

Execution-driven Simulation of Network Storage Systems

Yijian Wang and David Kaeli

Department of Electrical and Computer Engineering
Northeastern University
Boston, MA 02115
yiwang, kaeli@ece.neu.edu

Abstract

A number of new network storage architectures have emerged recently that provide shared, adaptable and high-performance storage systems for data-intensive applications. Three common storage networking architectures are Direct-Attached Storage (DAS), Network-Attached Storage (NAS), and Storage Area Network (SAN). Efficient implementations of each of these classes of storage architecture can have a significant impact on overall system performance.

To be able to tune both the performance of a network storage architecture and its underlying workload, an accurate simulation modeling environment can be very valuable. In this paper we present ParIOSim, a validated execution-driven simulator for network storage systems. This simulator can be used to accurately predict the performance of parallel I/O applications as a function of the underlying storage architecture. ParIOSim also provides a flexible simulation environment to guide system level storage optimizations. To evaluate the accuracy of our simulation environment, we compare the performance of ParIOSim to the performance obtained on an actual parallel system. To demonstrate the utility of ParIOSim, we report on the overall system performance obtained for a parallel I/O benchmark application as run on different storage architectures.

1 Introduction

As the performance gap between processors and disk subsystems continues to grow, storage system performance has become a limiting factor in the system performance of I/O intensive applications. Many

applications, such as multimedia servers, database systems and scientific simulations, require extensive I/O access. Given the growing importance of these applications, I/O performance throughput has begun to draw increased attention. New network storage architectures, such as NAS and SAN, have emerged in recent years to deliver high I/O bandwidth, flexible connections and large storage capacities [3]. Figure 1 shows three typical processor-to-storage architectures. For Direct Attached Storage (DAS), storage is directly attached to the computer processor by a dedicated link. This architecture is commonly found on today's PCs and workstations. For Network Attached Storage (NAS), the storage subsystem is attached to a network of servers and file requests are passed through a parallel file system to the centralized storage device. A Storage Area Network (SAN) uses a dedicated network to provide an any-to-any connection between processors and storage devices. The connection media in current SAN systems is usually FiberChannel or iSCSI [15]. The main benefit of using a SAN over a NAS is that I/O traffic is offloaded from a LAN/WAN to a dedicated high-speed storage sub-network.

In this paper, we present our recent work on network I/O acceleration and parallel I/O modeling. By carefully studying disk drive characteristics and network storage performance, we are able to accurately simulate the performance of parallel I/O applications as a function a specific storage architecture. ParIOSim also provides a flexible environment to guide storage system optimizations at the application level.

The remainder of this paper is organized as follows. Section 2 will present an overview of our execution-driven simulator ParIOSim. Section 3 will provide model validation results for a range of workloads run on both real hardware and our simulation framework, simulating multiple storage architectures. Section 4

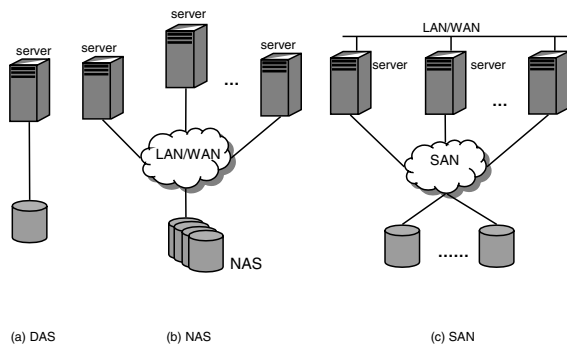


Figure 1. Topology of Direct Attached Storage (DAS), Network Attached Storage (NAS) and Storage Area Network (SAN).

will analyze how different storage architectures can impact application performance, and Section 5 will summarize this paper, discussing avenues for future work.

2 Execution-driven I/O Simulation

Our previous study on I/O characteristics [16] found that the overall performance of an I/O intensive application depends heavily on understanding file access patterns and mapping this to the underlying storage architecture. Because of the growing performance gap between processor/memory and storage systems, an efficient storage architecture is key in order to obtain good overall application performance. It becomes critical to be able to accurately predict the performance of a range of storage architectures for varying workloads.

