

Quality of Service support for Grid Storage Environments

Vijay Velusamy, Anthony Skjellum
High Performance Computing Laboratory
Department of Computer and Information Sciences
University of Alabama at Birmingham
Birmingham, AL, USA

Abstract: Grid computing enables the utilization of geographically distributed heterogeneous resources for processing data. In such large-scale applications, availability of data is essential for efficient utilization of the resources, especially since data utilized by the applications may not be available at the site of execution. The distributed nature of these grid applications and the disparate nature of storage resources warrant some guarantees to access the data that may be distributed among grid locations, while considering economic and utility values.

Quality of Service (QoS) serves to allow delivery of data in real-time to enable data consuming applications to execute without awaiting completion of data generating applications. Though there have been studies to analyze data streaming in grid computing environments, these strategies have minimal support for any type of QoS or utility value. This paper presents a novel approach to support QoS requirements for grid storage systems, by offering statistical QoS assurances using data replication, rather than a best-effort basis. A framework for QoS enabled data-intensive applications in a grid environment based on pNFS is designed for this purpose.

Keywords: Grid Storage, QoS, pNFS, Storage Economics

1. Introduction

In large-scale distributed applications designed for grid computing environments, availability of data is essential for efficient utilization of the resources in workflow systems. This is especially significant when data being generated by an application at one grid location is simultaneously required by one or

more applications at other grid locations, typically observed in large-scale scientific simulation applications. This distributed nature of grid applications warrants some guarantees to access the data that may also be distributed among grid locations, while considering economic values related to data transfer, storage and access.

In a real-time scientific visualization system, for example, generation of real-time data from multiple sources provides constant data updates. For some applications, reasonably recent data that is immediately available suffices for successful execution. This enables efficient progress of the overall simulation in a grid computing environment.

The availability of data to the various grid applications can be increased by replicating the data at storage locations more accessible to these grid locations. Quality of Service (QoS) serves to allow delivery of data in real-time to enable data consuming applications to execute without awaiting completion of data generating applications. Although there have been studies to analyze data streaming in grid computing environments, these strategies have minimal support for any type of QoS or utility value.

Current parallel file systems are designed for cluster storage environments and extended to grid storage environments by supporting data grid frameworks such as Grid Datafarm [1], Storage Resource Broker [2], or GridNFS [3] to such file systems. This paper presents a more fundamental framework for extending a parallel file systems standard based on Parallel Network File System (pNFS) and incorporating QoS guarantees combined with economic modeling.

2. Grid Storage Systems

Distributed storage technologies are generally designed for delivering performance to multiple clients accessing data from locations close to the application that is using this data. The global nature of data-oriented applications has influenced the adoption of storage resources, such as NAS or SAN for wide-area distributed data access.

2.1. Internet Back-plane Protocol

The Internet Backplane Protocol (IBP) is a middleware for managing and using remote storage units called storage depots to support Logistical Distribution Networks (LoDN) in large scale, distributed systems and applications. Data movement is globally scheduled for optimization of storage based on a model that takes into account all the network's underlying physical resources. It follows the principle of storage warehouses (depots) and distribution channels for logistics of industrial activity. However, IBP is a best-effort service for managing allocations on storage depots. In order to offer maximum scalability, IBP storage allocations are lightweight and are time-limited. Soft storage allocations are made using free space in the local file system, and may be reclaimed at any time by the local system. Hard allocations are made using dedicated IBP storage space, and are highly stable offering stronger guarantees that stored data will remain intact and accessible for a specified duration.

IBP facilitates the combined use of replication, fragmentation, and striping of file fragments across multiple depots to provide fault tolerance and accessibility. Although IBP allows the specification of a desirable striping and redundancy pattern (number of replicas), it does not allow the specification of where the replicas may be stored. Economic considerations are absent in IBP.

