

Deploying Celerra MPFSi in High-Performance Computing Environments

Best Practices Planning

Abstract

This paper describes the EMC® Celerra® MPFSi approach to scalable high-performance data sharing, with a detailed discussion of the MPFSi Client Intercept Driver and performance measurements demonstrating that MPFSi is an effective means of scaling client bandwidth to an EMC CLARiiON® or EMC Symmetrix® storage array infrastructure.

June 2006

Copyright © 2006 EMC Corporation. All rights reserved.

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED “AS IS.” EMC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

For the most up-to-date listing of EMC product names, see EMC Corporation Trademarks on EMC.com.

All other trademarks used herein are the property of their respective owners.

Part number H2237

Table of Contents

Executive summary	4
Introduction	4
MPFSi benefits.....	5
Audience	5
The storage demands of high-performance computing	6
The challenge of sharing data.....	6
Client I/O performance scaling.....	6
Data sharing with MPFSi	7
Parallel data access	7
MPFSi components.....	7
The MPFSi agent.....	8
The Celerra file server	11
Storage arrays	12
Connectivity.....	12
MPFSi performance measurements for Linux clients	13
Summary discussion regarding performance measurements	19
Conclusion	19
Bibliography	19

Executive summary

High-performance computing (HPC) applications often require large files to be read from or written to compute nodes concurrently. While each individual node may operate only on a small piece of the large file, maintaining data integrity requires tight file-locking for consistency. This introduces application delays as nodes wait for I/O requests of peer nodes to complete for the lock to be removed. For large file sharing, it would be desirable to partition the data set into smaller cells that could be locked individually and permit more parallel access by nodes. But in many applications, dividing the entire data set or a single large file into cells might not be possible, due to the continuous nature of the computing process.

This is particularly true in cases when the results of one computation phase are needed as input to the next. The analysis of geologic information in the oil and gas industry, for example, typically solves some form of continuous wave equation based on a terabyte-size seismic dataset, generating 3-D visualizations to detect possible holes in the structure in a second computation phase. The data must be written very quickly at one phase to enable the use of the same nodes for the next phase. In this environment, there are multiple concurrent writers, each manipulating a relative small number of blocks. Locking the file forces some writers to wait, reducing parallelism and increasing significantly the time required for the analysis. Finer-grained locking at the file block level is key to achieving desired performance levels.

This paper describes how EMC® Celerra® MPFSi technology accelerates NFS file sharing among high-performance computing nodes by use of a client intercept driver that enables high speed, low latency iSCSI topology to be used for data delivery. The MPFSi parallel data access architecture allows hundreds to thousands of HPC nodes to share files at speeds limited only by the EMC CLARiiON® or EMC Symmetrix® storage array configuration and channel topology.

The Celerra file server gateway acts as a metadata server and communicates to node-resident MPFSi agents by using the file mapping protocol (FMP) to facilitate each node's direct channel access to a highly scalable, high-performance storage system while using standard NFS file access semantics. The MPFSi agent software uses the Linux node's native iSCSI initiator as a direct connection to the storage SAN, reducing latency in bidirectional data delivery between node and storage, and facilitating as much as three to four times the bandwidth of conventional NFS.

Introduction

EMC Celerra MPFSi (Multi-Path File System) is a combination of patented technology and the NFS protocol that enables file sharing by hundreds to thousands of client nodes while realizing up to four times the aggregate bandwidth as compared to conventional NFS file serving. MPFSi accomplishes this without requiring any application changes. A client-resident MPFSi agent interacts with the Celerra file server through a special file mapping protocol (FMP) to split the file content data flow ("Data" in Figure 1) from the NFS metadata flow ("Control"), permitting file content data to move directly between HPC nodes (MPFSi clients) and EMC storage arrays by using an iSCSI link.

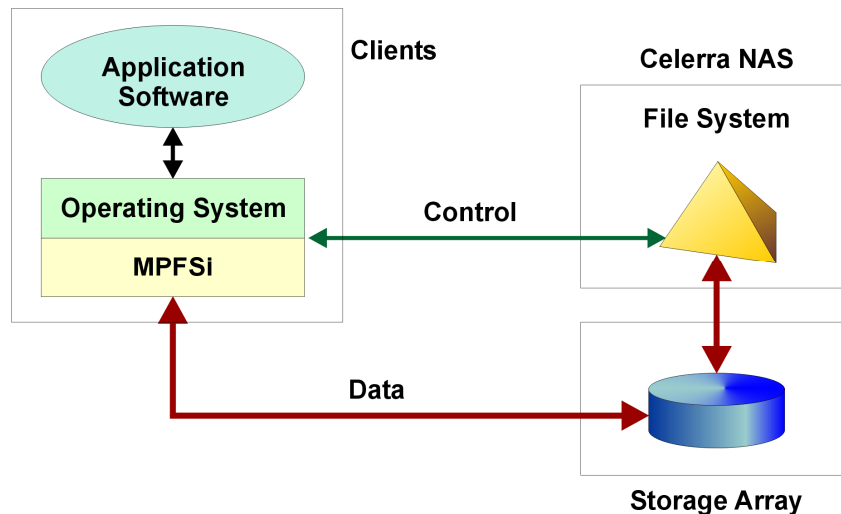


Figure 1. Simplified MPFSi topology

All of the file system operations, including block allocations, file locking, and metadata operation and logging, are performed by the Celerra server. But actual data movement of file contents occurs directly between the MPFSi client and the storage arrays without the involvement of the Celerra file server. By directing data movement to the low latency storage channel, a client realizes line speed data delivery, enabling higher bandwidth than conventional NFS would support. The MPFSi SAN data delivery also dramatically reduces the Celerra file server's workload as it is not moving data to and from the clients, permitting 10 times more clients to share the namespace of a single, file server blade. MPFSi is a very attractive shared file storage solution for HPC environments.

MPFSi benefits

By separating file system operations from data delivery, and by supporting direct parallel access to the storage devices, MPFSi offers:

- Use of NFS standards, making the MPFSi benefits available without changing client applications
- Ability to serve data to large numbers of clients limited only by the storage array bandwidth
- Ability to leverage high-performance block access caching built into the client operating environment
- Sophisticated distributed locking, enabling efficient concurrent access by multiple clients
- The extensive, proven capabilities and performance optimizations of Celerra NAS servers
- Ability to exploit the prefetching and caching features of CLARiiON or Symmetrix storage arrays
- Use of all security features of today's NAS environments
- Sharing of files for all NFS clients, with or without the MPFSi agent

Audience

This white paper is intended for HPC users (for example, architects, system or storage administrators) who have deployed or are contemplating deployment of a large scale, grid-computing environment using NFS to enable file sharing among the grid nodes, and who seek the highest possible data I/O bandwidth to maximize compute throughput.