While a significant body of work has been done to model the performance of processor and memory subsystems, only a few simulation frameworks have been developed in the area of high performance I/O. Many disk performance models and disk simulators have been developed, capturing the dynamics of an I/O at different system levels [2, 4, 6, 8, 12, 14]. DiskSim [6] is an accurate, highly configurable, disk drive simulator. DiskSim relies upon using detailed disk characteristics. DiskSim can be used as a stand-alone subsystem simulator, or as an element in a complete system-level simulation. In [12], Baragoda et al. describe a

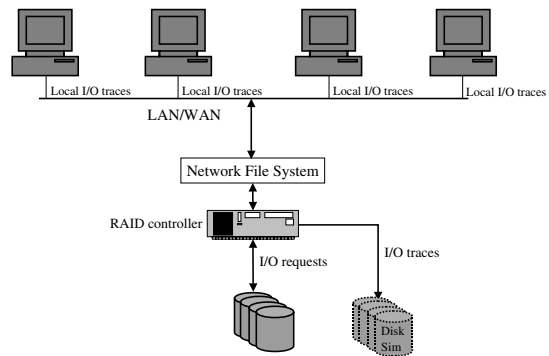


Figure 2. Execution-driven NAS simulation.

system-level simulator to predict performance of MPI-IO applications. Their focus is to study performance as a function of different collective I/O operations and file system caching algorithms. In [14], an abstract simulation model for heterogeneous RAID devices is described. To the best of our knowledge, none of these prior system-level storage simulators have been validated against multiple network storage architectures.

In this section, we present an overview of our execution-driven network storage system simulator ParIOSim. Figures 2 and 3 show models for NAS and SAN architectures, respectively. In Figure 2, a centralized network-attached RAID device is attached to a LAN/WAN of servers. All servers within the network can access this device with the support of a parallel file system. I/O requests from each server are sent to the centralized RAID device.

Figure 3 shows one variety of SAN which we will refer to for the remainder of this paper as *SAN-direct*, where disks are distributed across the network and each server is directly connected to a single disk. In this environment, we utilize I/O profiling to distribute portions of files to disks close to the processing nodes where they will be referenced (a profile run is used to guide file accesses).

In our ParIOSim framework, we model client processes and file server processes. For each I/O request associated with a client process, we execute the real parallel I/O access. In parallel to the physical I/O, we calculate the simulated I/O response time by capturing a dynamic I/O trace and sending it to the file server process, where physical disk operations are simulated. The dynamic I/O trace includes both filesystem-

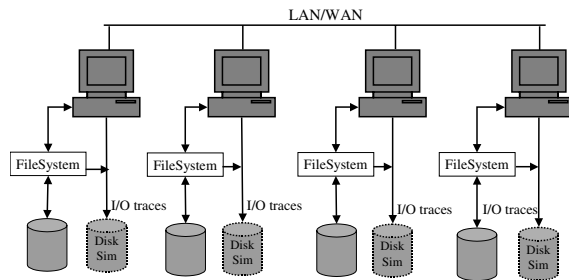


Figure 3. Execution-driven SAN-direct simulation.

tem meta-data (e.g. block size, logical block numbers, etc.) and the logical file access address (e.g. file offset, chunk sizes, read/write, etc.) We use DiskSim [6] as the underlying disk drive simulator. Techniques on how to automatically extract detailed disk characteristics can be found in [1, 10].

In our framework, processor computation and network communications are modeled by direct execution. Currently, we can only simulate the maximum number of nodes on the system. In the next section, we present a short validation study, comparing modeled and actual performance numbers.

3 Modeling Framework Validation

To begin to validate our approach to I/O performance modeling, we simulated a single node DAS architecture and compared it to measurements made on a single personal computer. The storage specification of this machine is described in Table 1.

