White Paper

High-Performance Storage Solutions for Visual Simulation

Daniel Bernal, David Egts

Abstract

Over the past several years, technology has become available that has made feasible the use of large-area, geographically specific (geospecific) visual databases by high-fidelity simulations. Most of these databases are created from aerial photography or satellite imagery of the exact location being represented.

Thanks to increased automation these geospecific databases are now more rapidly created and at a cheaper cost then traditional geotypical databases. This results in increased training value for pilots, with enhanced fidelity and realism. The tradeoff is increased computer system complexity and more complex visual database management. Building on our previous work [1], this paper takes an expanded look at the benefits of using a storage area network (SAN) in a multisimulator visual simulation environment. In addition to discussing lessons we've learned since the first paper's publication, we will also discuss the results of additional in-depth performance tests.

1.0 Background

Geospecific databases are emerging as the de facto standard in high-fidelity flight simulation. In recent years high-fidelity simulation applications have incorporated both government-provided and commercially available imagery sources. Color imagery with resolutions better than one meter per pixel are commonplace in today's large area databases. Government organizations like the Air Force Research Laboratory's (AFRL) Warfighter Training Research Division (AFRL/HEA) in Mesa, Arizona (*www.williams.af.mil*) have advanced the Air Force's Distributed Mission Training (DMT) Program by conducting research exercises using large-area geospecific visual databases. The databases used represent areas larger than 10 degrees latitude by 10 degrees longitude (approximately 600 miles square). Databases of this size can easily grow to over 75GB of data, and can be several times larger when including per-texel correlated sensor simulations. Database generation for visual systems is still very much a manual process, but state-of-the-art commercial toolkits and advanced techniques have improved the process to a point at which it is faster to generate a geospecific database than a geotypical database [1]. Considering the speed at which these databases are growing and the availability of imagery, we will soon be managing databases of the multiple-terabyte size. Managing such databases, and those that are even larger, will introduce many issues.

2.0 Managing Large Databases

2.1 Local Copy of Visual Database

Typically, visual systems store a copy of the visual database on a locally attached disk. Local disks are required for performance and real-time determinism. Figure 1 illustrates how data moves through a visual system when the database is stored on a locally attached disk. Data moves from disk to system memory to the graphics pipeline. Until recently, disk drives weren't large enough to accommodate a whole database of this size on one disk. Now, keeping the entire database on one removable disk drive makes management easier, but it is more expensive and a lower-performance solution. Even with larger, higher-bandwidth disk drives available, it is preferable to use multiple drives to achieve the performance benefit of striping across multiple drives. Creating a filesystem that physically resides (striped) across multiple drives will perform better, as more drives work together to provide the requested data. As graphics pipeline texture memory gets larger, and as visual databases get richer in content due to higher-resolution imagery, the need for multiple drives to feed the image generator becomes critical. A local copy of the visual database is acceptable for non-networked, single-cockpit simulators, but larger programs that require deployment of a large database across multiple simulators present new problems.

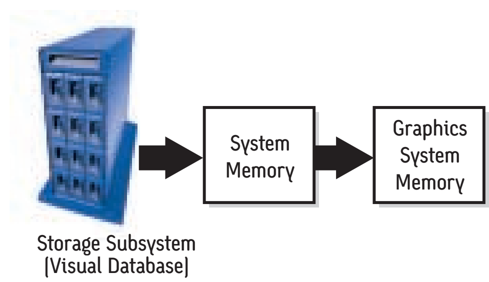

Fig. 1. Visual system data path

2.2 Visual Database Deployment

Configuration management problems grow linearly when networked, multicockpit simulators are used. Based on larger programs and considering the growth in multisimulator environments, a single filesystem shared by all simulators would keep database deployment and configuration management issues from expanding, even as more simulators are added. The task of deploying (copying/installing) databases of this size can take hours at a time. In a multisimulator environment the database must be replicated for each instantiation of the simulator in each location. Depending on local procedures for change control, this might be duplicated every time changes are incorporated to the database. As the databases get larger, the time to extract multiple copies grows tremendously. By extracting only a single copy from a single filesystem once, updates will be quicker and it will be easier to get the latest data online. Pilots will spend more time honing their skills in the simulator and less time waiting for database updates.

2.3 Database Security

Now that classified imagery is making its way into visual databases, many applications may be required to keep both classified and unclassified versions of these databases. Even if a given database doesn't contain classified information, you may be required to classify it due to the simple fact that the visual system is directly attached to a host computer that is either running classified data or is on a classified network. Managing this type of an environment is easy if the database fits on one or two disks, where the operator can simply remove the disks when the simulator is required to run unclassified. But as we've already mentioned, databases can be much larger than this. This management problem becomes even more difficult in multisimulator environments in which each image generator needs a local copy of the databases for each classification level. This condition significantly increases the security risk of someone inadvertently plugging the wrong drive into the wrong system at the wrong time, or forgetting to return a classified drive back to its secure storage vault.

Security hazards can be significantly reduced by using a single filesystem for each classification level. This way, you can adjust classification levels by simply switching the drive cables to the filesystem of the desired security level. Furthermore, similar to the data management issues mentioned in the last section, the time saved by having a single filesystem versus drive swapping enables pilots to get into the training environment faster and maximizes their training time.