The storage demands of high-performance computing

The challenge of sharing data

HPC initiatives are driving the deployment of new storage architectures. HPC performance is delivered by a federation of hundreds to thousands of commodity Linux blades, revolutionizing scientific, manufacturing, and business computing. While the computation problem is solved in a cost-effective manner by “grid” computing topologies, there remains the challenge of accessing the same data set by all of the compute nodes simultaneously. Scalability of the storage sharing is critical as the number of blades can increase dynamically, driven by the needs for new types of applications.

Storage access topology will have a direct impact on overall performance and resource utilization. Historic approaches to supplying data to HPC nodes involved system administrators performing manual data loading into the compute nodes. This effectively wasted valuable CPU resources, as the nodes were left idle during slow data access. In addition, there was the difficult task of dividing the data set into separate data cells to be processed by each cluster node independently.

Thus the major challenges for storage in HPC are to provide shared access to data, with guaranteed data integrity, and to provide high performance levels while minimizing idle compute time. Shared access to storage decreases the complexity of HPC by making data uniformly accessible, obviating the need to divide the entire data set or to replicate and distribute the data.

One approach Linux cluster administrators have tried is the use of one or more NFS servers. This approach works well for smaller environments of up to a hundred nodes, at which point the NFS servers become the application bottleneck. For larger or more I/O intensive environments, more NFS servers must be used, making management and load balancing difficult, and at times impossible.

Client I/O performance scaling

The total aggregate throughput of an HPC system is a function of the demands of individual nodes multiplied by the number of concurrent nodes. In many cases the throughput required by an individual node can be relatively modest (by HPC standards that is), but because the total number of nodes is large, the aggregate throughput can be very high. For example, 1,000 concurrent nodes, each demanding only 100 Mb/s, would generate an aggregate demand of $(1000 \times 100 = 100,000 \text{ Mb})$ approximately 12 GB/s.

The ability to support large numbers of clients, while critical to HPC, is not the only dimension of scalability. Per-client performance is also important. In practice, there are a number of factors, including caching and parallel access to storage, that combine to improve per-client performance.

Some types of applications, such as chip design and compilation, produce very high random I/O loads. While the compute nodes run these applications, they can touch hundreds of thousands of small files, performing mostly read operations, with a very few large write operations at the end of the job. Modal analysis applications, on the other hand, typically use datasets containing very large files, often greater than a gigabyte in size. Such applications use very large I/O operations (up to multiple megabytes) and demand very high storage system throughput rates. In all cases, HPC demands highly concurrent access to shared data.

In summary, providing secure, shared access to files requires that there be a common repository for the files, that is, a networked file system. Networked file systems incorporate a metadata server to track the structure of the files and the locations of file blocks on disk. The metadata server facility, which controls access to the files, is internally maintained by a file server when serving NFS file systems. But, as mentioned earlier, use of NFS file servers has performance scaling limitations and/or data management challenges for HPC environments. As we will see, the MPFSi architecture provides the solution to this problem by moving the file server out of the data path, “externalizing” the contents of the metadata server to permit each node to access the data blocks comprising the targeted file directly from the storage SAN.

Data sharing with MPFSi

The best way to explain how MPFSi implements data sharing across an HPC topology is to compare it to a standard NFS mechanism, from which it is derived. MPFSi has an agent that resides on each HPC node and uses NFS semantics to enable nodes to access shared files managed by a Celerra file server. For both NFS and MPFSi, file system metadata operations are processed by the Celerra file server using standard NFS semantics to assure security of access and data consistency:

- Inode allocation
- File attribute management
- File-block map allocation
- File access control
- Structure (bitmap) allocation
- Logging

But unlike conventional NFS file serving, data delivery occurs between client and storage directly over the SAN, freeing valuable Celerra file server cycles to support a larger number of clients than a conventional NFS file server does. The MPFSi agent and Celerra file server use FMP to pass metadata to the client, enabling file data read/write operations directly with the storage array over the SAN. The MPFSi client uses FMP to ask the Celerra file server for the location of the file or to allocate space for the file, then performs I/O directly to the on-disk volume. The Celerra file server locks the file or byte ranges within the file and identifies its respective location on the storage array. The Celerra file server will also inform clients about lock revocations and file mapping changes through FMP Notify messages.

Parallel data access

The MPFSi architecture has been designed such that all of the clients have direct access to storage that underlies the Celerra file server's file systems. This results in multiple levels of parallelism for data access. Other features include the following:

- MPFSi clients can read and write any data file concurrently by using the FMP protocol's range locking.
- The Celerra file server and all MPFSi clients can access *multiple* storage arrays in parallel. As such, total throughput of the system is bounded only by the aggregate throughput of the storage arrays or the SAN or LAN itself.
- All MPFSi clients can access (read-write) multiple Celerra servers concurrently (files cannot be shared across servers).
- NFS/CIFS clients and MPFSi clients can have concurrent read-write access to the same file.
- All MPFSi file systems on a Celerra are wide striped across multiple logical unit numbers (LUNs). This creates a fan-out effect where a single large I/O from a client is split into multiple smaller I/Os directed to a large number of LUNs.
- LUNs in the storage array are members of RAID groups (RAID 3 or RAID 5). This adds an additional level of parallelism by striping each LUN across multiple spindles. Each I/O sent by an MPFSi client to a LUN will be sent in parallel to a larger number of spindles.

MPFSi components

An MPFSi-based HPC storage topology comprises:

- A Celerra file server
- A number of HPC compute nodes as MPFSi clients (each with MPFSi agent and iSCSI initiator)
- The MPFSi client LAN infrastructure
- One or more CLARiiON or Symmetrix storage arrays
- The iSCSI + FC storage network

These components and their relationships are shown in Figure 2. Note that the storage network manifests as an FC switch that features both 2 Gb/s FC ports for storage array and Celerra file server connection, and Gigabit Ethernet iSCSI target ports (maximum of 40 with the MDS 9509 switch) for connecting to all the

HPC-node iSCSI initiators using the network infrastructure. The FC switch transparently handles bi-directional FC target to iSCSI initiator bridging.