We use a synthetic I/O workload to drive ParIOSim. In this workload, the processor generates a large number (1000) of I/O requests, both contiguous and non-contiguous, and sends them to a single disk device. We model a range of data chunk sizes (1, 2, 4, 8 and 16 disk blocks) and use difference seek distances (1, 2, 4, 8, 16 and 32 disk blocks). During workload execution time, our simulator captures the dynamic I/O trace and file system metadata, and uses them to drive DiskSim to predict I/O response times. Figure 4 shows modeled and actual performance, compar-

System	Linux RedHat 9
Processor	Intel XEON 2.0GHz
Memory Size	1GB
Number of Disks	1
Disk Drive	Western Digital WDC
Disk Model	WD800BB-75CAA0 (IDE)
Capacity	80GB
RPM	7200RPM
File System Block Size	4096B

Table 1. Hardware specifics of the PC system used in DAS Simulation.

ing I/O response times (measured in seconds) for each system. We show results for both reads and writes, varying both the size of the chunk accessed, as well as the seek distance. For all the measurements, the difference between modeled and real system response time differs by less than 3%.

3.1 Network Storage Simulation

Next, we compare two different network storage architectures running real (versus synthetic) workloads.

3.1.1 Network Attached Storage Simulation

To both perform physical measurements, and also to run our parallel simulation environment, we use a Beowulf cluster [7]. The specifications for this cluster can be found in Table 2. The Beowulf Cluster has 32 nodes (though we only ever use 25 maximum); each node has a local 8.4GB IDE disk and there is a shared SCSI RAID device directly attached to one node. For our NAS environment, all I/Os are directed to this shared RAID device. The first benchmark we used in this work is the NPB2.4/BT benchmark from the NAS Parallel Benchmark Suite [11] version 2.4. The application that we are using is the Block-Tridiagonal(BT), which is file-bound processing. The program dynamically generates a data file (1.5 GB) and then reads it back. Each parallel process periodically performs a sequential write, and the updated chunk is later read back. Three parallel I/O schemes are studied with this benchmark.

We use MPI-2 [9] Collective I/O, also called two phase I/O, as our I/O middleware. MPI-2 Collective I/O has been shown to provide high performance parallel I/O in numerous parallel I/O studies [5, 13, 14]. During simulation, each client process captures its own local I/O trace and sends it over the network

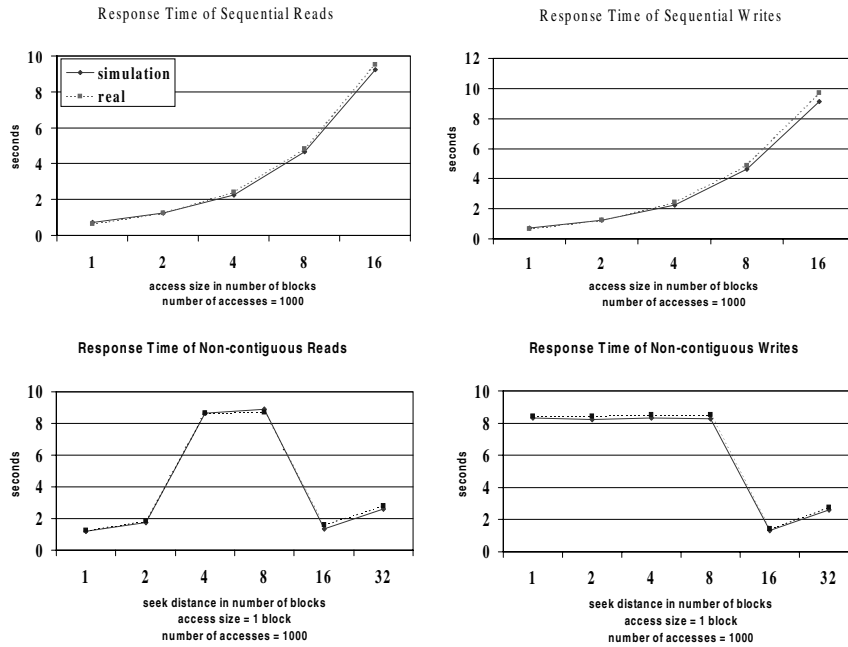


Figure 4. Validation results of DAS.

to the file server process; the file server process processes each I/O request in the model and returns the requested data back to the client processes. Our modeling framework captures the file server disk latency as well as the network overhead (we have found that the added overhead introduced by ParIOSim does not affect the accuracy of our model). Figure 5 reports the comparison between simulation results and real system performance of NPB2.4/BT.

