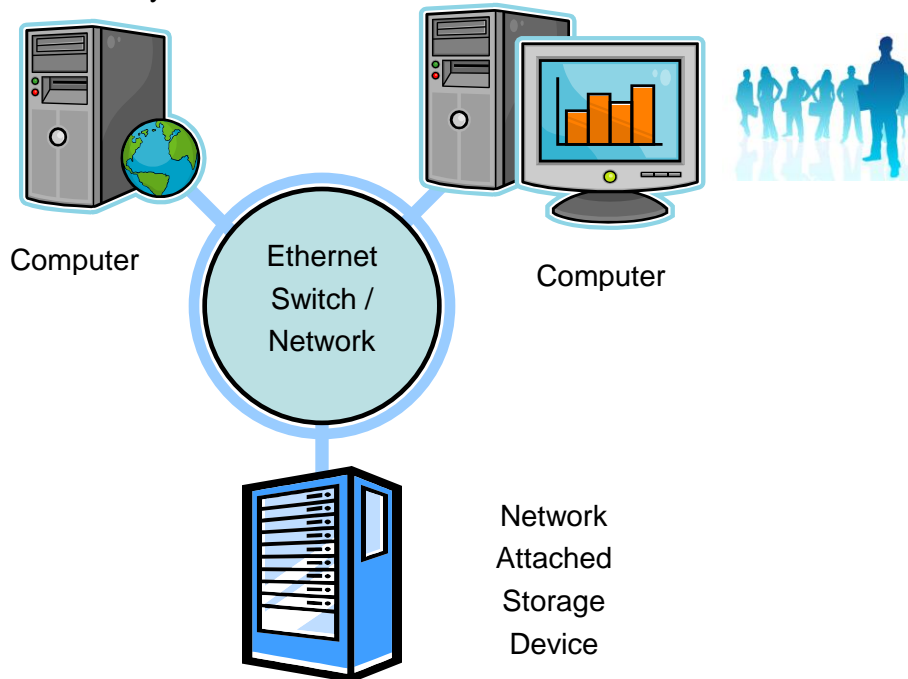
**Switch** – A switch is a device used to connect multiple devices or computers to a network. The switch is essentially a central connection point for these devices. Ethernet and Fiber Channel switches are the most common types of switches used in computer architecture. Ethernet switches are used in TCP/IP networking and Fiber Channel switches are primarily used for storage networks.

**NAS** – This stands for Network Attached Storage. NAS systems are typically computers with a specialized operating system designed to efficiently manage and share large amounts of storage over a network. NAS systems are usually accessed over a TCP/IP computer network rather than being directly connected to a computer. This allows multiple computers to share the same storage space which minimizes overhead by allowing centralized management of the hard disks. NAS systems are accessed over Ethernet using a file based protocol such as NFS or Microsoft Common Internet File System (CIFS). The performance of NAS systems depends heavily on memory cache (the equivalent of RAM) and network interface overhead (the speed of the router and the network cards). The benefit

is that the NAS system can share storage with any computer on the network.