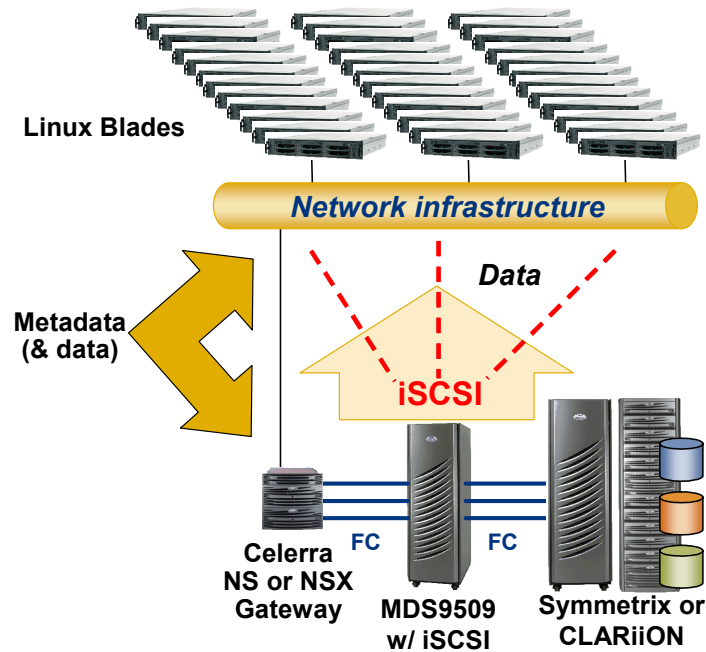


Figure 2. MPFSi components

The MPFSi agent

The MPFSi agent software runs as an installable file system on the Linux client platform. This file system is derived from the client's native network file systems and fully supports the NFS protocols. The agent selectively intercepts or forwards NFS protocol messages. Figure 3 illustrates the division of operations within the MPFSi agent.

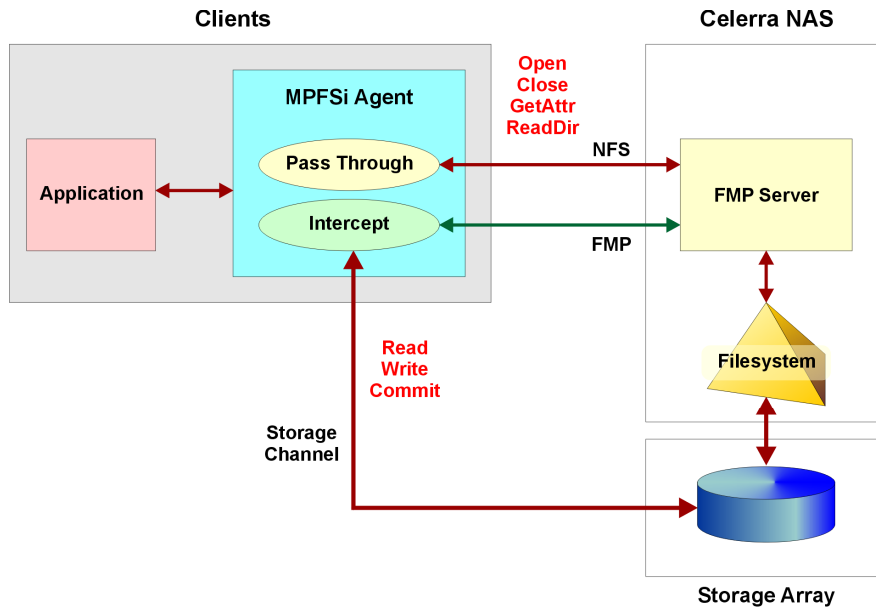


Figure 3. Splitting data and metadata operations at the MPFSi client

The MPFSi agent interacts with the Celerra file server for synchronization, access control, and metadata management. The MPFSi agent provides the following I/O benefits for HPC nodes:

- High bandwidth

A major benefit of the MPFSi architecture is its ability to move data across Gigabit Ethernet iSCSI SAN directly between HPC nodes and storage to provide increased bandwidth to the HPC node compared to conventional NFS. Under the right application circumstances, a single client node will be able to realize very close to the line speed of each of its Gigabit Ethernet links. Figure 4 illustrates the benefit an individual client would experience using MPFSi.

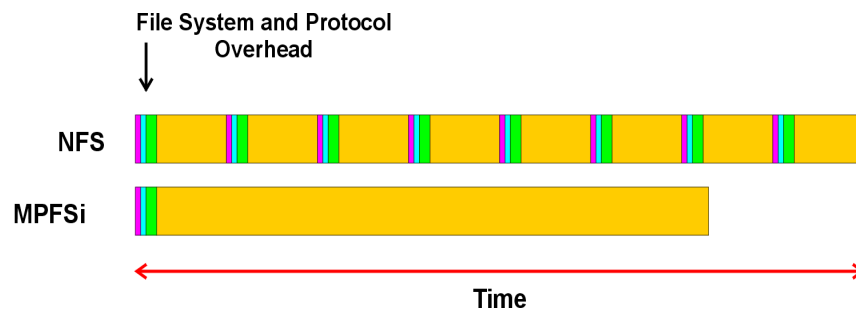


Figure 4. Higher client read/write throughput

Connecting the client directly to the storage array reduces TCP/IP overhead, and the same amount of data can be moved more quickly, increasing throughput. It is very important to note that this illustration reflects client operations and does not consider other significant benefits. The Celerra server is using only a fraction of the CPU resources it would normally require, permitting up to 10 times more clients to share a common namespace than the server could facilitate using conventional NFS.

The reason for using MPFSi is to provide high-performance *shared* access to common data. In an environment typical of HPC, where a great many clients are reading the same data and even the same file, the benefits of multiple levels of caching come into play. In such cases, aggregate throughput can

be much greater than that facilitated by conventional NFS file serving. With each node having its own low latency FC connection to storage, the aggregate bandwidth afforded to a cluster of HPC nodes will be limited primarily by horsepower of the storage infrastructure (that is, the number of storage processors, cache, and disks).

- Metadata management

The MPFSi client also benefits from a very efficient metadata extent caching mechanism for file block maps (extents). The client prefetches a large number of file block extents with each call to the FMP server. Most of the time entire file block maps are cached by the client before the data I/Os are sent to the storage array.

In Figure 5, the client requests the entire file map from the Celerra server, where it is already cached, requiring no disk operation. The client caches the file map and starts to issue I/O requests to the storage array using read prefetch to cache the file data and make it ready for the application. As the file is laid out sequentially on disk by the metadata server, using the read prefetch capabilities of the storage array enables the MPFSi client to get the data much faster than if the application was directly requesting the blocks.