3.1.2 SAN-direct Simulation

The local IDE disks across this Beowulf Cluster are set up as SAN-direct devices. Each node is a file server for its local disk. We also implemented file access profiling. The profiler optimizes data layout and distributes each I/O request in the network to its corresponding file server. More details about the design of our profiling system can be found at [16]. We use the same NPB2.4/BT benchmark by redirecting each I/O request to its local disk drive. Simulation latency includes local disk response time, network delay, and the distributor overhead. Simulation results and real system performance are compared in Figure 6. As we

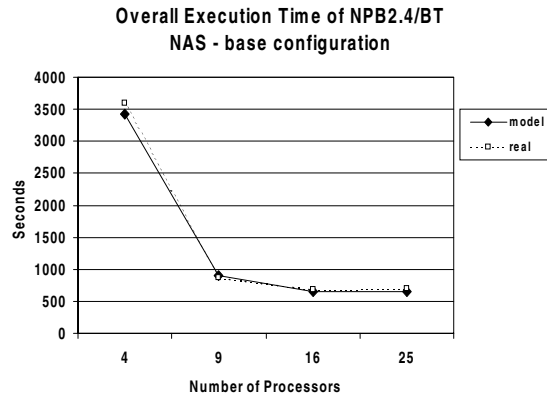


Figure 5. Execution time of NPB2.4/BT on the NAS.

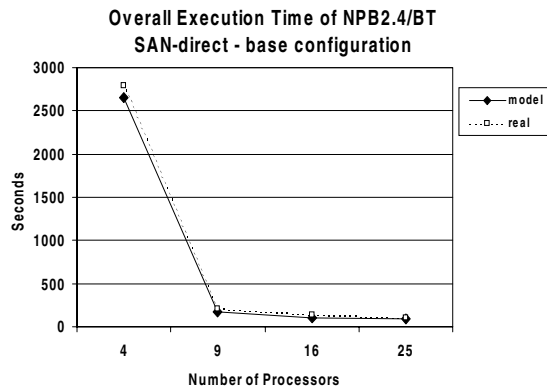


Figure 6. Results for Overall Execution time of NPB2.4/BT on SAN-direct.

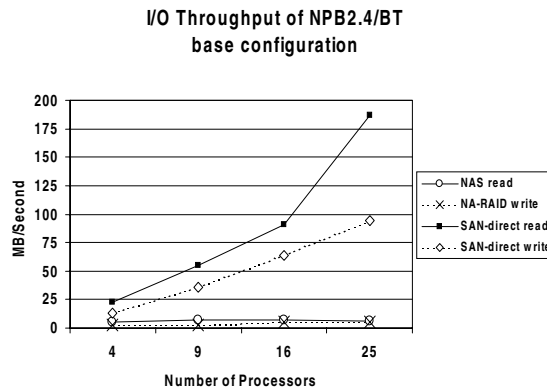


Figure 7. Results for I/O throughput of NPB2.4/BT on NAS and SAN-direct.

can see, the differences between our model and the actual system are very small. We have also obtained similar results for other workloads.

4 Performance Analysis

Having presented a brief validation of our parallel I/O simulator, we next investigate the impact of different storage architectures on the performance of the NPB2.4/BT benchmark. In Figure 7, we compare the I/O bandwidth of NAS and SAN-direct. The SAN-direct achieves a speedup factor of up to 32.8X on reads and 17.8X on writes versus NAS. Meanwhile, network RAID yields a scalable speedup as the number of processors increases. The disadvantage of the NAS architecture is that the centralized RAID device has serialized the I/O accesses over the network, and given the amount of data transferred, causes the network to become of a major bottleneck. The main benefits of SAN-direct are that it offloads the I/O workloads in the process network to a storage area network; it has improved data locality by redirecting I/O accesses to local disks; and it also improves I/O parallelism by creating multiple processor-to-disk connections.

4.1 Extending the Model

Next, we extend our model to consider the impact of moving to more current storage devices (i.e., a high performance cached fibrechannel disk device). Table 3

Disk Drive	Seagate Cheetah X15
Disk Model	ST-336752FC
Capacity	40GB
RPM	15,000
Cache Size	8MB
Interface	Fibrechannel
Transfer Rate	848Mbits/sec

Table 3. Hardware specifics of the Seagate Cheetah disk used in our simulation.

provides the specifics of the simulated disk drive under consideration.