3.0 Proposed Solution

Adopting rapidly evolving technologies and technologies from other domains is essential to meet the demands of sharing databases between image generators. A storage area network [3] implemented using the SGI® CXFS™ shared filesystem [4] is potentially the perfect solution for this environment. CXFS is a highly scalable, high-performance shared filesystem that allows multiple computers to share files attached via SAN. This particular solution allows multiple simulators to share one copy of a visual database that resides on a storage subsystem attached to each simulator through a SAN. The 64-bit scalability of CXFS supports file sizes as large as 9 million terabytes and filesystems as large as 18 million terabytes. In addition, CXFS supports up to 32 IRIX®, Solaris™, Windows NT®, and Windows® 2000 clients. This enables existing heterogeneous systems to share the same filesystem at local disk speeds, while performing complementary tasks such as image generation, instructor operation, brief/debrief, etc. Figure 2 illustrates how the data moves through the visual system if the visual database is attached to the visual system through a storage area network.

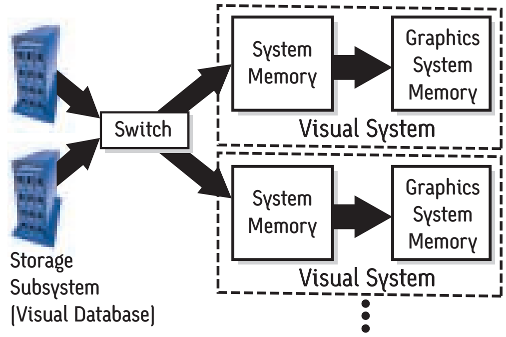

Fig. 2. Multiple visual system: data path with a SAN

3.1 Solution Background

Traditional network filesystems such as the Network File System (NFS) cannot be used because they do not have the local, deterministic disk performance required by a high-performance simulation application. As a result, a high-performance shared filesystem is necessary to meet many, if not all, of these requirements. A SAN with shared files is a high-speed, scalable fabric of servers and storage devices providing:

- Disk storage consolidation
- Data access and availability
- Centralized data management
- High, channelized bandwidth
- Shared data access
- Modular scalability

In a SAN, a Fibre Channel (FC) fabric is installed between all servers and all storage devices. This creates a storage data network, separate from the local area network (LAN).

SAN technology addresses the shortcomings of the direct-attach model by removing the LAN bottlenecks that direct-attach storage creates while maximizing availability and usage of all compute-related assets. Figure 3 shows a typical SAN.

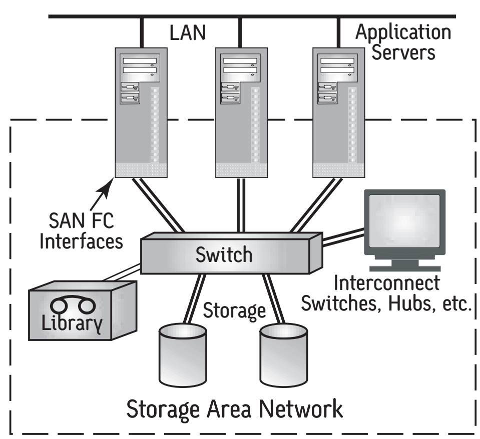

Fig. 3. Typical SAN

Shared filesystems enable data within a SAN to be available from multiple hosts. Most shared filesystems accomplish this data sharing by using protocols such as NFS. The drawback for multiple-simulator environments with more shared filesystems is the speed at which their databases would be transferred through the network-attached storage—i.e., at LAN speeds. A high-performance, shared filesystem that provides Fibre Channel transfer rates is imperative for the success of a SAN model with shared files.

As a SAN solution, CXFS solves the performance issues for multiple-simulator environments and provides the benefits of lower total cost of ownership, improved data availability, increased bandwidth, and increased interoperability. CXFS has also gained wide acceptance in the business enterprise, science, media, and national defense sectors.

3.2 The Test Environment

The initial tests in our original paper were conducted at the Distributed Mission Training Group at AFRL in Mesa, Arizona, over a period of two days. During that testing we determined that having a CXFS SAN to support multiple real-time visual simulations was very feasible. Simple tests were performed to quantify the disk subsystem bandwidth required when initializing and running the Aechelon C-NOVA™ run-time engine. Using the lessons learned from the initial tests at AFRL, we performed additional testing at the SGI Federal Reality Center™ facility in Silver Spring, Maryland. This demonstration facility is unique in that every node in the demonstration facility is connected to a CXFS SAN for day-to-day operations, running some of the most demanding graphics and I/O applications in the world. This was the perfect environment to perform experiments on CXFS within the context of visual simulation. For this exercise we used one SGI® Onyx® 3400 system in addition to two Silicon Graphics® Onyx2® systems. All three of these Onyx systems were preconfigured as clients on a SAN that included a Brocade® switch and SGI® TP9100 and Ciprico® 7000 RAID storage systems. Figure 4 shows the configuration of the SAN as used for these tests. Following are the hardware and software configurations of the systems used:

System 1 (Sim 1):
System 1 was an Onyx 3400 system configured with twelve 400 MHz MIPS® R12000® CPUs, 1.5GB memory, and two InfiniteReality3™ graphics pipes, each with two Raster Managers (RMs). Two PCI optical Fibre Channel host bus adapters provided connectivity to the SAN.