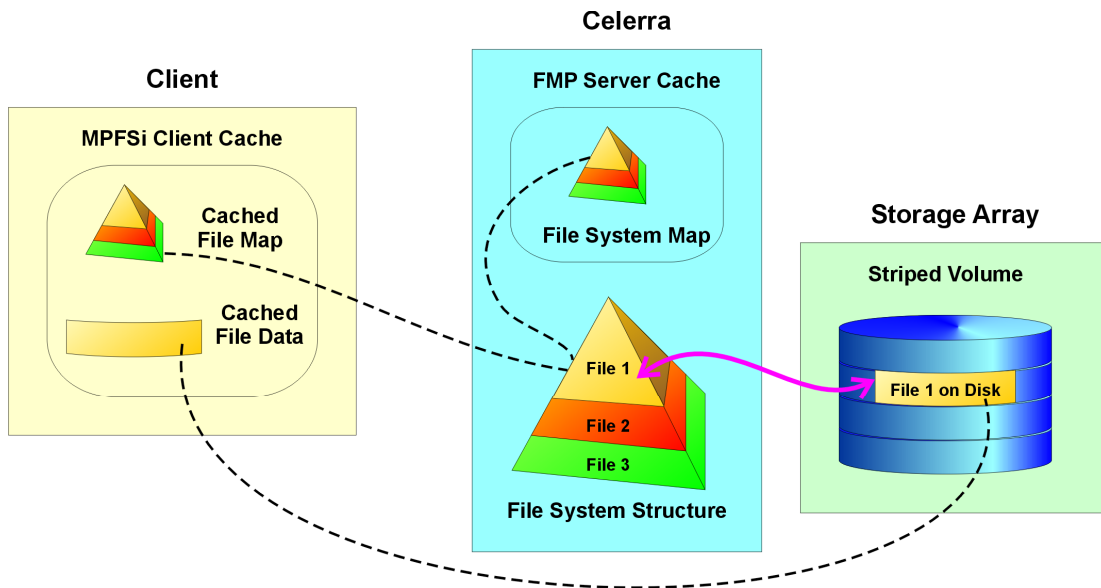


Figure 5. Metadata management

- Parallel data access

When HPC nodes access the same file concurrently, the data will reside in the cache of the disk array, allowing very high data throughput to a single file. The MPFSi client driver further accelerates data bandwidth when accessing a large file by using much larger I/Os sent from the array than conventional NFS uses, exploiting parallel access of volumes striped across RAID groups. With larger I/O size, HPC nodes can simultaneously take advantage of sequential layout of a large file on disk to avoid time-consuming track seeking activity encountered by conventional NFS disk access using small I/Os.

- Small I/O efficiency

When the application requests to read a small file (an I/O smaller than the on-disk file system block size of 8 KB), the MPFSi agent will ask the server to send the entire file content using a single NFS request. In effect, the data is delivered for the cost of the file map alone.

- Intelligent data caching

MPFSi agents use client memory to optimize data access and caching. Using a write-behind cache, a large amount of the application-write data can be cached and sent to be written to the storage array in a larger I/O. This is achieved by using the cache capabilities of block devices in the client kernel. Similarly the MPFSi agent does intelligent read-ahead or prefetching of the blocks for a volume and has them available in the block device buffers for maximum access performance of the application.

The Celerra file server

The FMP server on the Celerra is the central component of the MPFSi architecture. The FMP server is the owner of the file system, defines its on-disk format, and controls all access to the file system. Celerra core file services (on which MPFSi depends) provide clients with a uniform view of file data – independent of protocol (NFS) or the client operating system. Celerra provides concurrent access to the same data, by multiple clients.

The FMP server and other core Celerra file services provide the following:

- A file system capable of disk space reservation (uncommitted space allocations)
- Concurrent access to the same files through NFS
- Concurrent access to the same files for MPFSi clients and NFS and CIFS network clients
- Delivery of small I/Os to the MPFSi clients over NFS using network fabric

The FMP server can own and serve one or more file systems. The server also performs file system block reservation and preallocation of file space on the file system with a *delayed commit* policy. This enables MPFSi writers to allocate aggressively large chunks of contiguous file blocks, then free any unused space when the file is closed. This supports higher performance for sequential access, which in turn can exploit the storage arrays' caches by use of aggressive prefetch settings. Finally, the FMP server contains arbitration logic to support concurrent access to file data from large populations of compute nodes. The FMP server communicates with the client by means of the FMP and FMP notify protocols.

The FMP server manages the inode-level attributes for its file systems. This enables the clients to use their own caches to perform metadata prefetch and I/O merge locally, thus increasing performance and parallelism, and benefiting individual clients as well as the overall client community.

In addition to the core capabilities listed above, the MPFSi Metadata Server provides the following services for the HPC Cluster:

- Authentication – An important role of the FMP server is to identify and authenticate the MPFSi clients to access the storage system, similar to the way the Celerra file server authenticates the NFS and CIFS clients. When the MPFSi client wants to access a file system, the FMP server assures its identity and provides authorization, according to the security level dictated by policies within the environment.
- File and directory access management – The FMP server provides MPFSi clients with the structure of the files in the file system. When the MPFSi client requests an operation on a particular file, the server examines the permissions and access controls associated with the file and sends the client a map of the entire file to be cached by the client. The file map consists of the list of volumes ID, offsets, and block counts (a set of adjacent disk blocks or extents) in a run-length optimized form that permits the client to go to the storage array directly and access the disk blocks.
- Cache coherency – The MPFSi client will cache the data locally in the node's block storage device cache. Celerra's FMP connection to the client is validated by successful heartbeats between the server and the clients. If the heartbeat fails, the client stops any I/O activity and invalidates all the cached file maps and data blocks. The FMP server controls access to the file by managing locks to govern access permission to each file system block. Each block may be unlocked, locked for write by a single client, or locked for read by one or multiple clients. The FMP server allows multiple writers so if a block is modified by one client, the server notifies all the clients that have the file open using the FMP Notify protocol. Clients that access the same block range are notified to invalidate their local cache, go back

to the FMP server to refresh the (cached) local copy of that file map, and go to the storage array to read the appropriate data blocks.

- Scalability – The FMP server (Celerra) performs two metadata management tasks: file/directory access management and file system block allocation, enabling all client data traffic to pass directly to and from the storage array. Testing has shown that, compared to similar NFS workloads, the file/directory management is 7 percent of the CPU workload (write) and the block allocation is 4 percent of the CPU workload. The remaining 89 percent is typically used for data transfer for NFS. With MPFSi, these CPU cycles are offloaded from Celerra to the storage array. Thus, the number of clients that can be served by a single MPFSi-enabled Celerra server is an order of magnitude higher than that possible with conventional NFS.

Storage arrays

Celerra MPFSi solutions are implemented on one or multiple Symmetrix DMX™ or CLARiiON arrays. MPFSi-enabled clients connect to the storage at the highest possible speeds with one or more Gigabit Ethernet connections. MPFSi is able to exploit the advanced features of the EMC intelligent storage arrays, including online snapshots (TimeFinder®/FS), disaster recovery (SRDF®), and Nested Mount File System.

The storage array, which contains the logical volumes used by the FMP server to build the physical file systems, is responsible for all the operations related to the management of the disk blocks. It serves data to the metadata servers (MPFSi and conventional NFS) and to the MPFSi-enabled compute nodes. It is responsible for protecting data on disk and in cache. The storage arrays use powerful processing modules with a large number of tightly coupled CPUs, large mirrored RAM caches, and sophisticated algorithms to allow maximum performance. The storage arrays are tuned to maximize the throughput of the FC pipes. They offer multiple paths to the disk drives for both path redundancy and performance.