We again utilize the NPB2.4/BT benchmark and also consider an I/O workload from the SPEC_{hpc96} benchmark suite. The SPEC_{seis96.1.2} benchmark is one of three applications in the SPEC_{hpc96} benchmark suite. The application performs seismic data processing. The code consists of four phases. We only study the first two phases of this benchmark. During phase 1, the program dynamically generates a dataset (1.6 gigabytes) and during phase 2 it reads the dataset back. Each process writes 96KB chunks, and each process then reads back 2KB chunks (reads are from non-local disks.) Note that the number of processors (and disks) are different between NPB2.4/BT and SPEC_{seis96.1.2}. The reason is that the NPB benchmark requires us to use a perfect square for the number of processors. For the SPEC_{seis} benchmark, we attempt just to double the number of processors (though we don't have 32 nodes to run on so we use a maximum

Number of nodes	32 - (27 standard nodes, 4 RAID device host nodes and 1 SMP node)
Processor Type	Intel Pentium II 350 (standard nodes and RAID nodes) Intel Pentium II Xeon 450 (SMP nodes)
Memory	256MB SDRAM, PC100, ECC, (standard nodes and RAID nodes) 2GB (SMP node)
Disk adapters IDE SCSI	Onboard Intel PCI (PIIX4) dual ultra DMA/33 UltraWide SCSI
RAID device	Morstor TF200 with 6-9GB Seagate SCSI disks, 7200rpm, QLogic 64-bit PCI-Fibre Channel Adapter
RAID level	5
RAID capacity	36GB usable, one hot spare
IDE disk	IBM UltraATA DTTA-350840, 8.4GB, 5400rpm
File system	NFS 3
Network NIC	10/100 Ethernet Cisco Catalyst 2924 Switch Intel 82558 10/100Mb

Table 2. Hardware specifics of the Beowulf Cluster used in Network Storage Simulation.

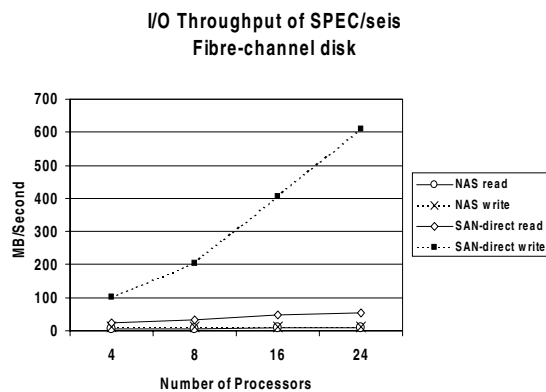


Figure 8. Results for I/O throughput of SPEC/seis for NAS and SAN-direct, using fibre-channel disks.

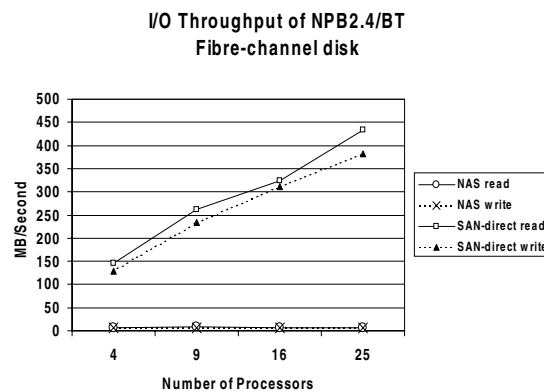


Figure 9. Results for I/O throughput of NPB2.4/BT for NAS and SAN-direct, using fibre-channel disks.

of 24 processors).

If we compare the I/O throughput obtained in Figure 9 with the results presented in Figure 7, we see that SAN-direct throughput is vastly improved (a 1054% and 1678% increase in bandwidth for 4 processors for reads and writes, respectively) with the introduction of the fibre-channel device. The NAS device enjoys a more moderate improvement (though for 4 processors, NAS obtains a 30% and 170% increase in throughput for reads and writes, respectively). Again,

SAN-direct is able to scale much better for writes because of its ability to provide parallel streams of data. But SAN-direct reads suffer from contention in the Ethernet network. Since all processors need to read non-local files in SPEC/seis, we see the impact of this in the poor scalability of reads in the SAN-direct architecture. To remedy this we introduce a SAN-all-to-all configuration, where all nodes have a direct connection to each disk.