System 2 (Sim 2):
System 2 was an Onyx2 system configured with eight 250 MHz MIPS® R10000® CPUs, 2GB memory, and two InfiniteReality2™ graphics pipes, each with two Raster Managers. Two PCI Fibre Channel host bus adapters provided connectivity to the SAN.

System 3 (Sim 3):
System 3 was an Onyx2 deskside system configured with four 400 MHz MIPS R12000 CPUs, 2.5GB memory, and one InfiniteReality3 graphics pipe with two Raster Managers. Two PCI Fibre Channel host bus adapters provided connectivity to the SAN.

System 4:
System 4 was an SGI® Origin® 2200 deskside server configured with four MIPS R10000 195 MHz CPUs and 1.9GB memory. Two PCI Fibre Channel host bus adapters provided connectivity to the SAN. This system served as the CXFS metadata server—a central clearing house for metadata logging, file locking, buffer coherence, and other necessary cluster coordination functions. This system was selected as the primary metadata server to allow the SAN-attached Onyx systems to focus on image generation, as opposed to metadata management. However, if the Origin server had suffered a hardware or software failure, any of the other nodes on the CXFS cluster would have taken over as the metadata server with no noticeable loss of performance.

System 5:
System 5 was a Silicon Graphics Fuel™ workstation configured with one MIPS® R14000™ 600 MHz CPU, one V12 VPro™ graphics card, and 1GB memory. One PCI Fibre Channel host bus adapter provided connectivity to the SAN. This Silicon Graphics Fuel workstation served as a centralized point for monitoring the performance of all CXFS nodes via Performance Co-Pilot™ software tools. The workstation could have also served as an instructor operator station, threat station, or another role that used the shared data stored on the SAN, but the necessary role-playing software was not available during the time of testing.

Storage Subsystem:
The storage subsystem used for the test was configured using two SGI® Total Performance 9100 (TP9100) redundant array of independent drives (RAID) storage systems. The TP9100 storage systems were connected to a Brocade® FC switch with optical cables. A Ciprico 7000 storage system was also connected to the SAN and visible to all CXFS clients, but was not used for testing. This was done to create a worst-case environment in which a single storage system provided data to all SAN clients.

Fibre Channel Switch:
A 16-port Brocade® SilkWorm® Fibre Channel switch was used as the switching hardware to connect the

simulators to the storage volume. All clients were connected to the switch using FC optical cabling.

Software Configuration:
Each SGI Onyx system ran IRIX® 6.5.18f with CXFS to provide the shared filesystem capability. An Origin® 2000 server (system 4) was configured as the primary metadata server for the CXFS cluster. Metadata is stored information that describes a file, such as a file's name, size, location, permissions, extended attributes, etc. [4]. In addition, a Silicon Graphics Fuel workstation was used as a display station for visualizing performance metrics collected from the three systems using Performance Co-Pilot [5].

Visual Database and Run-Time Configuration:
The primary visual database used was developed by Aechelon Technology, Inc. for the AFRL Roadrunner exercise. This visual database represents an area approximately seven degrees latitude by seven degrees longitude and is approximately 37GB of data. The visual database was stored on the storage subsystem we've described. Aechelon C-NOVA [5] was installed and configured to run from each of the simulator's system disks. The secondary visual database used was developed by Lockheed Martin [7]. This database was used to validate exercises running heterogeneous run times and databases off of the same SAN simultaneously.

