# SAS® Data Integration Server Using Network Appliance™ Storage

## Performance and Scalability Case Study of SAS Data Integration Using Network Appliance Storage

**Saad Jafri, Network Appliance, Inc.**

**Darrell Suggs, Network Appliance, Inc.**

**John Hall, SAS Institute Inc.**

**Tom Keefer, SAS Institute Inc.**

**April 2007 | TR-3559**

## Executive Summary

As corporate IT departments continue to implement operational systems, available data stores continue to grow at an exponential rate. Leveraging these large volumes of operational data for Business Intelligence can be critical to the corporate bottom line. Making great business decisions requires quick access to answers to the right questions based on integrated data. Data integration makes great business decisions possible.

IT departments responsible for SAS data warehouse operations continually strive for better solutions to solve critical data availability, manageability, and total cost of ownership issues. Providing mission-critical data to business users in a timely and cost-effective manner while reducing overall business risk enables effective business decision making. Network Appliance (NetApp) data storage platforms provide such a storage solution to SAS Data Integration for ETL (extract, transform, and load) processes as part of SAS data warehouse operations, by delivering speed, availability, and ease of storage management.

This paper characterizes the performance and scalability of SAS Data Integration Server running on a four-CPU and eight-CPU system using Network File System (NFS) and Fibre Channel (FC) with NetApp storage. In tests that simulate real-world data warehouse operations, the results show that NetApp storage easily delivers the performance demands of SAS Data Integration Server operations using NFS and FC.

Furthermore, this paper highlights the NetApp value-added features that deliver data protection, availability and ease of storage management, all of which further decrease total cost of ownership. These value-added features include FlexVol®, FlexClone™, Snapshot™, RAID-DP™, and SnapMirror technologies delivered via the high-performance NetApp Data ONTAP 7G operating system.

## Table of Contents

## 1. Introduction and Summary

This technical report provides the results of a study conducted jointly by Network Appliance (NetApp) and SAS Institute Inc. to characterize the performance and scalability of SAS Data Integration Server using the star schema dimensional data model in SAS Scalable Performance Data Server® on NetApp storage over Network File System (NFS) and Fibre Channel (FC). To highlight this performance and scalability, a series of tests were conducted using a 2-socket dual core (4 CPUs) and a 4-socket dual core (8 CPUs) Sun™ Fire V40z AMD Opteron™ Solaris™ 10 system.

The workloads used in this study simulate building a full star schema dimensional data model in SAS Scalable Performance Data Server. This is a typical data model for feeding the analytic and business intelligence capabilities of the SAS Intelligence Architecture, forming an integrated architecture for end-to-end delivery of intelligence that decision makers can act on. SAS Scalable Performance Data Server is an SMP-enabled, multi-user data server for data storage in data warehousing applications. It provides a high-performance data store for large data sets used for decision support applications and analytics.

The primary result of this study shows that NetApp storage meets the performance demands of SAS Data Integration Server using NFS and FC while scaling from 4 host CPUs to 8 host CPUs. This report details specific test cases, test configurations, and performance and scalability results. Additionally, appendix sections contain detailed hardware and software configuration information.

## 2. Hardware and Software Environment

### Storage Configuration

The NetApp storage system configuration is shown in Table 1.

**Table 1) NetApp storage system configuration.**

| COMPONENT | DETAILS | |
|---|---|---|
| Operating System | Data ONTAP 7.2.1 | |
| Storage System Model | FAS3070A (Two (2) Controllers in active/active configuration) | |
| Disks | Twelve (12) shelves containing a total of 166 FC disks, each 144GB, 15K RPM | |
| Connections from storage controllers to disk shelves | Six (6) Active / Six (6) Passive FC ports were used to connect to the shelves | |
| Storage connections to Server | NFS tests used 4 × 1Gbps Ethernet | FC tests used 4 × 4Gbps FC ports |

### Server Configuration

The server configuration is shown in Table 2. A Sun Fire V40z system was used for the purposes of generating the workload and depicting a typical SMP server environment. Any supported SMP server platform could be used to generate the workload. Results may vary depending on the server platform used.

**Table 2) Server configuration.**

| COMPONENT | DETAILS | |
|---|---|---|
| Operating System | Solaris 10 Update 3 | |
| System Type | Sun Fire V40z | |
| SAS Server | SAS 9.1.3 Service Pack 4 for Solaris for x64<br>SAS Scalable Performance Data Server 4.4 | |
| FC Host Attach Kit | NetApp FC Solaris Host Utilities 4.1 for Native OS | |
| Processor | Two (2) dual core 2.6 GHz AMD Opteron | Four (4) dual core 2.6 GHz AMD Opteron |
| Physical RAM | 16GB | |
| Swap Space | 32GB | |
| Network Connections to Storage | Two (2) PCI-X Intel® Pro/1000 MT and 2 onboard 1GbE NICs for NFS (total 4 x 1 GbE NICs) | Two (2) dual-port 4Gb Qlogic 2462 HBAs for FC |

## 3. Test Description

The tests conducted are designed to emulate customer experiences in a SAS data warehouse environment and demonstrate the throughput and scalability of creating a SAS star schema dimensional data model. These tests stress different parts of the ETL and SAS Data Integration Server processes of a high-speed parallel build of a SAS star schema dimensional model in SAS Scalable Performance Data Server, which is a typical data model for feeding the analytic and business intelligence capabilities of the SAS Intelligence Architecture.

The tests simulate a real-world task of loading data into a SAS data warehouse through SAS Data Integration Server. The tests models two retail environments with three years' worth of weekly order information (156 retail order text files; 1 for each week). As shown in Table 3, the first environment is representative of a small star schema dimensional data model containing approximately 78GB of raw data stored in 1 million customer records. The second environment is representative of a medium-sized star schema dimensional data model containing approximately 312GB of data stored in 10 million customer records. The size categories were determined based on star schema sizes in typical customer environments.

**Table 3) Size of text input data for SAS star schema dimensional data model build.**

| TEST SIZE | CUSTOMER TABLE ROWS | NUMBER OF TRANSACTION FILES | SIZE OF EACH ORDER TRANSACTION FILE | RAW DATA SIZE (INPUT) |
|---|---|---|---|---|
| Small | 1 million | 156 | 0.5GB | 78GB |
| Medium | 10 million | 156 | 2GB | 312GB |

A retail enterprise would use this information to analyze and predict outcomes based on historical patterns; apply statistical methods to determine problems and trends; and apply optimization, scheduling, and simulations to achieve best results as well as other uses.

These tests do not include any data warehouse exploitation such as scoring or reporting.

Using data representative of small and medium-sized star schema dimensional data models, the process for each test consisted of 3 steps executed in the following order:

1. Creating four dimension tables: time; product; store and customer. These tables are created from text files and loaded into SAS Scalable Performance Data Server. The final process is to add indexes to the tables.
2. Loading in parallel data from 156 retail order text files (3 years of retail order data; 1 text file for each week of the year) into partitioned fact tables stored in SAS Scalable Performance Data Server. During this load, dimension key lookups are performed against the newly created dimension tables. These keys are added to the partition fact tables in SAS Scalable Performance Data Server.

3. Performing an SAS Scalable Performance Data Server dynamic cluster snap to tie the partitions together into one large fact table. This finalizes the star schema dimensional data model build for consumption by SAS solutions analytics or SAS Business Intelligence.

The primary objective of the tests is to show performance and scalability of the SAS star schema dimensional data model built using NetApp storage as the number of processors on the Solaris server increased from 4 CPUs to 8 CPUs. A total of 8 different tests were performed, shown in Table 4. The measurements taken for each test are as follows:

- Total elapsed time for each test. This was derived by adding the completion times for all 3 steps required for the star schema dimensional data model build.
- Peak throughput achieved during the star schema dimensional data model build.
- Host CPU utilization on the Solaris server during peak throughput.