In Figure 10 we show the throughput for

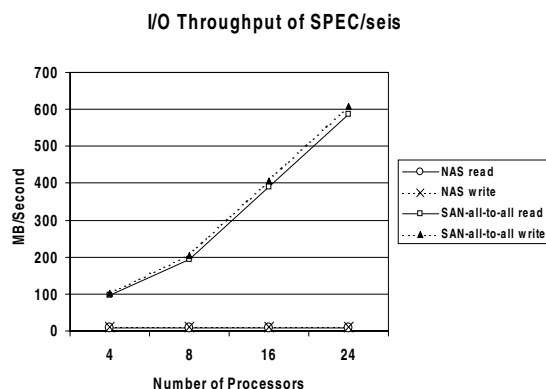


Figure 10. Results for I/O throughput of SPEC/seis for NAS and SAN-all-to-all, using fibre-channel disks.

SPEC/seis, and compare NAS and SAN-all-to-all. As we can see, the all-to-all connectivity provides much better read throughput versus the SAN-direct results. While the read performance is vastly improved, we will see later that the overall application throughput is not significantly affected.

In Figures 11 and 12 we show how performance scales in NPB2.4/BT and SPEC/seis when we use fibre-channel disks. We see that for NPB2.4/BT the performance increases dramatically when we move from 4 to 9 processors (and disks) for both SAN-direct and NAS. NPB2.4/BT is more I/O intensive than the SPEC/seis. In the NPB2.4/BT benchmark, the program spends more than 50% time servicing I/O, whereas SPEC/seis spends less than 50% of its time on I/O. In the NPB2.4/BT benchmark, the chunk size is a function of the number of processors. When the number of processors increases, the chunk size decreases, so less data is transferred over the network on each disk access. Therefore, there is little benefit obtained from increasing the number of nodes (past 9), as we see in our results.

For SPEC we see continual, though diminishing, speedup as we add processors. For the all-to-all configuration, we see that this configuration can slightly outperform the SAN-direct architecture. This slight increase in performance for the all-to-all configuration is due the decrease in contention in the 100 Mb Ethernet interconnect. For the SPEC benchmark, the I/O chunk size does not change when the number of pro-

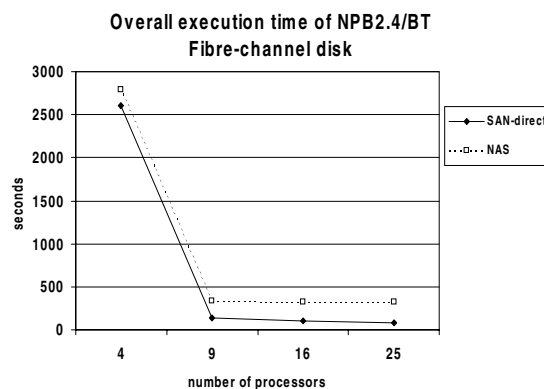


Figure 11. Execution time of NPB2.4/BT for NAS and SAN-direct, using fibre-channel disks.

cessors increases. Thus, the entire I/O work to be done can be processed more effectively when we split it across many processors (and disks) as we do in the SAN configurations.

5 Acknowledgements

This work was supported by CenSSIS, the Center for Subsurface Sensing and Imaging Systems, under the Engineering Research Centers Program of the NSF (Award Number EEC-9986821), by the Institute of Complex Scientific Software at Northeastern University, and by the NSF Major Research Instrumentation Program (Award Number MRI-9871022).

6 Summary and Future Work

In this paper we have presented ParIOSim, an execution-driven parallel I/O simulator. We have compared simulator accuracy against measurements made on two hardware platforms, a single node machine and a parallel machine. We performed this study using both synthetic and system-level I/O benchmark programs and found the results to be in agreement.