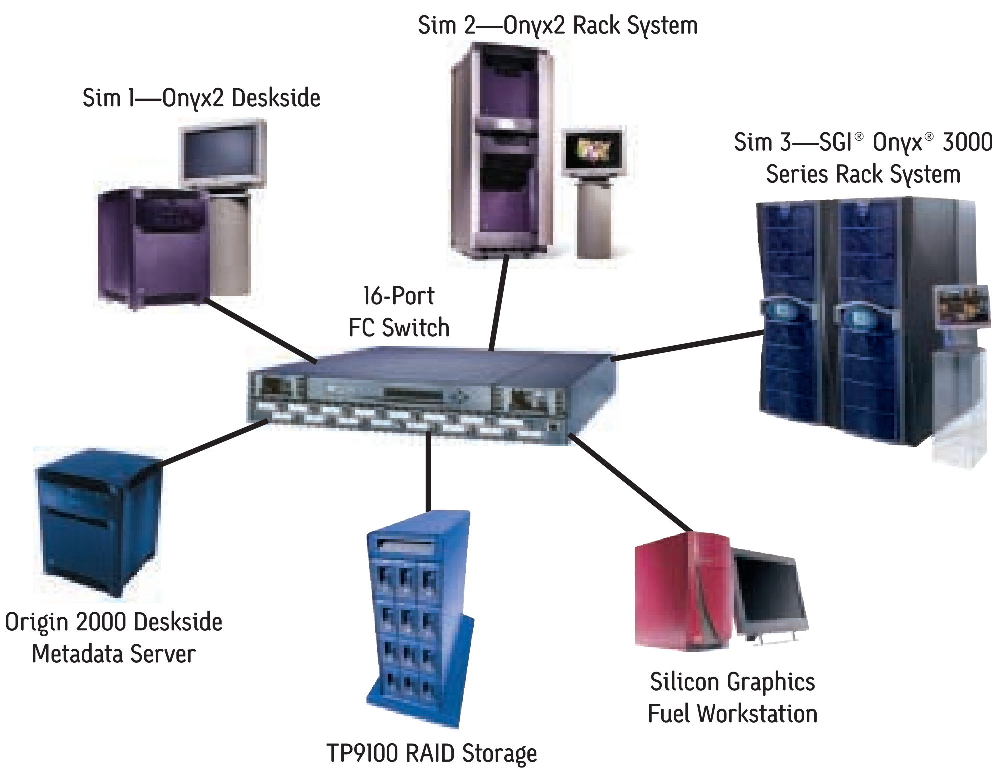

Fig. 4. SAN test configuration

4.0 Testing Results

Testing was performed to collect the following information:
- Disk read bandwidth during initialization
- Disk read bandwidth during run time
- Graphics subsystem swap buffer frame rate

To gather this information we used Performance Co-Pilot [5] during each test to gather the following metrics:
- Total bytes read (xvm.ve.read_bytes) for each graphics system in addition to the metadata server
- Total number of swapbuffer calls (gfx.all.swapbuf)—This metric represents the simulation frame rate for all graphics pipes in the system; the frame rate is totaled for multiple pipe systems (i.e., a two-pipe system with each pipe running at 60 Hz will record 120 swapbuf calls per second)

4.1 System Initialization Tests

The initialization tests were designed to test if the SAN could easily handle the bandwidth required to simultaneously initialize multiple simulators using either the same or dissimilar visual databases, all from a single shared storage subsystem.

4.1.1 Test 1: Multiple System Initialization at Same Location of Database—Offset Start

We initialized all three systems at the same time, and started systems flying a few seconds apart. Results showed a spike in access to disk by each system during initialization and at start of run. Simulation frame rate remained a steady 60 Hz during the short time the simulator remained in run mode. Figure 5 shows disk read access metrics and graphics subsystem swapbuf calls.

Fig. 5. Test 1 bytes read and frame rate

4.1.2 Test 2: Multiple System Initialization at Same Location of Database—Synchronized Start

We initialized all three systems at the same time and started systems flying at the exact same time as well. Results showed a spike in access to disk by each system during initialization and at the start of run. Simulation frame rate remained a steady 60 Hz during the short time we remained in simulator run mode. Figure 6 shows disk read access metrics and graphics subsystem swapbuf calls.

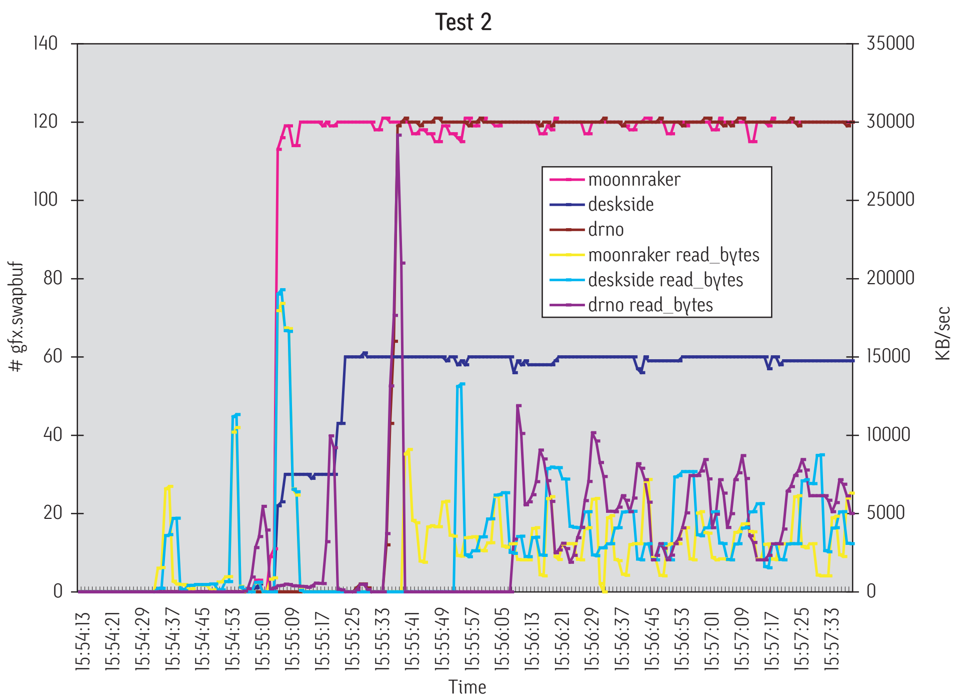

Fig. 6. Test 2 bytes read and frame rate

4.1.3 Test 3: Multiple System Initialization at Different Areas of Database—Synchronized Start

We initialized all three systems at the same time and started systems flying a few seconds apart. Results showed a spike in access to disk by each system during initialization and at the start of run. Simulation frame rate remained a steady 60 Hz during the short time we remained in simulator run mode. Figure 7 shows disk read access metrics and graphics subsystem swapbuf calls.