**Table 4) Test number and associated details for each test executed.**

| TEST # | TEST SIZE | # HOST CPUS | # OF NETWORK CONNECTIONS FROM HOST TO NETAPP STORAGE | STORAGE INTERCONNECT |
|---|---|---|---|---|
| 1 | Small | 4 | Four (4) 1GbE | NFS |
| 2 | Medium | 4 | Four (4) 1GbE | NFS |
| 3 | Small | 8 | Four (4) 1GbE | NFS |
| 4 | Medium | 8 | Four (4) 1GbE | NFS |
| 5 | Small | 4 | Four (4) 4Gb FC | FC |
| 6 | Medium | 4 | Four (4) 4Gb FC | FC |
| 7 | Small | 8 | Four (4) 4Gb FC | FC |
| 8 | Medium | 8 | Four (4) 4Gb FC | FC |

## 4. NFS Test Results

Figure 1 shows the total time required to create the SAS star schema dimensional data model in SAS Scalable Performance Data Server using NFS for the following test configurations:

- 4 CPUs (2 dual cores) with a small and a medium-sized star schema dimensional data model
- 8 CPUs (4 dual cores) with a small and a medium-sized star schema dimensional data model

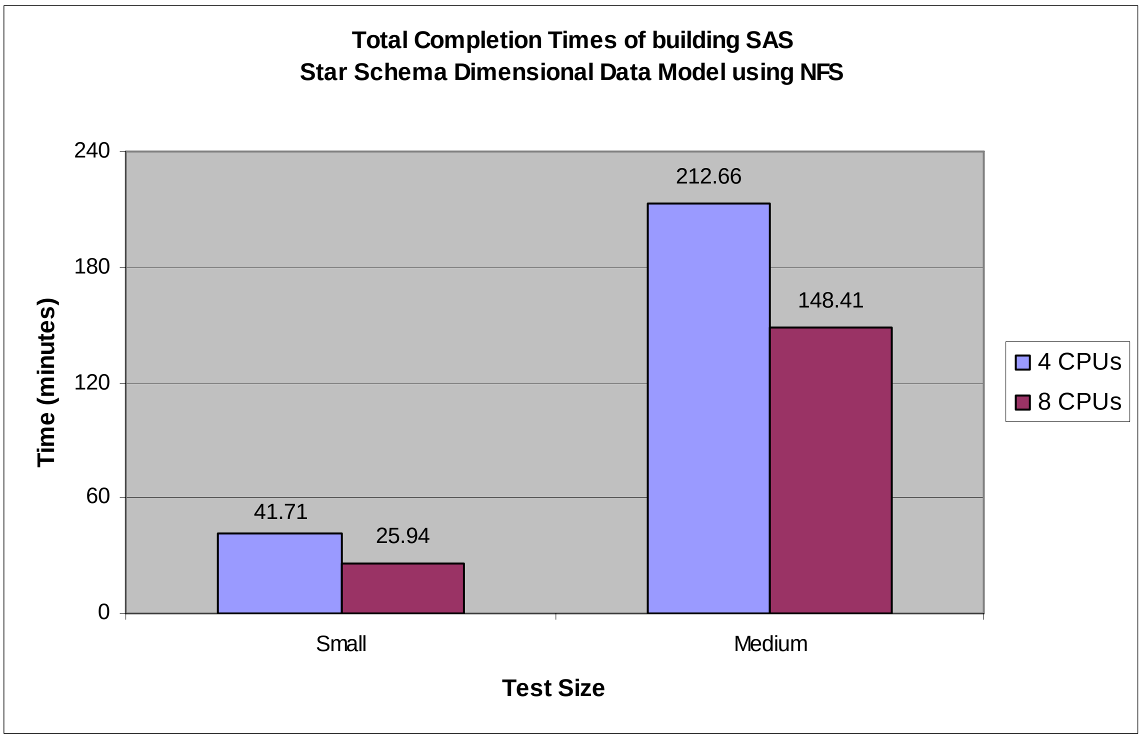


**Figure 1) Total completion times for building SAS star schema dimensional data model using NFS.**

Note: Shorter completion times indicate better performance in Figure 1.

Conclusions based on the results shown in Figure 1:

- Small data load completes 37% faster with 8 CPUs than with 4 CPUs.
- Medium data load completes 30% faster with 8 CPUs than with 4 CPUs.

Loading, validating, and indexing three years of retail data into SAS Scalable Performance Data Server from 156 retail order text files represent the majority of the time running these tests. This loading step consumes over 93% of the total time required for each test. The peak throughput was also reported during this step, as shown in Figure 2. Host CPU utilization during the peak throughput is shown in Figure 3.

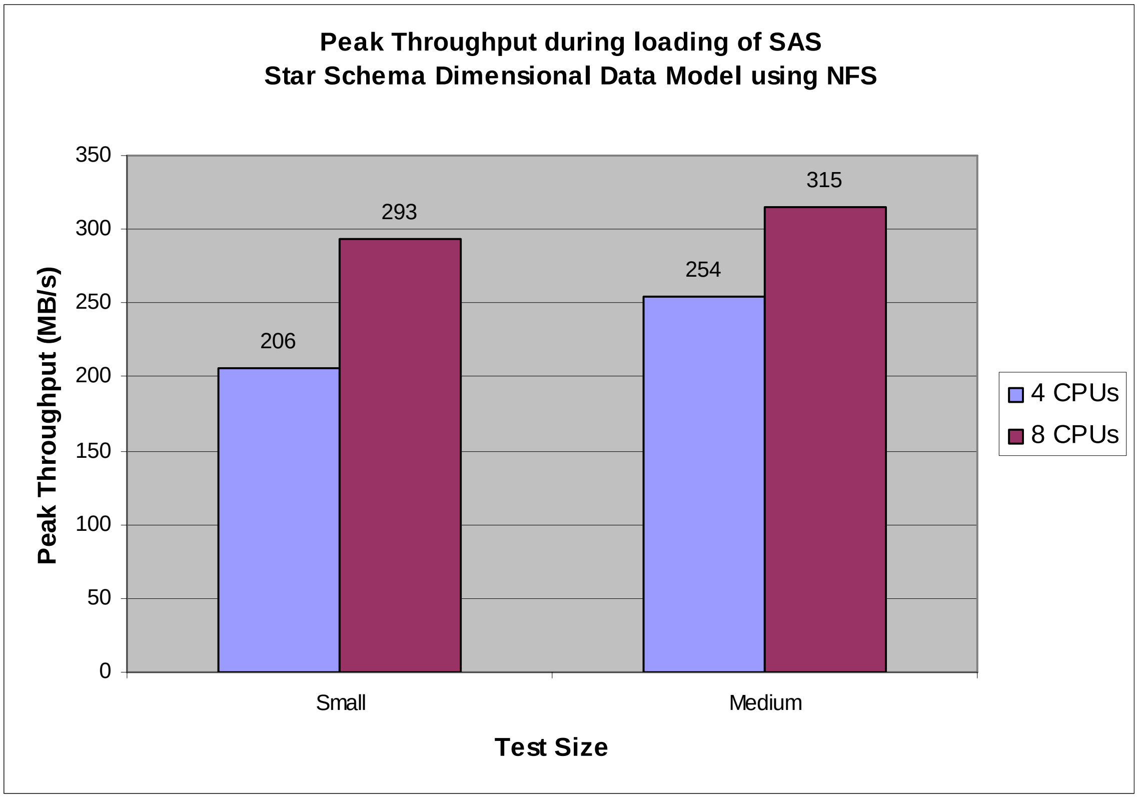


**Figure 2) Peak throughput during loading of SAS star schema dimensional data model using NFS.**

Conclusions based on the results shown in Figure 2:

- Peak load on small data improved by 42% (206MB/s to 293MB/s) when scaling from 4 to 8 CPUs.
- Peak load on medium data improved by 24% (254MB/s to 315MB/s) when scaling from 4 to 8 CPUs.