EMC storage arrays use advanced RAID algorithms using wide striping to increase the parallelism in disk access and take advantage of the disk geometry and the large caches. They are also optimized for efficient use of the cache and adapt to a large range of workloads. Disk blocks are arranged in RAID groups, presented as logical units, and are managed using built-in advanced disk management tools. The storage capacity and number of disks can be sized to fit a variety of situations.

EMC storage arrays can support different types of disk drives in the same frame, including FC disks of different sizes and speeds, and various ATA and SATA drives. Arrays are optimized for maximum performance for each specific type of disk drive. Optimization uses different cache algorithms and configurations, different types of RAID matching the disk characteristics, and different prefetch algorithms. Access to the data blocks is through offsets and lengths. Blocks can be organized into different hierarchies and can be accessed through different physical FC ports in parallel. All metadata management is left to the metadata servers that own the file systems (MPFSi and/or conventional NFS). As such, the storage arrays can focus on their primary functions – protecting and serving data as efficiently as possible.

Connectivity

HPC nodes employ Gigabit Ethernet IP connectivity to the Celerra server for metadata delivery, and to the FC switch that translates iSCSI protocol to and from FC protocol for attachment to CLARiiON and/or Symmetrix arrays. The extent of the topology is bounded by the physical limits of the SAN and IP network fabric and storage arrays.

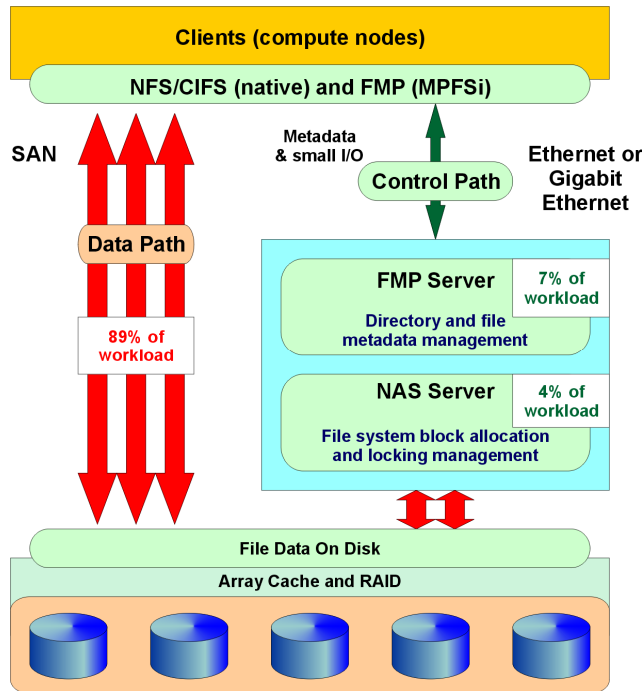


Figure 6. MPFSi architecture

For illustration, consider an NFS server powerful enough to serve 110 compute nodes at a given throughput rate. A similarly configured MPFSi solution (see Figure 6) would be able to serve about 1,000 compute nodes. This degree of scalability translates into more than just performance – it directly benefits operational complexity and cost as well. Other solutions would require multiple servers to support the same number of clients (perhaps 10 times as many). This alone adds considerably to complexity, regardless of the actual component cost of the individual servers. The administrative costs are clearly higher with such a system.

MPFSi performance measurements for Linux clients

MPFSi has been extensively characterized. When compared to NFS, MPFSi:

- Increases bandwidth to individual clients by two to four times
- Permits up to ten times the number of clients to be served by a Celerra file server

The combination of the bandwidth increase to clients and the larger number of clients supported by a Celerra file server permits hundreds of HPC nodes to work in concert on common data with significantly reduced execution duration than facilitated by conventional NFS. Note that maximum aggregate throughput to clients will be limited by the nature of the operation mix applied to a specific number of array controllers and disk drives.

The MPFSi performance scaling measurements shown below were made using a Celerra NS704 file server with four blades connected to one, two, three, and four CX700 arrays each configured with >100 146-GB 10,000-rpm disks. Synthetic I/O load generators (for example, IOzone) running on 10-50 x86-based Linux (RHEL 3.0) were used to generate client load to assess the performance advantage afforded by MPFSi as compared to NFS. Performance was measured for the following data access:

- For NS704 connected to single CX700 array with > 100 disks:
 - Random Read MB/s vs. I/O size
 - Sequential Read MB/s vs. I/O size
 - Random Write MB/s vs. I/O size

- Sequential Write MB/s vs. I/O size
 - MPFSi single/multiple client aggregate scaling (reads)
- For NS704 connected to one, two, three, and four CX700 arrays, each with > 100 disks:
 - MPFSi Data Mover + Array scaling – reads
 - MPFSi Data Mover + Array scaling – writes