We studied the I/O response time of the NPB2.4/BT and SPEC benchmarks and found that the SAN-direct device can yield significantly improved performance and provide increased I/O throughput as

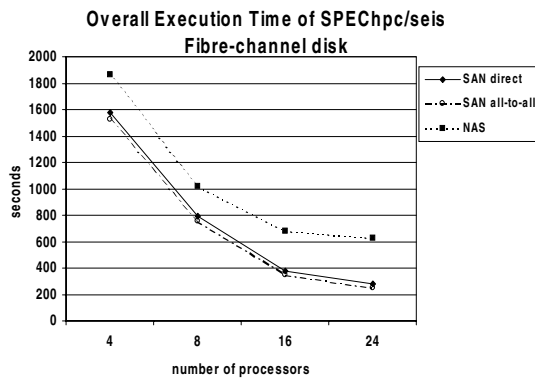


Figure 12. Execution time of SPEC/seis for NAS and SAN-direct, and SAN all-to-all, using fibre-channel disks.

compared to a NAS device. We showed the flexibility of our modeling framework by replacing the SCSI and IDE disk devices with fibre-channel disks, and produce throughput and scalability results. The main contribution of this work is the release of a simulation environment for users to test and evaluate different storage architectures and applications.

ParIOSim can presently only simulate the maximum number of nodes on the system. In future work, we will consider modeling networks as queuing systems, in order to be able to simulate much larger system configurations. We will also investigate more sophisticated network storage architectures, including a range of interconnect topologies and high-speed connection media based on fiber-channel and iSCSI.

References

- [1] B. Worthington, G. Ganger, Y. Patt and J. Wilkes. On-Line Extraction of SCSI Disk Drive Parameters. In *Proceedings of the ACM Sigmetrics Conference*, 1995.
- [2] D. Kotz, S. Toh and S. Radhakrishnan. A Detailed Simulation Model of the HP-97560 Disk Drive. Technical report, Dartmouth College, 1994.
- [3] David Sacks. Demystifying Storage Networking. Technical report, IBM, 2001.
- [4] F. Sorenson, E. Sorenson, J. Flanagan and H. Zhou. A System-Assisted Disk I/O Simulation Technique. 1999.

- [5] G. Memik, K. Kandemir and A. Choudhary. Design and Evaluation of a Compiler-directed Collective I/O Technique. In *Proceedings of the 6th Annual EuroPar Conference*, 2000.
- [6] J. Bucy and G. Ganger. The DiskSim Simulation Environment. Technical report, Carnegie Mellon University, 2003.
- [7] The Joulian Cluster at Northeastern University. <http://joulian.hpcl.neu.edu>.
- [8] K. Hwang, H. Jin and R. Ho. RAID-x: A New Distributed Disk Array for I/O-Centric Cluster Computing. In *Proceedings of the IEEE International Symposium on High Performance Distributed Computing*, August 2001.
- [9] Message Passing Interface Forum. <http://www.mpi-forum.org/>.
- [10] N. Talagala, R. Arpaci-Dusseau and D. Patterson. Microbenchmark-based Extraction of Local and Global Disk Characteristics. Technical report, University of California, Berkeley, 1999.
- [11] NAS Parallel Benchmark Suite. <http://www.nas.nasa.gov/Software/NPB>.
- [12] R. Bagrodia, S. Doco and A. Kahn. Parallel Simulation of Parallel File Systems and I/O Programs. In *Proceedings of the ACM/IEEE Supercomputing Conference*, 1997.
- [13] R. Thakur, W. Gropp and E. Lusk. Data Sieving and Collective I/O in ROMIO. In *Proceedings of the 7th Symposium on Frontiers of Massively Parallel Computation*, 1999.
- [14] T. Cortes and J. Labarta. Hraid: a flexible storage-system simulator. In *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications*, 1999.
- [15] X. He, P. Beedanagari and D. Zhou. Performance Evaluation of Distributed iSCSI RAID. In *Proceedings of the International Workshop on Storage Network Architecture and Parallel I/Os*, 2003.
- [16] Y. Wang and D. Kaeli. Profile-guided I/O Partitioning. In *Proceedings of the 17th ACM International Conference on Supercomputing*, 2003.