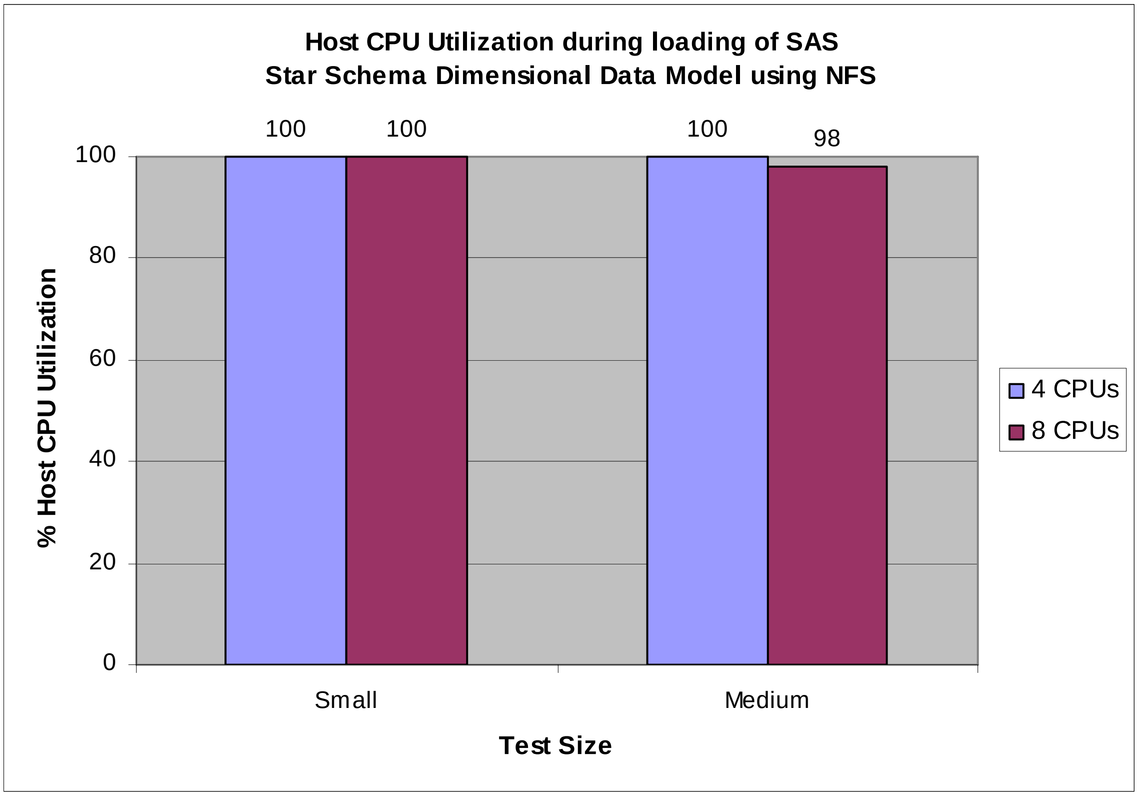


**Figure 3) Host CPU utilization during loading of SAS star schema dimensional data model using NFS.**

In all tests using NFS, the NetApp storage system easily met the demands of building the SAS star schema dimensional data model with resources to spare. Both disks and system bandwidth utilizations on the NetApp storage system were well below 50% for the duration of the tests. Figure 3 shows that the host CPUs were fully utilized during the tests, showing the limiting factor was the host CPUs in the overall scaling performance.

## 5. FC Test Results

The next set of tests used Fibre Channel (FC) storage interconnects. The FC tests were performed using the same methodology used for the NFS tests. In other words, the testing process, data volumes and CPU counts were constant in both sets of tests. The only difference in the two sets of tests is the storage interconnects.

Figure 4 shows the total time required to create the SAS star schema dimensional data model in SAS Scalable Performance Data Server using FC for the following test configurations:

- 4 CPUs (2 dual cores) with a small and a medium-sized star schema dimensional data model
- 8 CPUs (4 dual cores) with a small and a medium-sized star schema dimensional data model

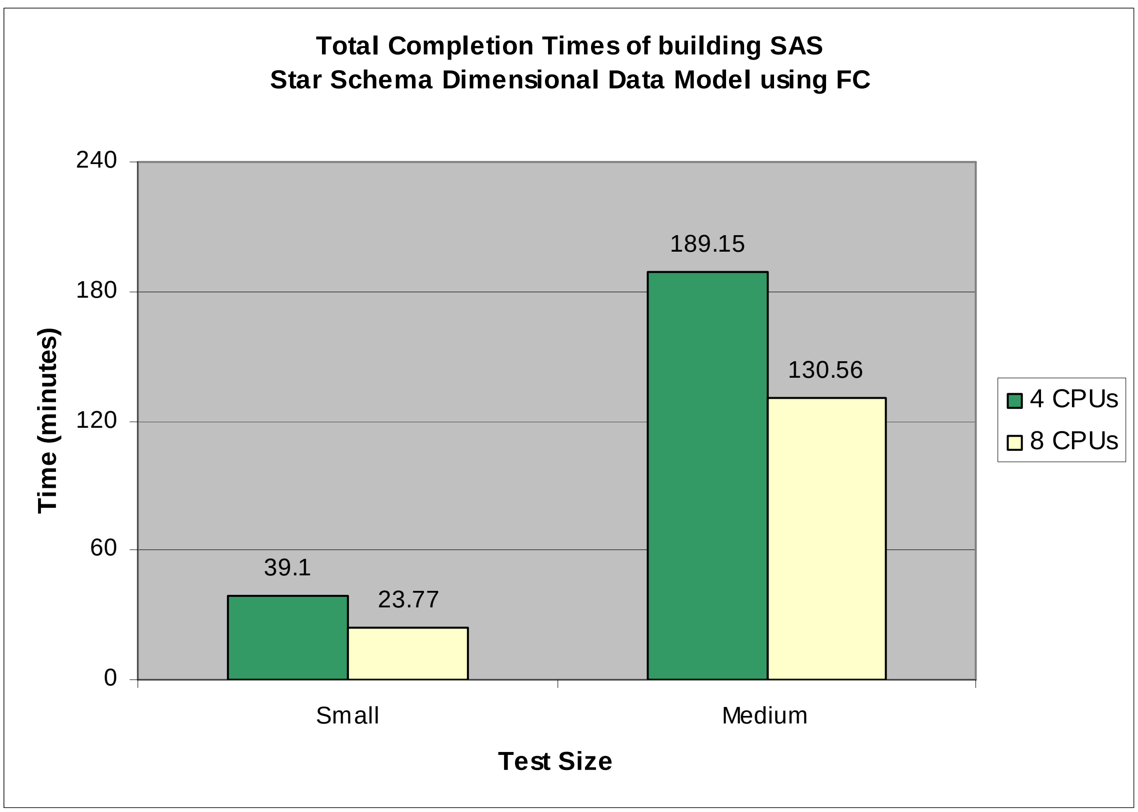


**Figure 4) Total completion times of building SAS star schema dimensional data model using FC.**

Note: Shorter completion times indicate better performance in Figure 4.

Conclusions based on results shown in Figure 4:

- Small data load completes 39% faster with 8 CPUs than with 4 CPUs.
- Medium data load completes 31% faster with 8 CPUs than with 4 CPUs.

As with the NFS tests, in the FC tests, loading, validating, and indexing three years of retail data into SAS Scalable Performance Data Server from 156 retail order text files also represent the majority of the time running these tests. This loading step consumes over 94% of the total time required for each test. The peak throughput was also reported during this step as shown in Figure 5. Host CPU utilization during the peak throughput is shown in Figure 6.

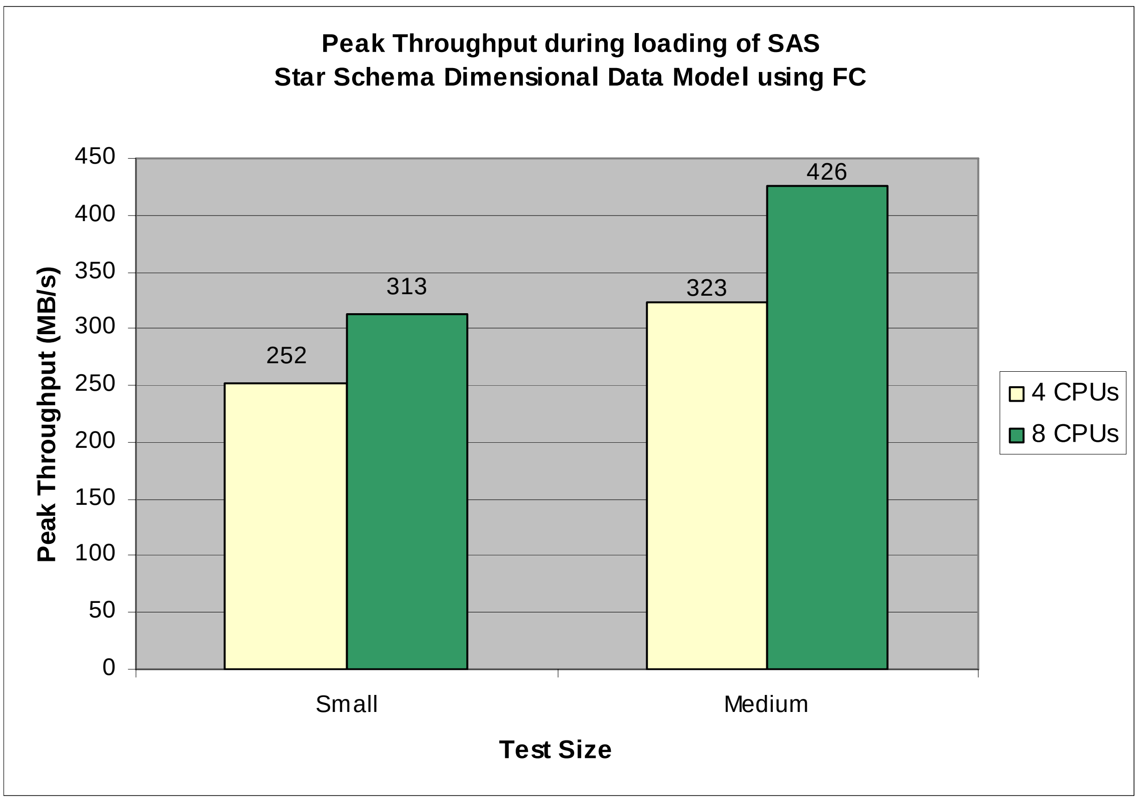


**Figure 5) Peak throughput during loading of SAS star schema dimensional data model using FC.**

Conclusions based on results in Figure 5:

- Peak load on small data improved by 24% (252MB/s to 313MB/s) when scaling from 4 to 8 CPUs.
- Peak load on medium data improved by 32% (323MB/s to 426MB/s) when scaling from 4 to 8 CPUs.