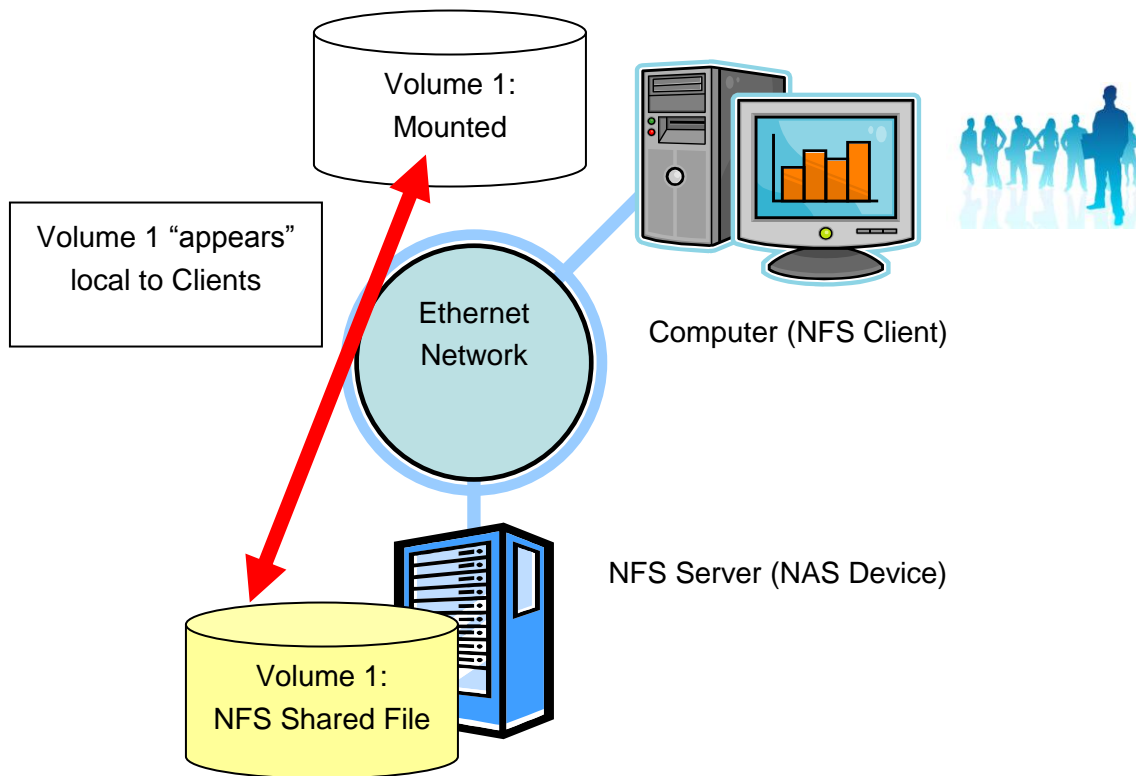
The disadvantage of NAS is that any network performance inefficiencies affect storage access for all systems accessing the network storage.

Network Appliance and EMC are examples of companies that sell NAS devices. NAS devices come with the file-system: NFS for UNIX or CIFS for Windows.



**Figure 5: Network Attached Storage (NAS) diagram – shared data**

NFS – stands for Network File System. NFS is the built in network file system in a UNIX and or Linux operating system (CIFS is the Windows equivalent). This UNIX standard was developed by Sun Microsystems Inc. NFS allows computers to access data over a network as easily as if the data is located on the local disks. NFS client capability is typically included for free with most major UNIX operating systems.



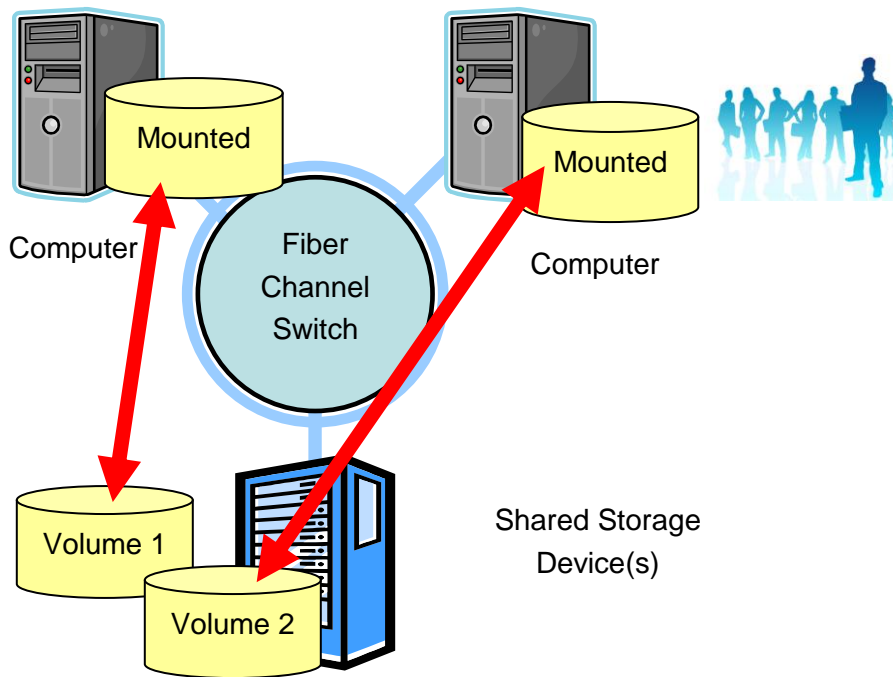
**Figure 6: Network File System (NFS) Diagram – Shared Data**

**CIFS – Common Internet File System** – CIFS is a network file system designed by Microsoft for sharing data across a local network (similar to NFS but for Windows). SMB (Server Message Block) was its predecessor (designed by IBM and Microsoft). The previous diagram for NFS also applies to CIFS. Many NAS devices also support CIFS as a mechanism for sharing files to Windows clients. CIFS client capability is built into all Microsoft Operating Systems.