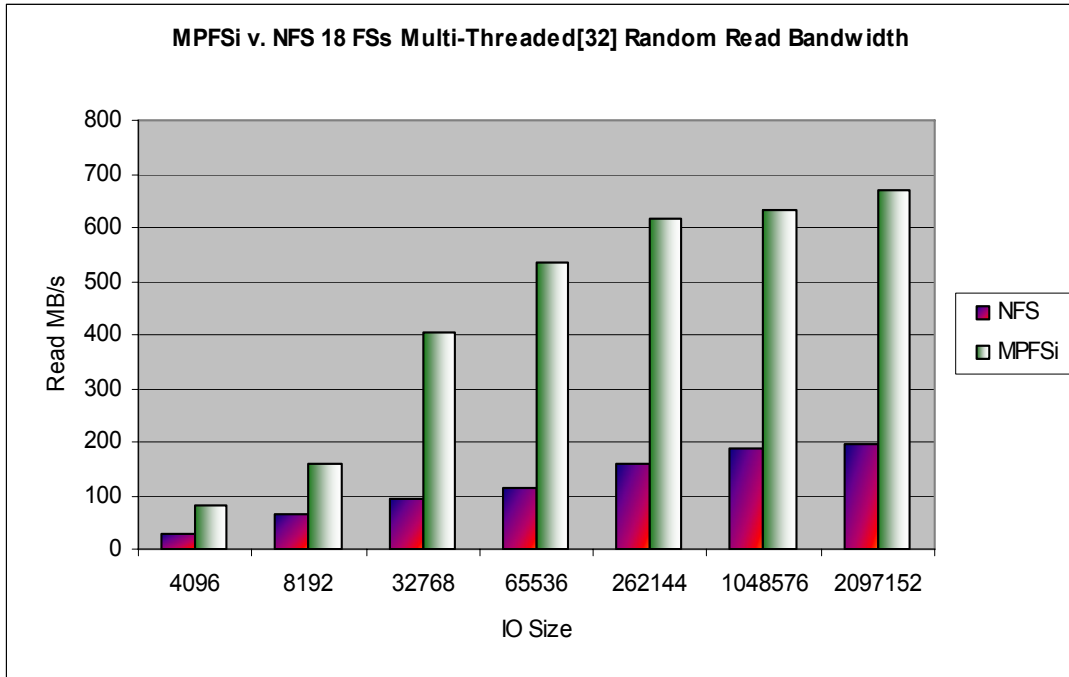


Figure 7. MPFSi Random Read performance for Linux clients

Figure 7 depicts bandwidth measurements across increasing I/O sizes for a random pattern of reads. With each of 10 MPFSi clients running 14 threads, basic NFS for a Celerra NS704, plus a single CX700 array, offers 30 MB/s to 194 MB/s aggregate bandwidth to the clients, each configured with 3 file systems and operating with 32 threads. Compared to NFS, for this same workload MPFSi facilitates a three times bandwidth improvement for I/O sizes at 4 KB, and up to four times improved bandwidth for 2 MB I/Os. Note that MPFSi Random Read performance for these clients approaches an asymptote at 700 MB/s aggregate bandwidth, which is close to the limit of CX700 hardware configuration being tested. Compared to NFS, MPFSi enables use of a much greater percentage of the available bandwidth of the CX700's 100-plus drives while facilitating file sharing to multiple clients.

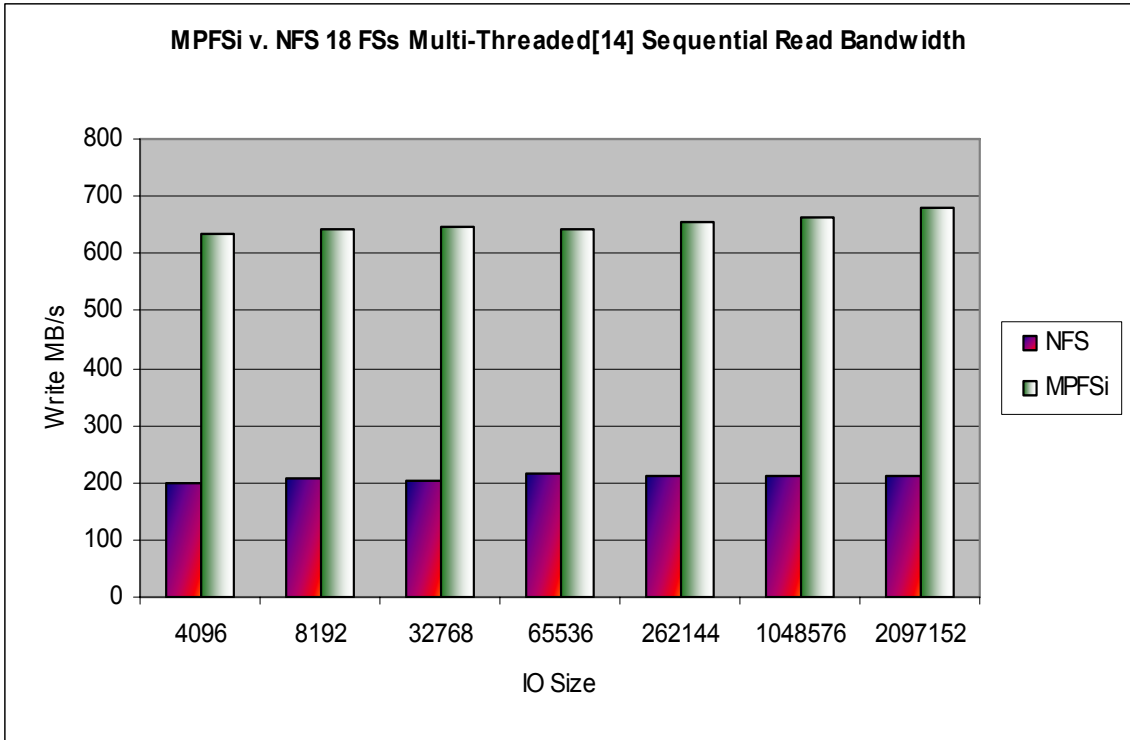


Figure 8. MPFSi Sequential Read performance for Linux clients

Applying a Sequential Read workload to the NS704+CX700 platform exhibits a similar performance improvement when comparing MPFSi-enabled clients to clients using basic NFS. MPFSi offers three to four times the improved bandwidth at all I/O sizes, as shown in Figure 8.

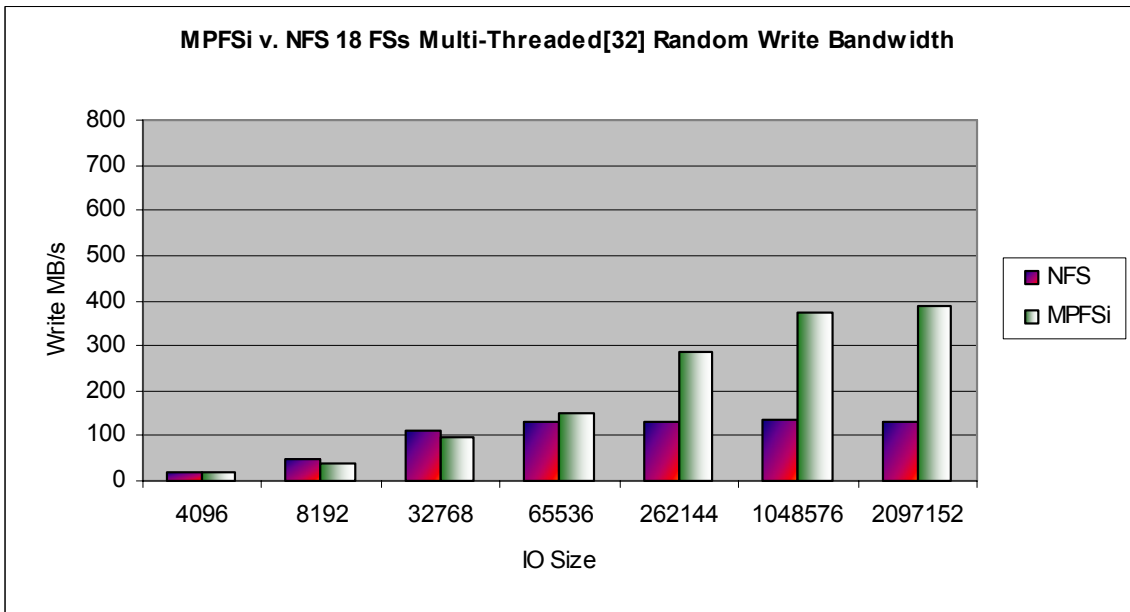


Figure 9. MPFSi Random Write performance for Linux clients

As with reads, MPFSi demonstrates Random Write performance that is superior to conventional NFS once the I/O size is at least 256 KB. Sustained bandwidth for a CX700 with > 100 disks can approach 400 MB/s or approximately three times the straight NFS performance, as shown in Figure 9.