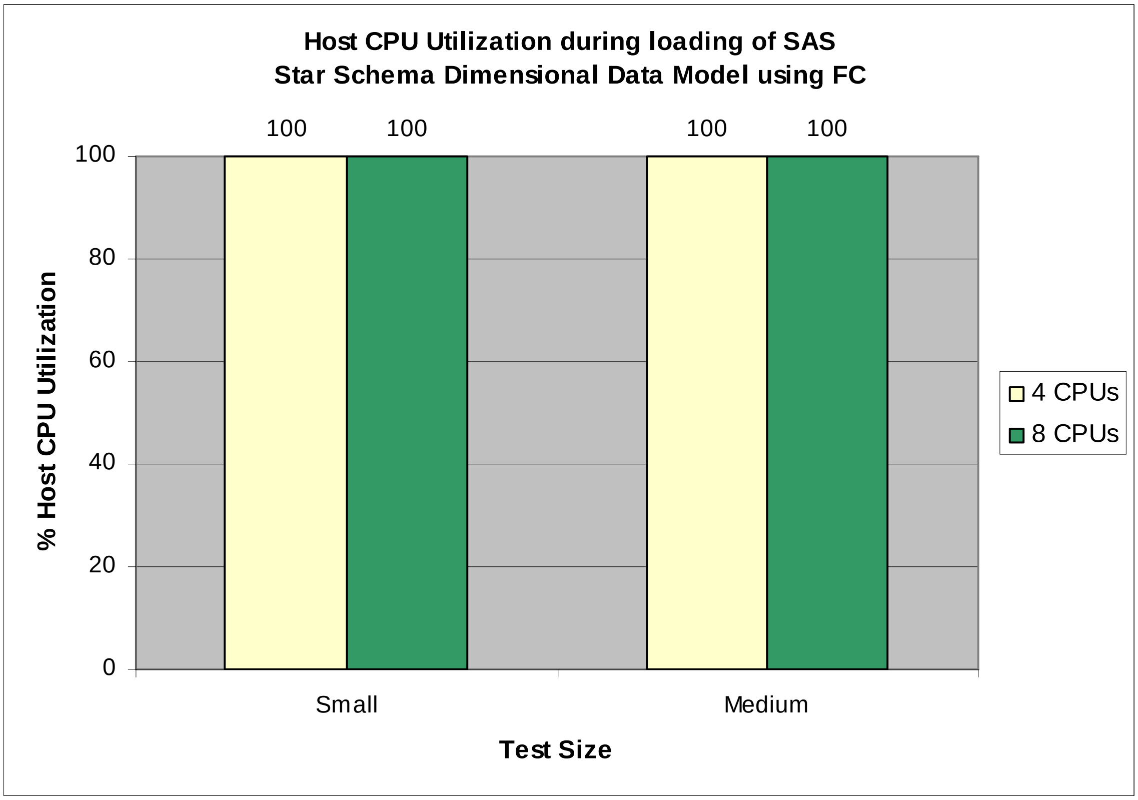


**Figure 6) Host CPU utilization during loading of SAS star schema dimensional data model using FC.**

In all tests using FC, the NetApp storage system easily met the demands of the SAS star schema build with resources to spare. Both disks and system bandwidth utilizations on the NetApp storage system were well below 50% for the duration of the tests. Figure 6 shows that the host CPUs were fully utilized during the tests, showing the limiting factor was the host CPUs in the overall scaling performance.

Figure 7 illustrates the difference in completion times of the SAS star schema dimensional data model build between NFS and FC for all 8 tests.

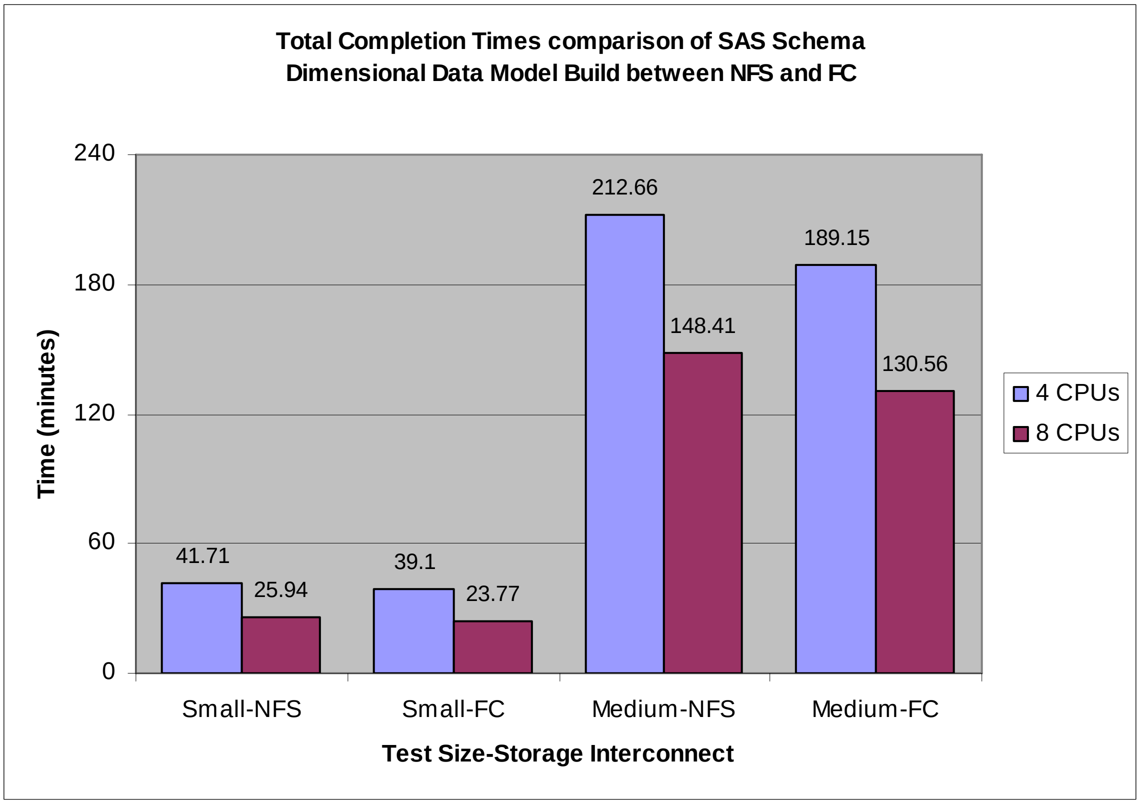


**Figure 7) Total completion times comparison of SAS star schema dimensional data model build between NFS and FC.**

Conclusions based on results in Figure 7:

- Completion times for small data ranged between 6% - 8% faster on FC compared to NFS.
- Completion times for medium data ranged between 11% - 12% faster on FC compared to NFS.

## 6. Value-Add features

In an enterprise environment, SAS data warehouse administrators and architects typically undertake the following activities on a routine basis:

- Data model definition and creation
- Data warehouse activities such as object creation and modification
- Capacity analysis and planning
- Data warehouse backup and recovery
- Data warehouse disaster recovery and planning
- Data warehouse performance tuning
- Data warehouse maintenance

Without the right data storage solution, each one of these tasks can be increasingly difficult and time consuming, resulting in difficult challenges which add to the high cost of business operations. Using NetApp storage platforms in a SAS data warehouse environment helps alleviate the challenges associated with each one of these activities. The end result is a reduction in the total cost of ownership to the business.