**SAN – Storage Area Network** SANs enable multiple hosts to share a storage device but not data or the same physical storage blocks. It is important to note that a shared file-system is required in conjunction with a SAN to share data between multiple hosts. SANs use a block-based protocol which generally runs over an independent, specialized storage network, but not Ethernet. SANs typically use a fibre channel switch to allow multiple hosts to connect to the same storage device.

When a shared file system is installed in conjunction with a SAN, data can be shared by multiple hosts. Examples of shared file systems include: Redhat GFS (Linux); IBM GPFS (AIX); HP Polyserve Matrix (Windows or Linux) and Sun Microsystems QFS (Solaris). It is important to check to determine whether your operating system, storage hardware, and network equipment are supported by your shared file system choice. Refer to the vendor support matrix of a particular file-system to see whether your environment is supported.

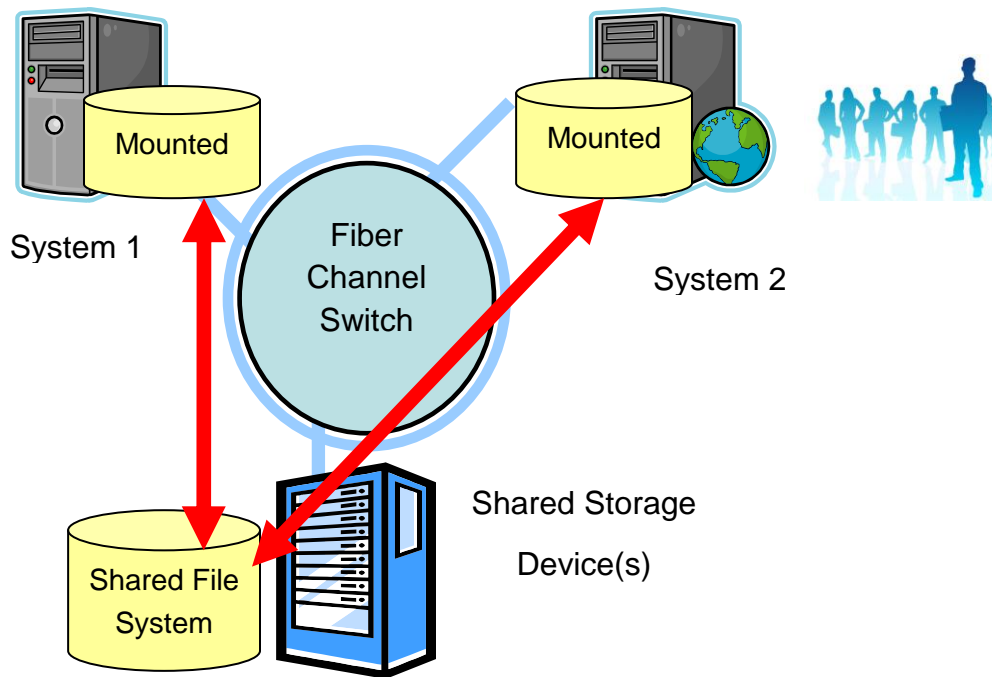
Note: The following file systems are NOT shared file systems. They are for single server access only: AIX JFS, AIX JFS2, Veritas VXFS, Sun Microsystems ZFS, Windows FAT32, and Windows NTFS. Shared file systems are typically not included with the operating system and are an additional expense.