2.2. Parallel Network File System

Parallel NFS [3-5] (pNFS) is a standard that provides extensions to the NFSv4 standard [6] with support for exporting storage systems in a scalable manner. Distribution of data in storage subsystems in NFSv4 is limited to RAID technology and does not support a more flexible layout scheme. The pNFS extensions provide specifications on protocols for supporting a metadata server which provides clients with layout information on files that may be striped on storage subsystems (RAID of disks or disk arrays that could use interconnects such as Fiber Channel, SCSI, or

iSCSI). In this model, once the clients obtain the metadata from the metadata server, the clients will directly communicate with the storage subsystems for accessing the data, thereby avoiding any bottlenecks associated with accessing data through a single server, as the case may be in NFSv4.

Although pNFS is designed for scalable access to data that is exported by NFSv4 servers, pNFS proves ideal for studying QoS issues related to grid storage environments, since it has the potential to be used for exporting file systems in wide area networks as well. The suitability of pNFS for cluster file systems as well as grid storage environment is a major trait for this study.

2.3. GridNFS

GridNFS [3] is an effort to explore the adaptability of NFSv4, combined with parallel data access paradigms supported by parallel NFS (pNFS) to a grid environment. In this case, clients access data that is distributed in multiple servers, controlled by a metadata server, and file locking mechanisms (status information) provided on separate servers. QoS issues and economic considerations are absent in the GridNFS project.

3. QoS for pNFS based Grid File System

The Parallel Network File System (pNFS) extensions to NFSv4 provide specifications on protocols for supporting a metadata server which provides clients with layout information on files that may be striped on storage subsystems (RAID of disks or disk arrays that could use interconnects such as Fiber Channel, SCSI, or iSCSI). In this model, once the clients obtain the metadata from the metadata server, the clients will directly communicate with the storage subsystems for accessing the data, thereby avoiding any bottlenecks associated with accessing data through a single server, as the case may be in NFSv4.

Even though pNFS was targeted at data exported by NFSv4 servers, pNFS proves ideal for studying QoS issues related to grid storage environments, since it has the potential to be used for exporting file systems in wide area networks as well. The suitability of pNFS for cluster file systems as well as grid storage environment is a major trait for this study. Also, pNFS specifies the protocols for the client to access the metadata server, and obtaining the layout(s) of the file. The actual data access, metadata operations use traditional NFSv4 protocols to access the storage

system. The pNFS specification does not address the issues related to selection of layouts by client, economic value associated with layouts, and QoS issues related to accessing the data.

The Application accesses data through the pNFS Client (Figure 1). The Client requests the layout for the data from the pNFS Server. The pNFS Server returns the layout for the data. The pNFS Client accesses the storage system via the Layout Driver and I/O Driver directly using parallel I/O technology, such as SCSI, iSCSI, InfiniBand or Fiber Channel. Among possible simple layouts for distributing data are:

- Round Robin – Blocks striped on storage nodes in round robin fashion (e.g., RAID0)
- Replicated – Each block exists on more than one storage node (e.g., RAID1)
- Parity – Round Robin striping with distributed parity (e.g., RAID5)
- Nested – Layouts comprising simpler ones

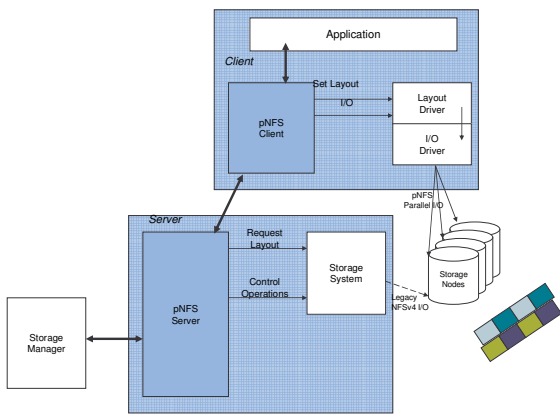


Figure 1. pNFS Architecture

There are typically three levels of granularity associated with applications accessing data using pNFS.

1. Single pNFS Server associated with single storage system that provides multiple layouts (Figure 2): The single storage system controls a set of storage nodes, on which there could be multiple copies of the same data in different layouts.