In addition to the high performance the NetApp storage platform delivers to the SAS data integration process, proven by these test results, the Network Appliance storage platform also brings value-add features that provide solutions to the activities associated with maintaining the integrity and the high availability of a data warehouse.

Table 5 shows a summary of NetApp value-add features that help alleviate challenges associated with the activities listed above.

**Table 5) Value-add features and solutions.**

| SAS DATA WAREHOUSE ADMINISTRATOR / ARCHITECT ACTIVITY | SAS DATA WAREHOUSE ADMINISTRATOR / ARCHITECT CHALLENGES | NETAPP SOLUTION |
|---|---|---|
| SAS data model definition, creation, and management | Change management of the underlying storage to match changes in the data model can be difficult and time consuming. | Flexible storage model that is easily adaptable to data model changes. |

**Table 5 (continued)) Value-add features and solutions.**

| DATA WAREHOUSE ADMINISTRATOR / ARCHITECT ACTIVITY | DATA WAREHOUSE ADMINISTRATOR / ARCHITECT CHALLENGES | NETAPP SOLUTION |
|---|---|---|
| SAS data warehouse activities such as object creation and modification | Creating mirror images or copies of SAS data warehouses can be time consuming, difficult and can consume valuable storage resources. | Easy, space-efficient, and relatively inexpensive way to make copies of a SAS data warehouses for a dev/test/prod environment and for recoverability. The key feature that facilitates this is FlexClone technology. |
| Capacity analysis and planning | Accurately forecasting storage requirements for SAS data warehouse is difficult. Over or under allocation of storage resources happens frequently.<br><br>Data layout is essential for optimal performance of SAS software.<br><br>Changing predefined storage allocations can be difficult and time consuming. It also introduces the potential for data loss. | Easy allocation, expansion, or contraction of storage using NetApp FlexVol technology.<br><br>Using FlexVol technology, you can seamlessly spread SAS data warehouse data files across multiple disk spindles, resulting in optimum performance.<br><br>Virtually eliminate the potential for data loss and SAS data warehouse down time with NetApp RAID-DP technology and controller clustering. |

**Table 5 (continued)) Value-add features and solutions.**

| DATA WAREHOUSE ADMINISTRATOR / ARCHITECT ACTIVITY | DATA WAREHOUSE ADMINISTRATOR / ARCHITECT CHALLENGES | NETAPP SOLUTION |
|---|---|---|
| SAS data warehouse backup and recovery | Backup images consume valuable storage resources.<br><br>The process used to produce backup images is typically very time and system resource intensive.<br><br>Production systems may have to be taken off-line while a backup image is made, resulting in loss of productivity.<br><br>Because of the overhead involved, backup images are not generated frequently, resulting in longer recovery times when portions of a data warehouse must be restored. | Use NetApp storage architecture, OS and Snapshot technology to:<br>▪ Reduce the time required to create backups to less than one minute<br>▪ Significantly reduce the amount of disk space needed to maintain backup images<br>▪ Reduce restore times by creating more frequent backups<br><br>Quickly restore a SAS data warehouse from a Snapshot backup image with NetApp SnapRestore® technology. |
| SAS data warehouse disaster recovery planning | Maintaining full copies of data for recovery purposes consumes lots of valuable storage resources.<br><br>Keeping backup copies synchronized with current data is complex, difficult and expensive. | Create a mirrored image of a SAS data warehouse on a separate system that can be stored off-site with SnapMirror technology.<br><br>By synchronizing this mirrored copy with the primary SAS data warehouse at regular intervals, the mirror copy can become the primary SAS data warehouse environment in the event the original SAS data warehouse environment is destroyed or severely damaged. |

**Table 5 (continued)) Value-add features and solutions.**

| DATA WAREHOUSE ADMINISTRATOR / ARCHITECT ACTIVITY | DATA WAREHOUSE ADMINISTRATOR / ARCHITECT CHALLENGES | NETAPP SOLUTION |
|---|---|---|
| SAS data warehouse performance tuning | Data loading and data access cannot keep up with the response times required in application Service Level Agreements (SLAs). Duplicate copies of the SAS data warehouse are needed to maintain availability required in SLAs.<br><br>Data I/O is a bottleneck to SAS data warehouse performance. | SAS data warehouse performance can be improved measurably and quickly because NetApp storage systems make use of all available spindles.<br><br>Using FlexVol, SAS data warehouse data files can be spread across multiple disk spindles, resulting in optimum performance. |
| SAS database maintenance | Duplicating data for testing software migrations, bug fixes, upgrades or patches is time consuming, costly and uses valuable storage resources.<br><br>Migrating SAS data warehouse to newer application versions without negatively impacting the existing SAS data warehouse can be challenging<br><br>Re-allocation of storage is difficult and time consuming. | Easy and relatively inexpensive way to make copies of a production SAS data warehouse through FlexClone technology without consuming valuable storage space or impacting the SAS data warehouse.<br><br>Use FlexClone clones of your SAS data warehouse FlexClone clones to test migrations, apply bug fixes, upgrades and patches.<br><br>With FlexVol, you can easily reallocate storage used by your SAS data warehouse and its data files. |

## 7. Conclusions

This paper demonstrates that Network Appliance storage systems easily deliver the performance demands of SAS Data Integration Server for ETL (extract, transform, and load) processes using Network File System (NFS) and Fibre Channel (FC) as the number of host CPUs scales up.

The data in this report shows that NFS and FC are both viable storage system interconnects for SAS ETL operations. As expected, the FC system was faster, but the difference in completion times of full SAS star schema dimensional data model build ranged only from 6% to 12% across the various tests. In both the NFS and FC test runs performance was inhibited only by available server CPU resources while sufficient unused bandwidth remained in the NetApp storage systems.

Regardless of the storage system interconnect, NetApp storage systems are uniquely positioned to bring unparalleled flexibility and scalability to enterprise-class SAS data warehouse environments. NetApp storage systems provide a feature-rich integrated platform and data protection environment, while supporting customers' high-performance storage needs.