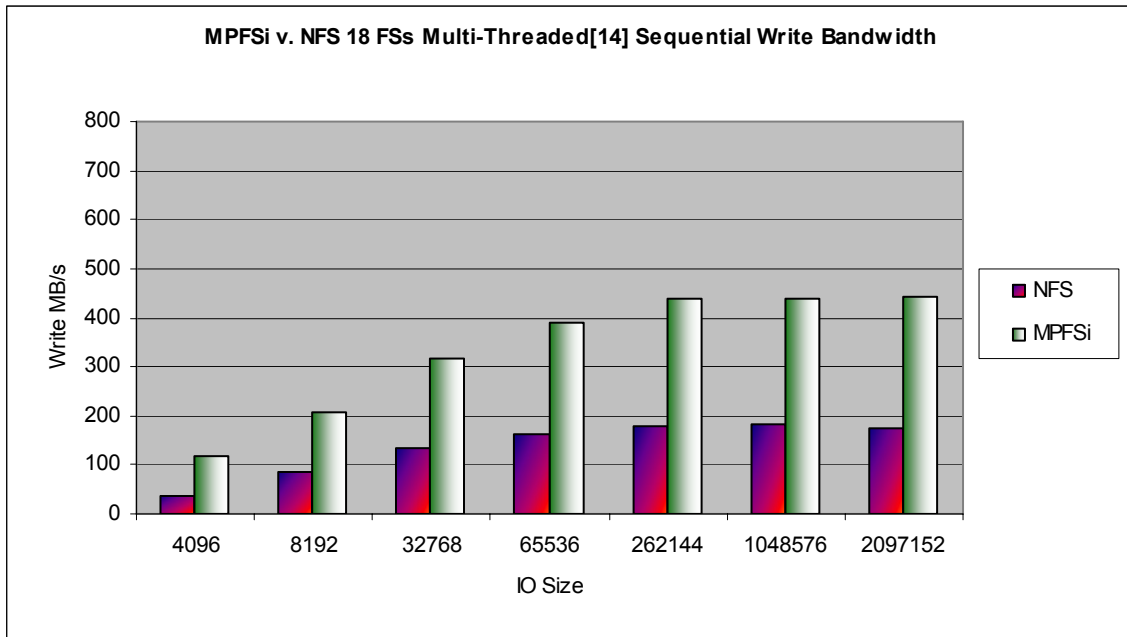


Figure 10. MPFSi Sequential Write performance for Linux clients

As with Sequential Reads, MPFSi Sequential Write performance exceeds straight NFS by >2 times at all I/O sizes and exceeds 400 MB/s at 256 KB I/O size, as shown in Figure 10.

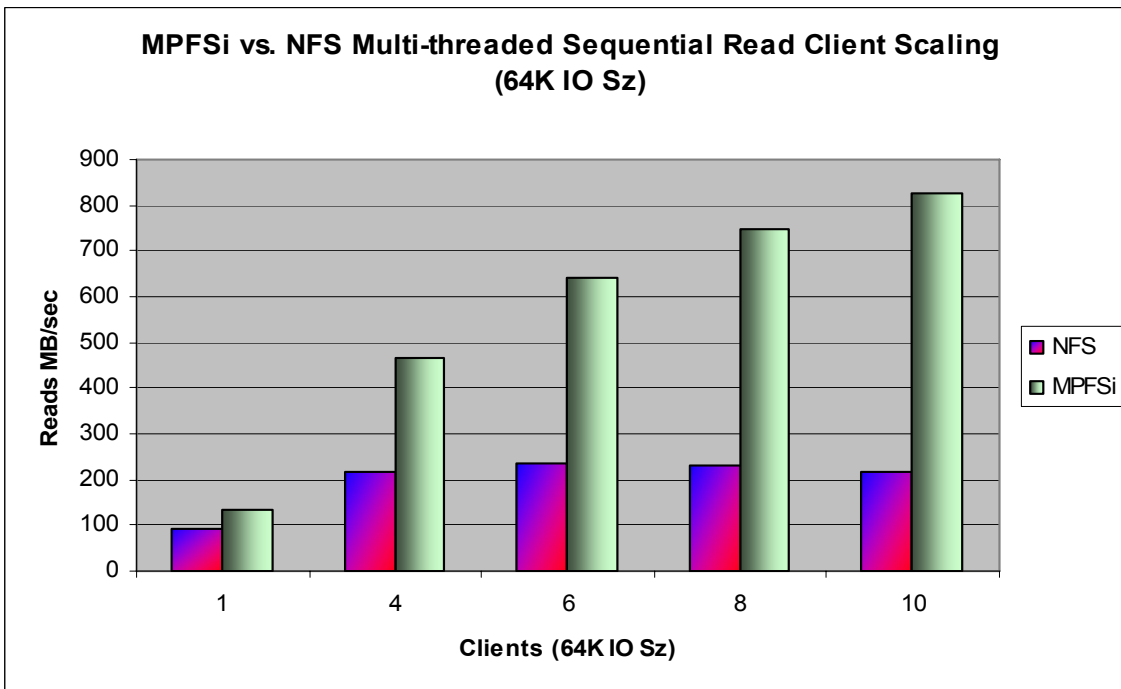


Figure 11. MPFSi single/multiple client aggregate scaling - reads

Additional lab measurements have shown that the maximum sustained Sequential Read bandwidth to a single client with a single Gigabit Ethernet SAN connection exceeds 130 MB/s. When three additional clients are connected, aggregate Sequential Read bandwidth increases by more than three times to 468 MB/s, as shown in Figure 11. Ten clients can achieve over 800 MB/s while NFS aggregate bandwidth remains at 200 MB/s. The performance scaling virtue of having each client use its own iSCSI connection to the storage SAN is clearly apparent.

From this test, one can conclude that each of the 10 clients is realizing approximately better than 80 MB/s of sustained read activity using MPFSi to access a single CX700 array. A key value proposition of MPFSi technology is that the client to IP SAN topology can be scaled to achieve greater aggregate MB/s bandwidth than a single array can facilitate. Figure 12 shows that when a 50-client environment is connected to one, two, three, and four NS704 Data Mover blades each connected to one CX700 array, the result is a near linear aggregate bandwidth increase as each successive array is added. In the case of reads, a single MDS9509 SAN switch can facilitate at least 2.6 GB/s to a HPC cluster of Linux clients, which is four times what NFS can offer for the same 400-plus disk storage pool configuration. (Experiments have shown that adding a fifth array with 100-plus disks would permit the SAN to exceed 2.6 GB/s throughput.) Figure 13 shows similarly that MPFSi can facilitate three times the NFS bandwidth, or > 1.3 GB/s, when performing sequential writes across a storage pool of one, two, three, and four arrays.