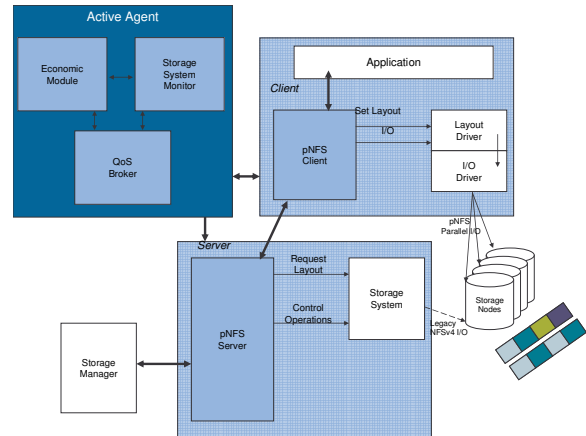


Figure 2. QoS Support for pNFS : Single Server – Single Storage System

2. Single pNFS Server associated with multiple storage systems that provides multiple layouts (Figure 3): Each storage system controls a set of storage nodes, which may contain one or more copies of the same data with different layouts (replicas).

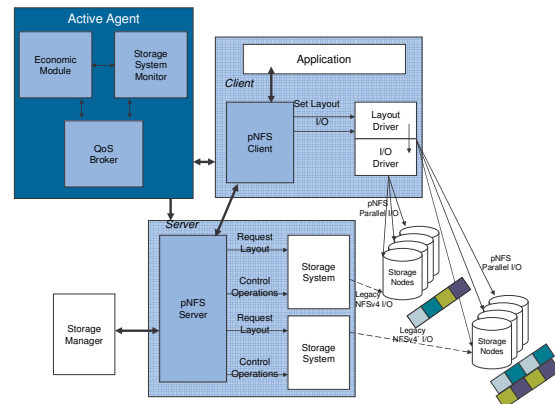


Figure 3. QoS Support for pNFS : Single Server – Multiple Storage Systems

3. Multiple pNFS Servers associated with multiple storage systems that provides multiple layouts (Figure 4): Each storage system controls a set of storage nodes, which may contain one or more copies of the same data with different layouts (each pNFS Server has a set of metadata associated with layouts).

On a Wide Area Network a grid storage environment is similar to 3, where multiple pNFS Servers provide different layouts. Multiple portions of a file could be

accessed from different storage systems (Server + storage node pair in this case).

It is assumed that making replicas and maintaining consistency, synchronization is taken care of by the Storage Manager. Typically such management services are provided by the storage vendor and are outside the scope of this study.

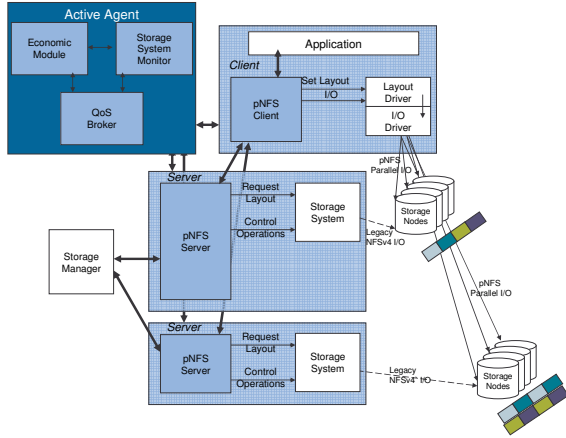


Figure 4. QoS Support for pNFS : Multiple Servers – Multiple Storage Systems

4. Active Agent

The Active Agent (in Figure 2 - Figure 4) designed to provide QoS guarantees with economic consideration will be further explained in the following sections. The QoS Broker obtains information about servers, layouts from Server, uses Economic module to decide which layout application should use. The QoS Broker contains information about location of replicas in various servers. In a grid storage environment, the QoS Broker is designed to communicate with Metadata Catalog (MCAT) servers to obtain this information. The Storage System Monitor monitors status of storage subsystem related to load, network traffic from Server. The Economic module has information on cost models for accessing storage, suggests optimization parameters to QoS Broker.

4.1. Storage System Monitor

The Storage System Monitor monitors status of storage subsystem related to load, network traffic from Server periodically.

4.2. Economic Module

The Economic Module obtains and stores (updates when necessary) the data transfer and storage costs. The Economic module uses a producer-consumer model to evaluate the cost of accessing the data.