## Appendix A: Network Diagrams

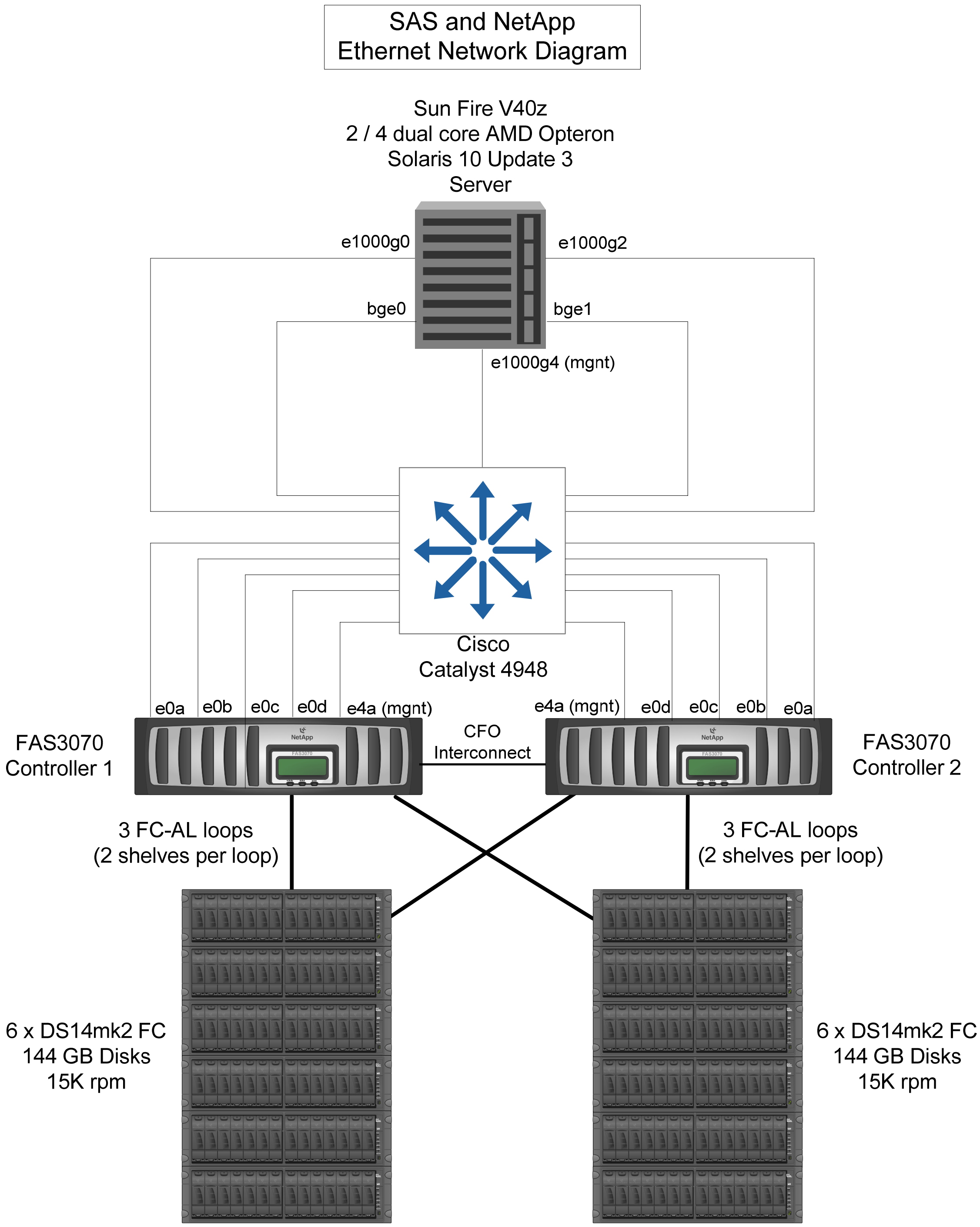


**Figure 8) Network diagram for NFS tests.**

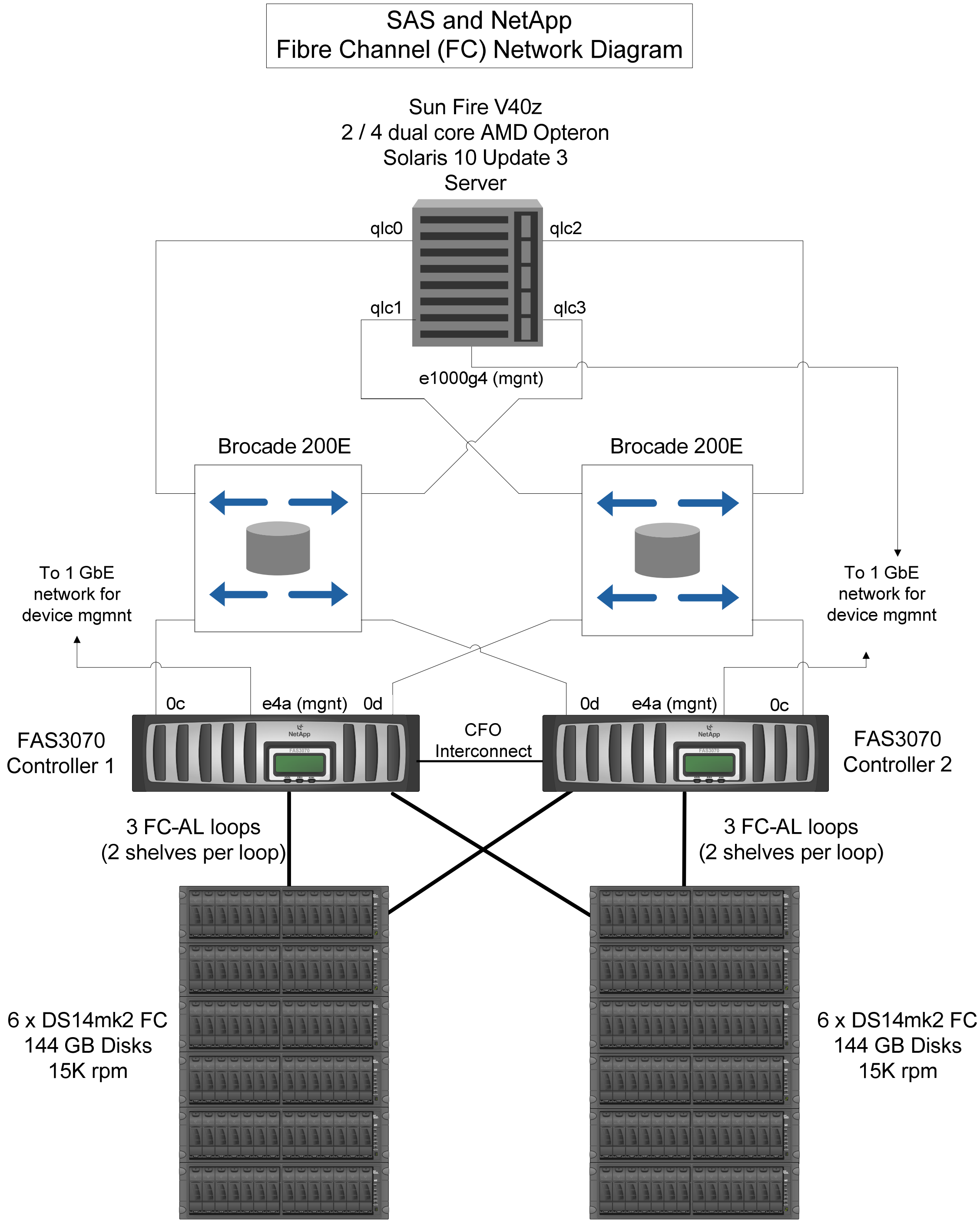


**Figure 9) Network diagram for FC tests.**

## Appendix B: Storage System Layout for NFS Tests

Table 6 shows the storage system layout for the NFS tests.

**Table 6) Storage system layout for NFS tests.**

| NETAPP STORAGE CONTROLLER | AGGREGATE NAME | VOLUME NAME | SIZE | DESCRIPTION |
|---|---|---|---|---|
| Controller 1 | Aggr0 | | 3 Disks | |
| | | /vol/vol0 | 90GB | Root Volume |
| | Aggr1 | | 80 Disks | |
| | | /vol/vol1 | 6357GB | SAS Data Volume |
| Controller 2 | Aggr0 | | 3 Disks | |
| | | /vol/vol0 | 90GB | Root Volume |
| | Aggr1 | | 80 Disks | |
| | | /vol/vol1 | 6357GB | SAS Data Volume |

All RAID groups were 16 disks in size and configured to use double parity. Volume guarantee was used on vol1 volume containing SAS data.

## Appendix C: Storage System Layout for FC Tests

Table 7 shows the storage system layout for FC tests.

**Table 7) Storage system layout for FC tests.**

| NETAPP STORAGE CONTROLLER | AGGREGATE NAME | VOLUME NAME | LUN | SIZE | DESCRIPTION |
|---|---|---|---|---|---|
| Controller 1 | Aggr0 | | | 3 Disks | |
| | | /vol/vol0 | | 90GB | Root Volume |
| | Aggr1 | | | 80 Disks | |
| | | /vol/vol1 | | 6357GB | SAS Data Volume |
| | | | /vol/vol1/sasdata | 100GB | SAS code, scripts, metadata, logs and input text files for dimension creation |
| | | | /vol/vol1/sasinput1 | 2000GB | Retail order text files |
| | | | /vol/vol1/sasoutput1 | 100GB | SAS output |
| | | | /vol/vol1/sasoutput3 | 100GB | SAS output |
| | | | /vol/vol1/sasoutput5 | 100GB | SAS output |
| | | | /vol/vol1/sasoutput7 | 100GB | SAS output |
| Controller 2 | Aggr0 | | | 3 Disks | |
| | | /vol/vol0 | | 90GB | Root Volume |
| | Aggr1 | | | 80 Disks | |
| | | /vol/vol1 | | 6357GB | SAS Data Volume |
| | | | /vol/vol1/sasinput2 | 2000GB | Retail order text files |
| | | | /vol/vol1/sasoutput2 | 100GB | SAS output |
| | | | /vol/vol1/sasoutput4 | 100GB | SAS output |
| | | | /vol/vol1/sasoutput6 | 100GB | SAS output |
| | | | /vol/vol1/sasoutput8 | 100GB | SAS output |

All RAID groups were 16 disks in size and configured to use double parity. File guarantee was used on the volume containing LUNs for SAS data instead of the default volume guarantee because LUNs have space reservations guarantee enabled by default. Since LUNs are essentially files within the volume, there was no need to have space guarantee at the volume level.

## Appendix D: NFS Configuration

To achieve a balanced workload across the NetApp storage controllers, half of the SAS data input and output files were placed on NetApp storage controller 1 and the other half on NetApp storage controller 2.