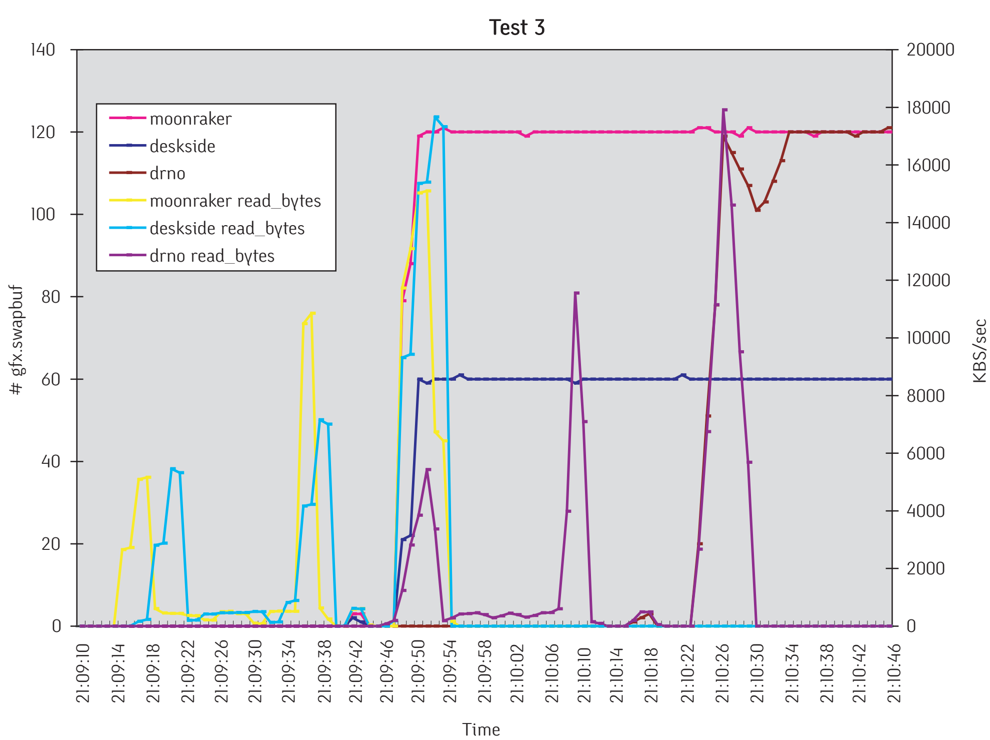

Fig. 7. Test 3 bytes read and frame rate

4.1.4 Test 4: Multiple System Initialization Using Different Visual Run Times and Databases

We initialized one system using Aechelon C-NOVA run-time software and two systems using Lockheed Martin TOPSCENE™. C-NOVA loaded the Nellis AFB visual database and the TOPSCENE system loaded a Washington, D.C., visual database. All three systems were initialized at exactly the same time, but we started them flying a few seconds apart. Results showed a spike in access to disk by each system during initialization and at the start of run. Simulation frame rate remained a steady 60 Hz during the short time we remained in simulator run mode. Figure 8 shows disk read access metrics and graphics subsystem swapbuf calls.

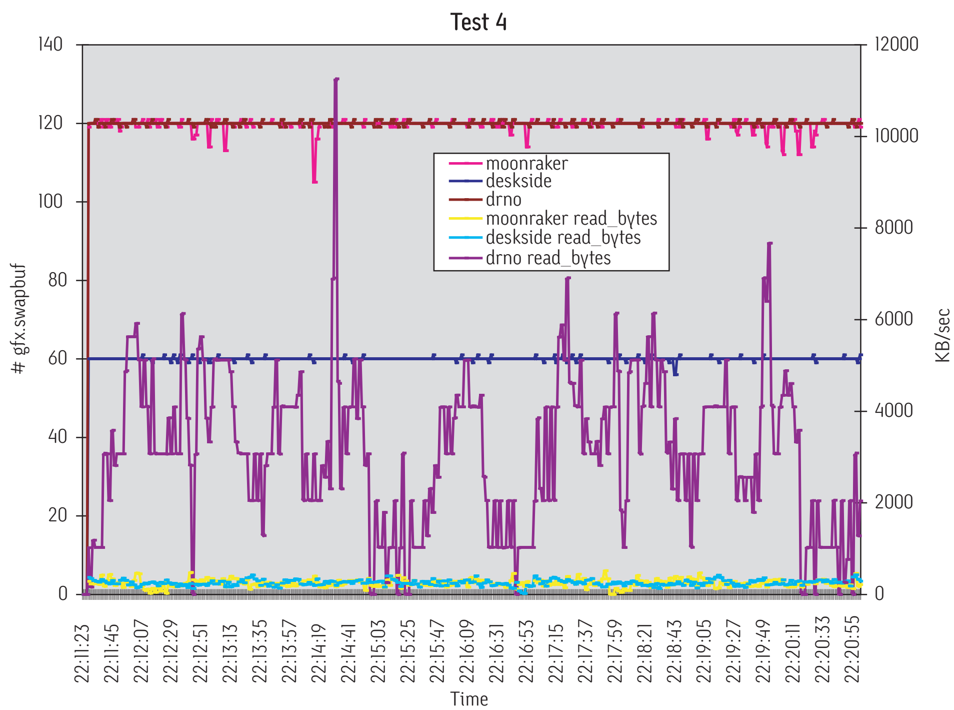

Fig. 8. Test 4 bytes read and frame rate

4.2 System Real-Time Tests

The real-time tests were designed to test whether the SAN could easily handle the real-time requirement from three individual systems/simulators running from exactly the same visual database on the same storage subsystem. All tests in this section were running C-NOVA, which loaded the Nellis AFB visual database. All simulators ran the same preprogrammed flight path in the pattern of a circle approximately 200 km.