4.2.1. Single Provider – Single Consumer model

The Single Provider – Single Consumer model is the most basic model that might be considered. This is analogous to one storage system accessing/copying data from another storage system (as shown in Figure 5).

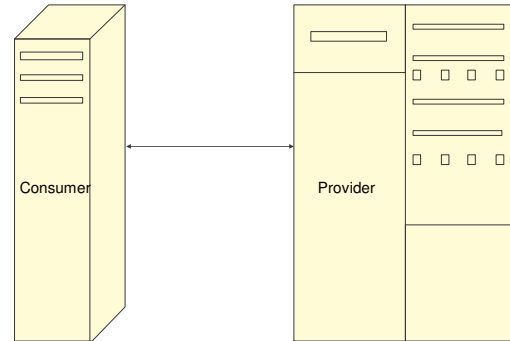


Figure 5. Single Provider – Single Consumer model

Let BW_{max} be the maximum bandwidth achievable between the two systems. However, in reality, BW would be the bandwidth achieved for transferring data. The fraction of bandwidth achieved f is given by

$$f = \frac{BW}{BW_{max}} \quad [1]$$

The time taken to transfer l bytes of data is the sum of setup time and time for transferring l bytes at the bandwidth BW . The cost is the sum of the cost for transferring l bytes of data over time T at the fraction of bandwidth achieved f .

$$\begin{aligned} Time(l) &= \alpha + (1/BW)l \\ Cost(l) &= (BW/BW_{max})T(l)c + d \\ &= (f\alpha + l/BW_{max})c + d \end{aligned} \quad [2]$$

where α is the time for setting up the data transfer, c is the variable cost associated with the data transfer

(analogous to per Byte/Megabyte cost transfer), and d is the fixed cost for accessing the provider (analogous to a monthly access fee).

4.2.2. Single Provider – Multiple Consumers Model

In a single producer - multiple consumers model (Figure 6), data stored in a single provider is accessed by multiple consumers.

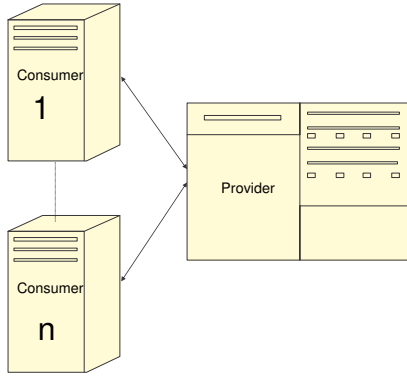


Figure 6. Single Provider – Multiple Consumers model

Consider the non-cooperative case, in which each consumer accesses the data from the single producer separately. Each of the n consumers accesses the producer with different bandwidth, and different costs are associated for each of the consumers. Assuming that all the consumers access the data simultaneously, and that the provider is able to support all the consumers at once, the total time taken to access the data is the maximum time taken by any consumer to access the data. The cost for accessing the data is the sum of the cost for each of the consumers accessing the partial data from each of the producers using a single provider – single consumer model.

In the cooperative case (overlap), in which one consumer accesses the data from the provider, and the remaining consumers access the data from this consumer. Assuming that there is overlap, wherein the consumers need not wait until the data is accessed by the consumers from the producer; the cost is the sum of costs for accessing all parts of the data. If there is partial overlap, the cost is the sum of accessing the parts which are located in another consumer and the cost of accessing the remaining parts from the provider. If there is perfect overlap, the

cost is the sum of accessing the parts which are located in another consumer.

$$\begin{aligned}
 \text{Time } (t) &= \max \left(\alpha_i + \frac{l_i}{BW_i} \right) \\
 \text{Cost } (t) &\neq \sum_{j=1}^n \sum_{i=1}^{\text{parts}} \left(\left(f_{ij} \alpha_j + \frac{l_j}{BW_{\max_j}} \right) c_j + d_j \right) \quad \text{separate streams} \\
 &= \sum_{i=1}^{\text{parts}} \left(\left(f_i \alpha + \frac{l_i}{BW_{\max}} \right) c + d \right) \quad \text{perfect overlap} \\
 &= \sum_{j=1}^n \sum_{i=1}^{\text{parts}} \left(\left(1 - R_i^j \right) \left(\left(f_{ij} \alpha_j + \frac{l_j}{BW_{\max_j}} \right) c_j + d_j \right) \right. \\
 &\quad \left. + R_i^j \left(\alpha_j + \frac{l_j}{BW_{\max_j}} \right) c_j + d_j \right) \quad \text{partial overlap} \\
 R_i^j &= 1 \text{ if Replica } i \text{ is located in consumer } j, 0 \text{ otherwise [3]}
 \end{aligned}$$