The following steps were taken to map the NFS mountpoints used by SAS to the NetApp storage volumes:

1. On the NetApp storage controllers, an IP address and a unique hostname were assigned to each network interface port. Each IP address was configured on a separate subnet in order to ensure appropriate network traffic flows through the desired interface.
2. On the Sun server, an IP address was assigned to each of its network interface ports. Each IP address was defined on a separate subnet that matched a corresponding interface on a storage controller. In other words, there was one server IP address and one storage controller IP address defined in each subnet.

To separate management traffic from SAS data I/O, dedicated management NICs were used on the server and NetApp storage controllers. Table 8 illustrates IP address configuration on the Sun Solaris 10 server and storage controllers used for SAS traffic.

**Table 8) IP address and NIC assignments for NFS tests.**

| FOR SAS WORKLOAD | | | | |
|---|---|---|---|---|
| **NETAPP STORAGE CONTROLLER** | **PHYSICAL NETWORK INTERFACE** | **IP ADDRESS** | **HOSTNAME** | **DESCRIPTION** |
| Controller 1 | e0a | 10.61.100.52 | controller1-e0a | For SAS input traffic to the server |
| | e0c | 10.61.101.52 | controller1-e0c | For SAS output traffic from the server |
| Controller 2 | e0a | 10.61.102.52 | controller2-e0a | For SAS input traffic to the server |
| | e0c | 10.61.103.52 | controller2-e0c | For SAS output traffic from the server |
| **SAS HOST** | | | | |
| Server | e1000g0 | 10.61.100.51 | server-e1000g0 | For SAS input traffic from NetApp storage controller 1 |
| | e1000g2 | 10.61.101.51 | server-e1000g2 | For SAS input traffic from NetApp storage controller 2 |
| | bge0 | 10.61.102.51 | server-bge0 | For SAS output traffic to NetApp storage controller 1 |
| | bge1 | 10.61.103.51 | server-bge1 | For SAS output traffic to NetApp storage controller 2 |

Directories were created to hold SAS data under /vol/vol1 on NetApp storage controllers 1 and 2 as shown in Table 9.

**Table 9) Directory layout under /vol/vol1 on NetApp storage controllers 1 and 2 for NFS mounts.**

| STORAGE CONTROLLER | VOLUME NAME | DIRECTORY NAME | DESCRIPTION |
|---|---|---|---|
| Controller 1 | /vol/vol1 | | SAS Data Volume |
| | | sasdata | SAS code, scripts, metadata, logs and input text files for dimension creation |
| | | sasinput1 | Retail order text files |
| | | sasoutput1 | SAS output |
| | | sasoutput3 | SAS output |
| | | sasoutput5 | SAS output |
| | | sasoutput7 | SAS output |
| Controller 2 | /vol/vol1 | | SAS Data Volume |
| | | sasinput1 | Retail order text files |
| | | sasoutput2 | SAS output |
| | | sasoutput4 | SAS output |
| | | sasoutput6 | SAS output |
| | | sasoutput8 | SAS output |

NFS mountpoints were created on the server and mapped to NetApp storage volumes as shown in Table 10. This mapping was done in the /etc/vfstab file on the server, and the following NFS mount options were used for each mountpoint:

forcedirectio,llock,bg,hard,rsize=32768,wsize=32768,vers=3,proto=tcp

**Table 10) NFS mountpoint to NetApp storage volume mapping.**

| NFS MOUNTPOINT | NETAPP STORAGE VOLUME |
|---|---|
| /sasdata | controller1-e0a:/vol/vol1/sasdata |
| /sasinput1 | controller1-e0a:/vol/vol1/sasinput1 |
| /sasinput2 | controller2-e0a:/vol/vol1/sasinput2 |
| /sasoutput1 | controller1-e0c:/vol/vol1/sasoutput1 |
| /sasoutput2 | controller2-e0c:/vol/vol1/sasoutput2 |
| /sasoutput3 | controller1-e0c:/vol/vol1/sasoutput3 |
| /sasoutput4 | controller2-e0c:/vol/vol1/sasoutput4 |
| /sasoutput5 | controller1-e0c:/vol/vol1/sasoutput5 |
| /sasoutput6 | controller2-e0c:/vol/vol1/sasoutput6 |
| /sasoutput7 | controller1-e0c:/vol/vol1/sasoutput7 |
| /sasoutput8 | controller2-e0c:/vol/vol1/sasoutput8 |

The following 2 lines were added to the /etc/system file on the server to increase the value of NFS tunable kernel parameters from their default values.

```
set nfs:nfs3_max_threads=64
set nfs:nfs3_nra=64
```

The nfs3_max_threads parameter controls the number of kernel threads that perform asynchronous I/O for the NFS version 3 clients. This number was increased to 64 from the default of 8 to achieve a higher number of simultaneous outstanding I/O operations at any given time.

The nfs3_nra parameter controls the number of read-ahead operations that are queued by the NFS version 3 clients when sequential access to a file is discovered. These read-ahead operations increase concurrency and read throughput. Each read-ahead request is generally for one logical block of file data. This number was increased to 64 from the default of 4 to achieve a higher number of outstanding read-ahead requests for a specific file at any given time.

## Appendix E: FC Configuration

After installing the FC HBAs in the Sun Solaris 10 server, physical network connections were made as per the FC network diagram shown in Figure 9 in Appendix A of this document.

The following steps were taken to configure FC connectivity between the server and the NetApp storage controllers:

1. Used the default native Solaris 10 Qlogic driver (qlc) for the 2 dual ported FC HBAs on the server.
2. Used the default native Solaris 10 multipathing (MPxIO) software for path management, single device instance, and failover.
3. Used the default single_image cfmode on both NetApp storage controllers.
4. Installed the NetApp FC Solaris Host Utilities 4.1 for Native OS on the server. Executed the “basic_config” script found in the install directory of these utilities. This script adds NetApp specific SCSI tunable in the /kernel/drv/sd.conf file on the server. Specifically, the following lines are added at the end of the file:

   sd-config-list="NETAPP LUN", "netapp-sd-config";

   netapp-sd-config=1,0x9c01,64,0,0,0,0,0,0,0,0,0,300,30,30,0,0,8,0,0;

   The line in the first bullet above is vendor and product identification of the device which is used in response to SCSI inquiry command. The line in the second bullet sets the SCSI device queue depth to 64 signified by the third comma separated field.

   A reboot of the server was needed for the above settings to take effect.
5. Created a single igroup on each NetApp storage controller, containing all 4 WWPNs of the FC HBA ports on the server.
6. Enabled asymmetric logical unit access (ALUA) on each igroup created on the NetApp storage controllers. ALUA defines a standard set of SCSI commands for discovering and managing multiple paths to LUNs on FC.
7. Created LUNs on NetApp storage as per the data shown in Table 7 in Appendix C.
8. Mapped each LUN to the appropriate igroup using a unique LUN ID. After this mapping, the server was rebooted to discover all the LUNs and associated FC paths. Using the sanlun utility included as part of the NetApp FC Solaris Host Utilities 4.1 for Native OS, we verified that each LUN was displayed as a single instance even though it was visible to the server via multiple paths to the NetApp storage controllers. The following command was used on the server to verify there was only one unique disk/raw-device for each LUN:

   sanlun lun show
9. Ran the Solaris 10 format utility to create a Solaris 10 partition on each LUN. Slice 1 was allocated approximately 200MB and tagged as root partition, and slice 6 was allocated the remaining space and tagged as the usr partition (Note:  This is not /usr partition or directory of the OS.  It is just a tag to identify slice 6 where user data will reside).

10. Created a ufs file system on the slice 6 partition of each LUN using its respective raw device, which was determined by running the following command on the server:

    sanlun lun show

To achieve a balanced workload across the 2 NetApp storage controllers, half of the SAS input and output data files were put in LUNs on NetApp storage controller 1 and the other half in LUNs on NetApp storage controller 2.

Table 11 shows the mapping of LUNs to mountpoints used by SAS.