4.2.1 Test 5: Multiple Systems Running at Same Area of Database

All three systems were set to run mode at exactly the same time, and data was collected for several minutes. This shows multiple systems accessing the same textures on a shared filesystem at the same time. Results revealed that the simulation frame rate remained a steady 60 Hz during the time that data was collected. Figure 9 shows disk read access metrics and graphics subsystem swapbuf calls.

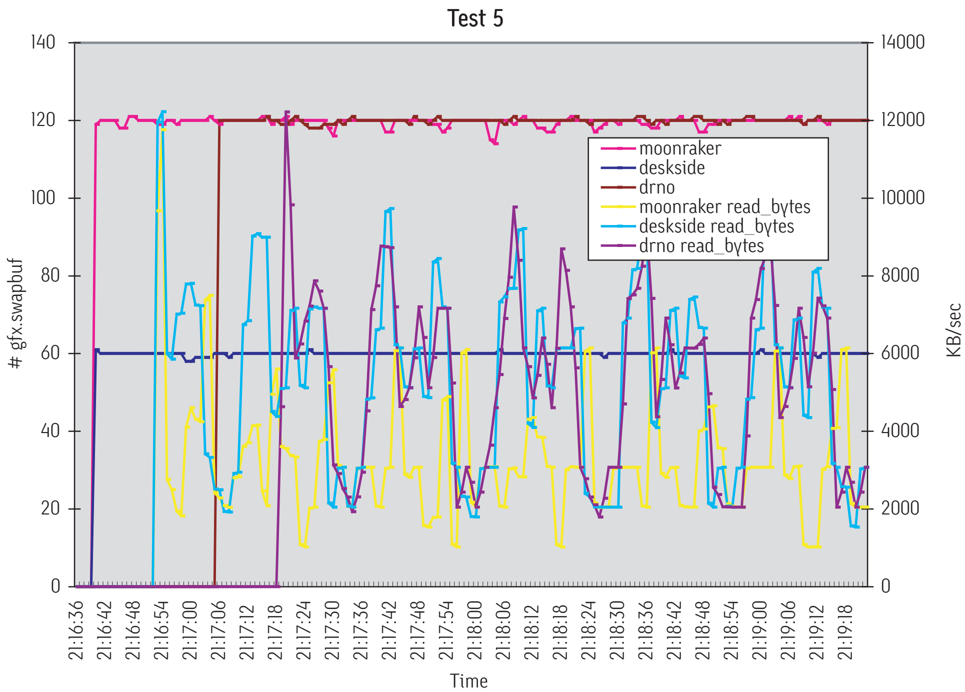

Fig. 9. Test 5 bytes read and frame rate

4.2.2 Test 6: Multiple Systems Running at Different Areas of the Database

All three systems were set to run mode at different times and data was collected for a few minutes. This test was performed to simulate multiple simulators flying simultaneous solo missions, or in a dogfight in which they access textures on the shared filesystem at the same time. Results revealed that the simulation frame rate remained a steady 60 Hz during the time that data was collected. Figure 10 shows disk read access metrics and graphics subsystem swapbuf calls.