Note that while bandwidth scaling for conventional NFS would require adding one or more Data Movers, segmenting the application's namespace, MPFSi does not require any additional Data Movers as each Data Mover's CPU is observed to be utilized as low as 10 percent. It is only supplying metadata while the real work of data delivery is being directly handled between each client and the storage arrays.

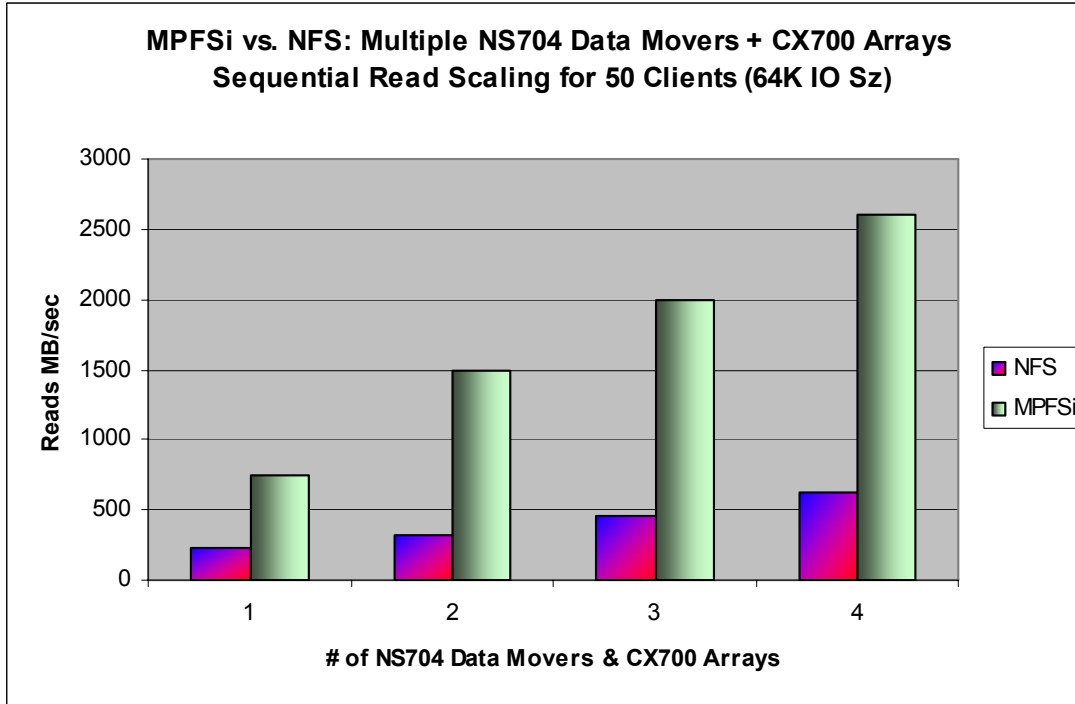


Figure 12. MPFSi Data Mover + Array scaling - reads

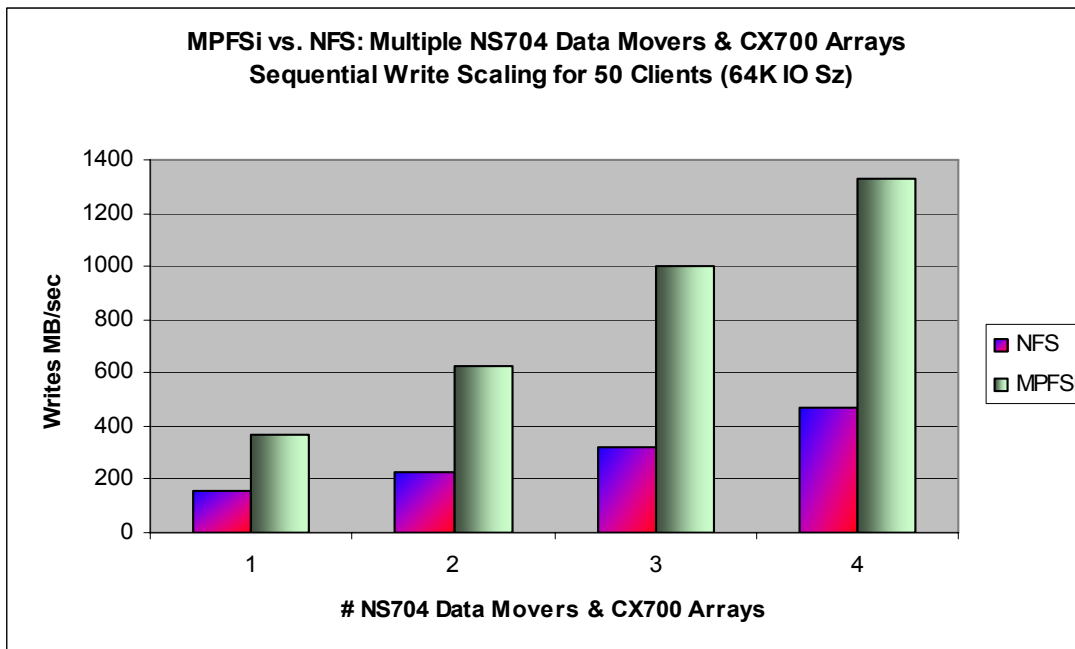


Figure 13. MPFSi Data Mover + Array scaling - writes

Summary discussion regarding performance measurements

We have shown that MPFSi performance can be up to four times the sustainable bandwidth of NFS for random and sequential reads and writes. And we have established that scaling the absolute sustainable performance to one or all client nodes in a HPC environment is primarily a matter of providing more backend storage infrastructure to support increased data access activity. Thus, MPFSi offers the standards-based simplicity of NFS data sharing with block access performance attributes – a perfect solution for today’s HPC storage demands.

Conclusion

The future of HPC is becoming increasingly dependent on computation nodes sharing critical information at high aggregate throughput. MPFSi facilitates this sharing using standard NFS access protocol, at a level of performance scaling unmatched by competing approaches. MPFSi’s parallel storage access architecture provides centralized, easy to implement and managed standards-based NAS access control with high-speed parallel data delivery of SAN, yielding client bandwidth increases of two to three times over conventional NFS. MPFSi is well suited for a diverse set of applications that access data in I/O chunks of 32 KB or greater, such as CAD, computational fluid dynamics, seismic exploration, proteomics, data warehousing.

For more information on MPFSi and other EMC products, please visit www.EMC.com.

Bibliography

1. S. Faibish, et al., “Joint SAN and NAS architecture of Disk Based Storage for Media Applications”, *Proceedings of the 144th SMPTE Technical Conference and Exhibit*, Pasadena, October 23-26, 2002.
2. Draft-Welch-pnfs-ops-02.txt, Draft 02 of FMP protocol, www.ietf.org.