In order to minimize the total cost (as the case is usually), the cost of accessing the data from other consumers should be maximized (although this cost is an extremely negligible value). This is akin to increasing the number of parts in the other consumers, and accessing minimal amount of data from the provider directly.

4.3. QoS Broker

The QoS Broker obtains and stores the information about servers including but not limited to storage system latencies, bandwidth. The QoS Broker obtains the layouts from Server for the files that are available at the server. The Economic module has information on cost models for accessing storage, suggests optimization parameters to QoS Broker. The QoS Broker contains information about location of replicas in various servers. In a grid storage environment, the QoS Broker is designed to communicate with Metadata Catalog (MCAT) servers to obtain this information.

QoS Broker uses a progressive replica selection model. Initially QoS Broker selects the replica location with least cost (using the cooperative or non-cooperative consumer model as applicable) that satisfies the QoS requirement and replica created. When the QoS Broker discovers that the current replicas are insufficient to satisfy the QoS requirements, the replica location (from remaining list of replica locations) is used to create a new replica. The QoS Broker suggests to the application, the replica with least cost that would satisfy the QoS requirements.

4.4. Advantages of Active Agent Approach

There are several advantages to using the Active Agent approach.

- **Enhanced Performance:** In large grid environments, when multiple consumers are involved, selection of replicas becomes significant since the effective bandwidth of the individual replicas would be reduced. By choosing partial download of data (cooperative or non-cooperative model), the consumers are able to efficiently download the portion of data they require, making the data available with reduced latency.
- **Improved Failure Handling:** In the approach above, when one of the replicas is unavailable, based on a time-out value, the active-agent is able to make data available from an alternative location.
- **Cost reduction for producer as well as consumer:** When storage/network economics are considered, the consumer tries to minimize the cost of accessing the data, while the producer tries to minimize the cost of replication. However, in order to satisfy the data needs of a large data-oriented workflow system on computational grids, an efficient optimization strategy is expected by using this approach.

5. Relevant Work and Background

Replication techniques and economic models for grid environments have been explored, albeit as separate entities. The following are some of the strategies used.

5.1. Replica Placement

Replica placement has been studied for distributed computing systems as well as grid storage systems [7]. Kosar and Livny [8], have described a framework in which computational and data placement jobs are treated and scheduled differently by their corresponding schedulers, where the management and synchronization of both type of jobs is performed by higher level planners. Ranganathan [9] has analyzed when and where to place a replica in a decentralized fashion to meet availability goals.

In [8, 10] the authors have considered cost of availability for replica placement. Since there is usually a cost associated with dynamic replica deployment, client access patterns, replica failure patterns along with a required consistency level and a target level of availability are used to minimize replication cost.

5.2. Economics based file Replication

In [11], Bell, et.al. have developed an economic model for replica selection and dynamic replica optimization. Data files are “purchased by computing elements for running jobs and by storage elements to make an investment that will improve their expected future revenue. These files are sold by the storage elements to either computing elements or other storage elements. Computing elements try to minimize their file purchase cost, while storage elements attempt to maximize their profits.

5.3. Replica Placement as a Game

Geels and Kubiawicz [12], argue for replica management economies based on game theory. In their economic systems, individual machines are autonomous-free to choose which replicas they host. Auction based economies, mechanism design, computer sovereignty and federated environments are discussed.

6. Implementation Status

We have implemented a prototype of the proposed replication system model for Linux platform (Kernel version 2.6.12). Implemented is the Storage System Monitor. Emulations are performed using the Emulab – Network emulation test bed. The pNFS reference implementation developed at University of Michigan which is ported over PVFS2 file system [13] is used for this study. Parameters such as cpu speed, network latency and bandwidth are varied for emulating the grid computing and storage environments. Real-Time Application Interface (RTAI Linux) is used for providing real-time support for the applications, active agent architecture and storage subsystems.