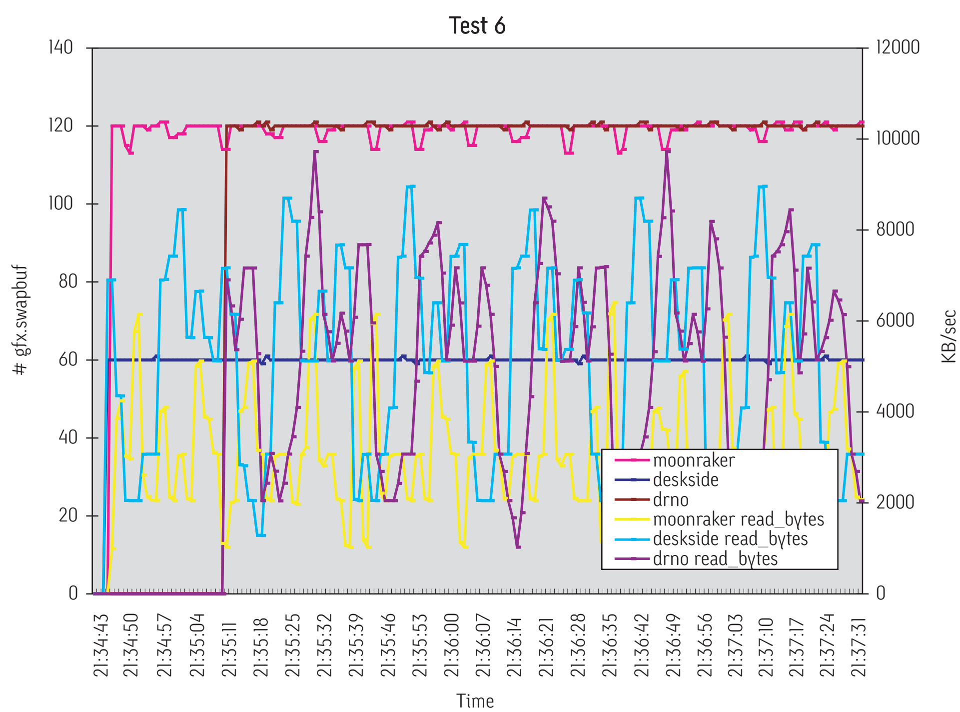

Fig. 10. Test 6 bytes read and frame rate

4.2.3 Test 7: Multiple Run Times Accessing the Same Storage Subsystem

Two TOPSCENE run times were initialized and one C-NOVA run time was set to run mode. Data was collected over a period of 10 minutes. This test was performed to simulate multiple simulators flying different types of aircraft on different training exercises in which they access different textures on the shared filesystem at the same time. Results revealed that the simulation frame rate remained a steady 60 Hz during the time that data was collected. Figure 11 shows disk read access metrics and graphics subsystem swapbuf calls.

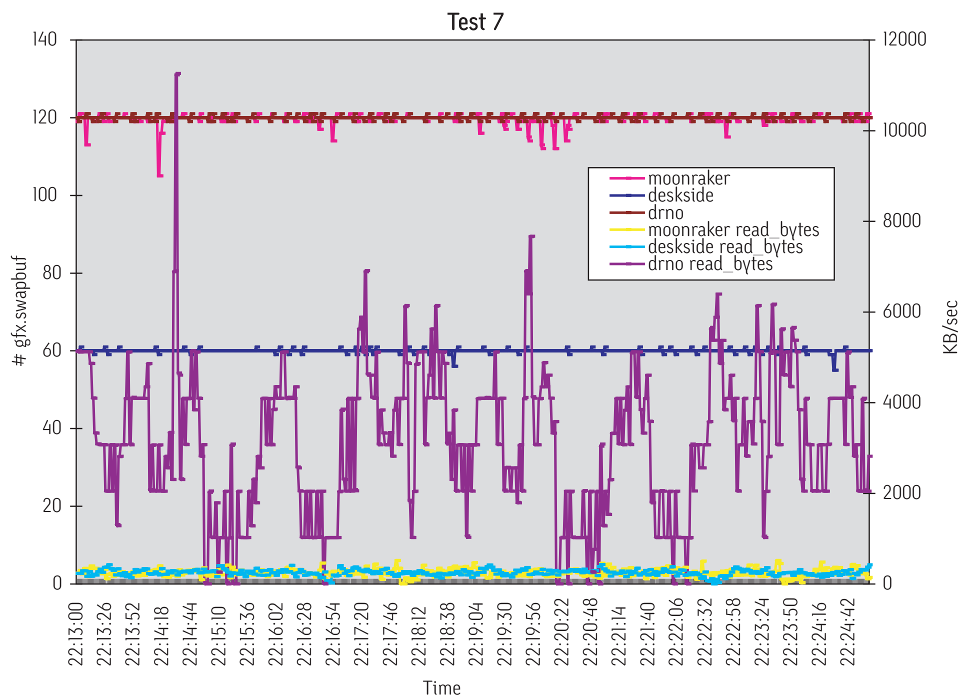

Fig. 11. Test 7 bytes read and frame rate

4.2.4 Test 8: Multiple Run Times Accessing the Same Storage Subsystem

This test is identical to test 5 but data was collected overnight. This test was performed to simulate a worst-case extended mission to test if CXFS can provide the required storage bandwidth and availability over an extended period of time. Results revealed that the simulation frame rate remained a steady 60 Hz over a period of 10 hours. Figure 12 shows disk read access metrics and graphics subsystem swapbuf calls.