**Figure 7: Storage Area Network (SAN) Diagram – No Data Sharing**

**CFS** – stands for Clustered File System (sometimes more logically referred to as a shared file system). A CFS allows multiple servers to simultaneously share the same data from a shared SAN. Clustered file systems are considered enterprise storage because they can enable scalable high-performance, high availability, simplified management, and data integrity. HP Polyserve Matrix Server, Sun QFS, IBM GPFS, and Linux GFS are all types of CFS. The critical features these file systems provide are data sharing along with data locking. Data locking prevents multiple servers from writing to the same data/file at the same time.

The term cluster is used in the name as the shared file system was originally focused at sharing file systems between servers for high availability. Over time CFS has become more widely used as a mechanism to share data for computational collaboration across multiple compute engines (grid nodes).



**Figure 8: Clustered File System (CFS) or Shared File System Diagram – shared data**

**iSCSI – Internet or IP based storage protocol.** iSCSI allows for the transmission of storage protocols over standard TCP/IP networks (Ethernet). Storage can be easily shared across an existing network versus running new dedicated storage protocol connections. Although not as fast as dedicated or pure storage protocols, it has the advantage of lower cost and is easy to deploy on an existing network.

**Hybrid File Systems** – These file systems provide the same shared capability as a CFS, but they usually are a combination between NAS and SAN storage. Examples of these technologies are iSCSI based file systems like EMC’s Multi-Protocol File System (MPFSi). Storage clients accessing the file system are connected to only Ethernet cables which are in turn connected to storage devices which are connected to the network. Special network equipment and software help do the translation of Ethernet IP packets to storage protocol. These file systems vary by implementation. You should talk to your vendor of choice for exact details. Both NFS and CIFS can be supported in these environments which makes it easy to connect most operating systems (Windows or UNIX).



## Considerations when building a storage architecture to support A SAS server or SAS Grid



### Server questions

What is my estimated i/o throughput needs to support all of my simultaneous users?

What file systems do I need? How many?

Do the file systems need to be shared between users or grid nodes?

### Storage network questions

NAS (Ethernet), SAN (fiber) or Hybrid (both)?

What is the maximum sustainable throughput for the network equipment?

Number of connections? Type?

### Disk Array questions

Total Disks?

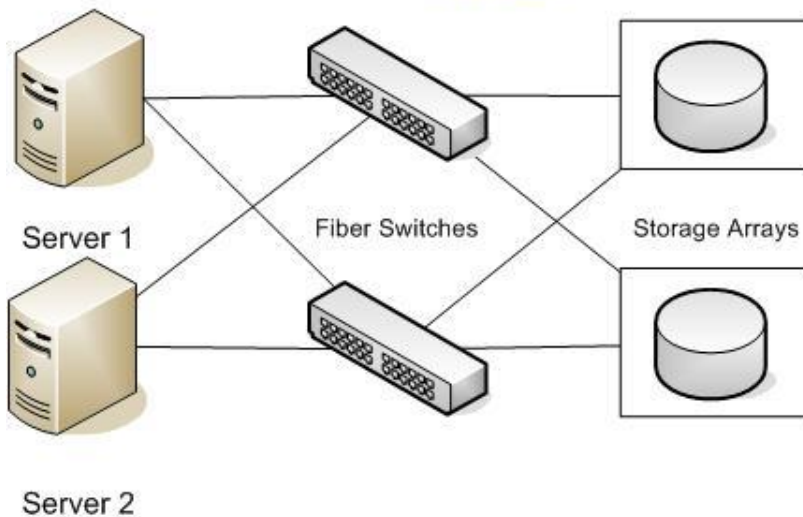
Speed of Disks?

What is max sustained 50 percent read/write throughput rate?

RAID Level of Luns?

Number of LUNS/volumes?

### Always draw a diagram



### Important questions to consider

What is your file system type (shared or not)?

What file system is it (QFS, GPFS, NFS, CIFS)?

Is my storage network separate from my general internal communication network?

**Figure 9: Considerations when building storage architecture**



**THE  
POWER  
TO KNOW.**

SAS INSTITUTE INC. WORLD HEADQUARTERS SAS CAMPUS DRIVE CARY, NC  
27513

TEL: 919 677 8000 FAX: 919 677 4444 U.S. SALES: 800 727 0025 [WWW.SAS.COM](http://WWW.SAS.COM)

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. © indicates USA registration. Other brand and product names are trademarks of their respective companies. Copyright © 2008, SAS Institute Inc. All rights reserved