**Table 11) Mapping of LUNs to mountpoints.**

| STORAGE CONTROLLER | LUN | MOUNTPOINT | DESCRIPTION |
|---|---|---|---|
| Controller 1 (sas1) | /vol/vol1/sasdata | /sasdata | SAS code, scripts, metadata, logs and input text files for dimension creation |
| | /vol/vol1/sasinput1 | /sasinput1 | Retail order text files |
| | /vol/vol1/sasoutput1 | /sasoutput1 | SAS output |
| | /vol/vol1/sasoutput3 | /sasoutput3 | SAS output |
| | /vol/vol1/sasoutput5 | /sasoutput5 | SAS output |
| | /vol/vol1/sasoutput7 | /sasoutput7 | SAS output |
| Controller 2 (sas2) | /vol/vol1/sasinput1 | /sasinput1 | Retail order text files |
| | /vol/vol1/sasoutput2 | /sasoutput2 | SAS output |
| | /vol/vol1/sasoutput4 | /sasoutput4 | SAS output |
| | /vol/vol1/sasoutput6 | /sasoutput6 | SAS output |
| | /vol/vol1/sasoutput8 | /sasoutput8 | SAS output |

For each LUN in Table 11, entries for the slice 6 partition of the respective Solaris disk (/dev/dsk/c*) and raw device (/dev/rdsk/c*) reported by the “sanlun lun show” command were added to the /etc/vfstab file. The forcedirectio mount option was also used for each mountpoint. To illustrate this, below are the relevant contents of the /etc/vfstab file for the /vol/vol1/sasdata LUN with the column heading in bold:

- **device to mount**

  /dev/dsk/c6t60A98000486E5365566F2D6D394E432Fd0s6

- **device to fsck**

  /dev/rdsk/c6t60A98000486E5365566F2D6D394E432Fd0s6

- **mountpoint**

  /sasdata

  **FS type**

  ufs
- **fsck pass**

  -
- **mount at boot**

  yes
- **mount option**

  forcedirectio

## Appendix F: SAS Configuration

SAS 9.1.3 Service Pack 4 for Solaris for x64 and SAS Scalable Performance Data Server 4.4 software was installed in the /sas9 directory on the Sun Solaris 10 server and licensed appropriately. After installing the SAS software, the relevant directories and symbolic links shown in Table 12 were created.

**Table 12) Relevant directories and symbolic links created for SAS ETL tests.**

| DIRECTORIES CREATED UNDER / | SYMBOLIC LINKS CREATED UNDER /SASDATA | DIRECTORIES CREATED UNDER /SASDATA |
|---|---|---|
| sasinput1 | input1 -> /sasinput1 | bin_star |
| sasinput2 | input1 -> /sasinput1 | code |
| sasoutput1 | output1 -> /sasoutput1 | input |
| sasoutput2 | output2 -> /sasoutput2 | logs |
| sasoutput3 | output3 -> /sasoutput3 | spdsmeta |
| sasoutput4 | output4 -> /sasoutput4 | |
| sasoutput5 | output5 -> /sasoutput5 | |
| sasoutput6 | output6 -> /sasoutput6 | |
| sasoutput7 | output7 -> /sasoutput7 | |
| sasoutput8 | output8 -> /sasoutput8 | |

The symbolic links were created under /sasdata in order to redirect the application to appropriate locations on the NetApp storage system during the SAS star schema dimensional data model build process. SAS code executed during the tests expected input and output data directories to be located under the /sasdata directory on the server.

The following environment variables were set for the root user as all of the SAS jobs were run as root:

```
PATH=$PATH:/opt/NTAP/SANToolkit/bin:/usr/sanlun/man:/usr/ucb
export PATH
##########################################
#TEST ENVIRONMENT SETTINGS
##########################################
SUITE=/sasdata
export SUITE
LOGS=$SUITE/logs
```

```
export LOGS
CODE=$SUITE/code
export CODE
SAS=/sas9/SAS_9.1
export SAS
INPUT=/sasdata/input
export INPUT
INPUT1=/sasdata/input1
export INPUT1
INPUT2=/sasdata/input2
export INPUT2
```

The following parameters were set in the libnames.parm file in the SAS software installation directory - /sas9/SAS_9.1/spds/site:

```
libname=tmp pathname=/tmp;
libname=saslib pathname=/sasdata/spdsmeta roptions="datapath=('/sasdata/output1'
'/sasdata/output2' '/sasdata/output3' '/sasdata/output4' '/sasdata/output5'
'/sasdata/output6' '/sasdata/output7' '/sasdata/output8')";
```

These parameters tell the SAS Scalable Performance Data Server:

- The name of the libname
- The location of the SAS Scalable Performance Data Server metadata directory
- The locations of output directories where the SAS Scalable Performance Data Server will stripe the output data across all defined directories

The following SAS Scalable Performance Data Server parameters were set in the spdsserv.parm file located in the SAS software installation directory - /sas9/SAS_9.1/spds/site:

```
SORTSIZE=512M;
INDEX_SORTSIZE=512M;
GRPBYROWCACHE=512M;
BINBUFSIZE=32K;
INDEX_MAXMEMORY=30M;
WORKPATH="/var/tmp";
NOCOREFILE;
SEQIOBUFMIN=64K;
```

```
RANIOBUFMIN=4K;
MAXWHTHREADS=64;
MAXSEGRATIO=75;
WHERECOSTING;
RANDOMPLACEDPF;
MINPARTSIZE=256M;
TMPDOMAIN=TMP;
MAXGENNUM=200;
SQLOPTS="reset plljoin";
```

Among the variables, this file defines:

- The workpath directory location; this area is used by SAS as a temporary space for operations such as sorting when running SAS jobs.
- The minimum partition size for output data; based on the value of this variable, output data will be striped in chunks across multiple output directories that are defined in libnames.parm.
- The maxgennum parameter; this controls the number of SAS jobs that can simultaneously be launched.

The lib.sas file in the /sasdata directory was created as it is called by SAS code when executing the SAS star schema build.

Contents of lib.sas file:

```
/* used by bulk_load.sas */
/* used by bulk_load_validate.sas */
/* used by bulk_load_spds.sas */
/* ONLY ONE SASLIB LIBNAME CAN BE UNCOMMENTED AT ONE TIME!!!! */
/* UNCOMMENT TO USE DATASETS */
/* libname saslib BASE "/sqfs1/output/sasdemo"; */
/* UNCOMMENT TO USE SAS Scalable Performance Data Server!!!!! */
libname saslib SASSPDS IP=YES LIBGEN=YES USER="ANONYMOUS"
schema="saslib" Serv="5190" HOST="localhost";
/* used by ds_copy.sas */
/* libname ds_in BASE "/sqfs1/output";
libname ds_out BASE "/sqfs1/output"; */
```

## Appendix G: Details on SAS Jobs Completion Times

Table 13 details a break down of the total completion times for the SAS star schema build using NFS. It shows the completion times for each individual step required to complete the build of the SAS star schema dimensional model. These steps are defined in Section 3, "Test Description" in this document. Typically total completion is the most important metric to the customer instead of the completion time of individual steps because the data warehouse is not available for use until all three steps are complete.

**Table 13) Completion times for each step of SAS star schema dimensional data model build using NFS.**

| | NFS | | | |
|---|---|---|---|---|
| | 4 CPUs | | 8 CPUs | |
| | Test # 1<br>Small Test Size | Test # 2<br>Medium Test Size | Test # 3<br>Small Test Size | Test # 4<br>Medium Test size |
| **Dimension Creation Times (minutes)** | 0.92 | 8.62 | 0.98 | 9.58 |
| **Order Load/Data Validation/Indexing Times (minutes)** | 40.67 | 203.87 | 24.83 | 138.65 |
| **SAS Scalable Performance Data Server Snap Times (minutes)** | 0.12 | 0.17 | 0.13 | 0.18 |
| **Total Completion Times (minutes)** | 41.71 | 212.66 | 25.94 | 148.41 |

Table 14 details a break down of the total completion times for FC. It outlines the completion times for each one of the 3 steps required to complete a full build of the SAS star schema dimensional model. Once again, typically customers would consider total completion times more important than completion times for each individual step.

**Table 14) Completion times for each step of SAS star schema dimensional data model build using FC.**

| | FC | | | |
|---|---|---|---|---|
| | 4 CPUs | | 8 CPUs | |
| | Test # 5 Small Test Size | Test # 6 Medium Test Size | Test # 7 Small Test Size | Test # 8 Medium Test size |
| **Dimension Creation Times (minutes)** | 0.63 | 5.55 | 0.7 | 7.08 |
| **Order Load/Data Validation/Indexing Times (minutes)** | 38.42 | 183.53 | 23.02 | 123.43 |
| **SAS Scalable Performance Data Server Snap Times (minutes)** | 0.05 | 0.07 | 0.05 | 0.05 |
| **Total Completion Times (minutes)** | 39.1 | 189.15 | 23.77 | 130.56 |

## Acknowledgments

Special thanks to the following for their contributions:

Network Appliance, Inc.

Lee Dorrier, Dan Morgan, Glenn Colaco, Steve Daniel,

Alliance and Workload Engineering Lab Team

SAS Institute Inc.

Nancy Wills, Patrick McDonald, Cheryl Doninger