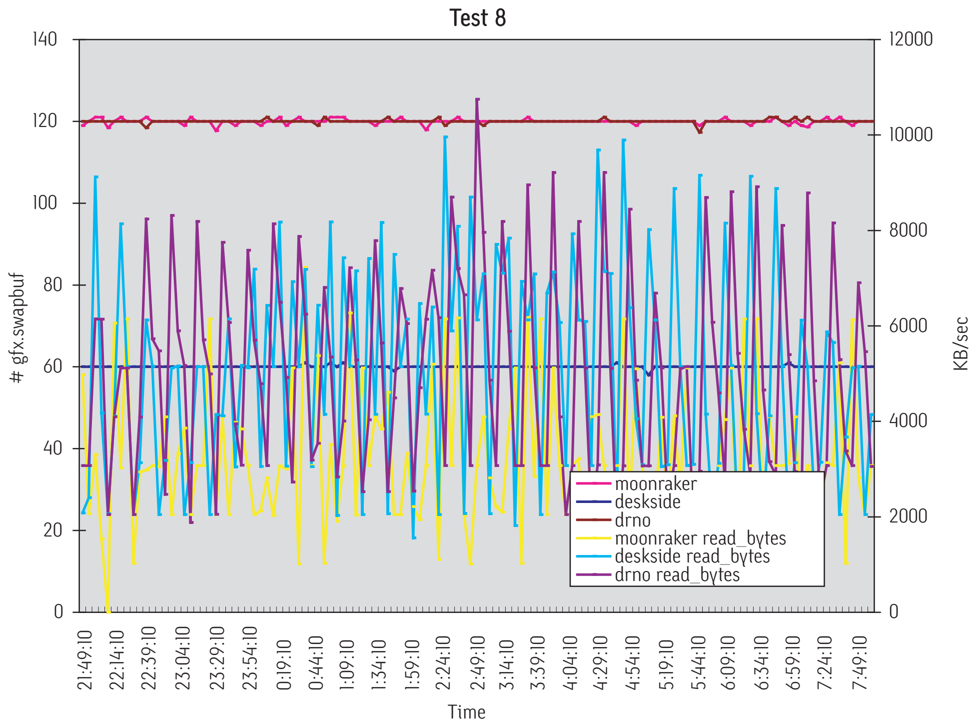

Fig. 12. Test 8 bytes read and frame rate

4.3 Analysis of Tests Performed

After running the previously discussed performance tests, we are confident that a SAN based on CXFS can provide the necessary bandwidth and availability to initialize and provide deterministic real-time data to simulators with characteristics similar to those of locally attached storage. As the number of simulators and size of databases grow, higher-performance storage systems such as striped TP9100 storage systems or single or multiple SGI® TP9400 storage systems may be necessary. Fortunately, CXFS can easily scale to operate on these larger storage systems, as proven in other domains [8,9].

As illustrated in the graphs, some swap buffer rate samples were slightly lower than 60 Hz per pipe, followed by the next samples being slightly higher than 60 Hz per pipe. This anomaly is due to the minimally invasive nature of Performance Co-Pilot. Performance Co-Pilot is designed to collect performance data by consuming as little system resources as possible so that the performance monitoring processes do not skew performance results. If samples had been collected at greater time intervals, the sampling errors would have been normalized to 60 Hz.

5.0 Future Work

Given the positive results, we will continue to explore the application of CXFS in SAN environments for visual simulation. Additional tests that must be performed include testing this environment with more simulators configured with more graphics pipes, as well as testing the simulators running an actual training exercise. With the recent introduction of InfiniteReality4™ graphics, featuring 1GB of texture memory, another area of investigation is the use of larger and higher-resolution image-specific databases. Based on the encouraging results of our testing to date, we strongly believe that CXFS will provide the needed performance to satisfy this challenge.

CXFS currently includes support for clusters up to 32 clients, with support for 64 planned. In addition to clients for IRIX, Solaris, Windows NT, and Windows 2000, CXFS clients for Linux®, Windows® XP, and other flavors of the UNIX® operating system will be available soon.

6.0 Conclusions

Our testing revealed that implementing a SAN in a multisimulator environment was overwhelmingly successful within the limits explored. We successfully initialized and ran three simultaneous-simulation applications, all accessing the same visual database on a high-performance SAN shared filesystem. No noticeable difference was seen between having the visual database locally stored and having SAN attached. The disk-read performance required by the visual system run time was well within the capabilities of a SAN. In addition, a SAN-attached visual database demonstrated performance capabilities far exceeding those required to serve three simulators. This testing, in addition to the recent availability of the higher-performance, 2Gb per second Fibre Channel technology, indicates no reason why a SAN-attached visual database could not be considered as an alternative to direct attach in a multiple-simulation environment.

A SAN-attached visual database with a high-performance shared filesystem is a viable alternative to local-attach disk for its advantage in reducing the total cost of ownership, especially in a multiple-simulator environment. Today's version of CXFS can allow commodity simulation solutions based on Windows to share the same filesystem with high-fidelity visual systems like Silicon Graphics® Onyx2® RealityMonster® running Aechelon C-NOVA or Lockheed Martin TOPSCENE.

7.0 Acknowledgements

Special thanks to James Cubeta, Scott Zurawski, and Robert Dilmore of SGI Federal Silver Spring Reality Center facility for their support, time, effort, and resources to conduct this testing.

8.0 References

[1] Daniel Bernal, David Egts, and Stephen Yost, "The Use of a Storage Area Network in Multisimulator Visual Simulation Environments," White paper, *www.sgi.com/pdfs/3276.pdf*

[2] Graham Beasley, "Geo-Specific Databases for Flight Simulators," White paper, SimTecT '93 Proceedings

[3] SGI SAN Product Datasheet, *www.sgi.com/Products/PDF/2819.pdf*

[4] "SGI CXFS: A High-Performance, Multi-OS SAN Filesystem from SGI," *www.sgi.com/Products/PDF/2691.pdf*

[5] "Performance Co-Pilot: Monitoring and Managing System-Level Performance," *www.sgi.com/software/co-pilot*

[6] C-NOVA Product Description, Aechelon Technology, Inc., *www.aechelon.com*

[7] TOPSCENE Product Description, Lockheed Martin Corporation, *www.lockheedmartin.com/factsheets/product639.html*

[8] "Supercomputer Transition at Fleet Numerical Meteorology and Oceanography Center," *www.sgi.com/pdfs/3168.pdf*

[9] "SGI at Laboratoire Éclair: Going for Speed with CXFS," *www.sgi.com/pdfs/3179.pdf*

Corporate Office
1600 Amphitheatre Pkwy.
Mountain View, CA 94043
(650) 960-1980
www.sgi.com

North America 1(800) 800-7441
Latin America (52) 5267-1387
Europe (44) 118.925.75.00
Japan (81) 3.5488.1811
Asia Pacific (65) 771.0290

3196 [11/22/2002] J14160