7. Summary and Future Work

The distributed nature of grid computing allows applications to run across geographical boundaries. In order to support accessibility over wide areas data needs to be provided to applications with certain levels of QoS guarantees along with economic considerations related to transferring, storing and

accessing data. Grid storage environments at present either lack or provide minimalist QoS guarantees combined with economic issues. In order to extend traditional cluster storage systems or parallel file systems to grid storage, QoS guarantees along with economic considerations have to be incorporated. A fundamental framework for extending parallel file systems standard based on parallel Network File System (pNFS) and incorporating QoS guarantees combined with economic modeling has been presented in this paper.

Possible extensions include analyzing the proposed approach to work-flow systems, streaming media in content delivery networks with QoS issues, internet-based storage and file-sharing systems. Future studies could include the applicability of the proposed approach to caching mechanisms in distributed computing systems.

Acknowledgements

The authors would like to thank Arkady Kanevsky from Network Appliance, Inc. for providing valuable insights into this project.

References

- [1] O. Tatebe, Y. Morita, S. Matsuoka, N. Soda, and S. Sekiguchi, "Grid Datafarm Architecture for Petascale Data Intensive Computing," presented at 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2002), 2002.
- [2] C. Baru, R. Moore, A. Rajasekar, and M. Wan, "The SDSC Storage Resource Broker," presented at IBM Centre for Advanced Studies Conference (CASCON'98), Toronto, Canada, 1998.
- [3] D. Hildebrand and P. Honeyman, "Exporting Storage Systems in a Scalable Manner with pNFS," presented at 22nd IEEE - 13th NASA Goddard (MSST2005) Conference on Mass Storage Systems and Technologies, Monterey, CA, 2005.
- [4] G. Gibson and P. Corbett, "pNFS Problem Statement," in *Internet Draft*, 2004.
- [5] G. Gibson, B. Welch, G. Goodson, and P. Corbett, "Parallel NFS Requirements and Design Considerations," in *Internet Draft*, vol. 2004, 2004.
- [6] S. Shepler, B. Callaghan, D. Robinson, R. Thurlow, C. Beame, M. Eisler, and D. Noveck, "Network File System (NFS) version 4 Protocol," Network Working Group, The Internet Society April 2003.
- [7] H. Lamahamedi, Z. Shentu, B. Szymanski, and E. Deelman, "Simulation of Dynamic Data Replication Strategies in Data Grids," presented at 12th Heterogeneous Computing Workshop (HCW'03) in conjunction with IPDPS 2003, Nice, France, 2003.
- [8] T. Kosar and M. Livny, "A Framework for Reliable and Efficient Data Placement in Distributed Computing Systems," *Journal of Parallel and Distributed Computing*, 2005 (to appear).
- [9] K. Ranganathan, A. Iamnitchi, and I. Foster, "Improving Data Availability through Dynamic Model-Driven Replication in Large Peer-to-Peer Communities," presented at Global and Peer-to-Peer Computing in Large-Scale Distributed Systems Workshop, Berlin, Germany, 2002.
- [10] H. Yu and A. Vahdat, "Minimal replication cost for availability" in *Proceedings of the twenty-first annual symposium on Principles of distributed computing* Monterey, California ACM Press, 2002 pp. 98-107
- [11] W. H. Bell, D. G. Cameron, R. Carvajal-Schiaffino, A. P. Millar, K. Stockinger, and F. Zini, "Evaluation of an Economy-based File Replication Strategy for a Data Grid," presented at International Workshop on Agent based Cluster and Grid Computing at CCGrid 2003, Tokyo, Japan, 2003.
- [12] D. Geels and J. Kubiawicz, "Replica Management should be a Game," presented at SIGOPS European Workshop, 2002.
- [13] D. Hildebrand and P. Honeyman, "Scaling NFSv4 with Parallel File Systems," presented at Cluster Computing and Grid (CCGrid05), Cardiff, UK, 2005.