A Peer-to-Peer Infrastructure for Distributing Large Scientific Data Sets over Wide Area High-Performance Networks: Experimental Studies Using Wide Area Layer 2 Services

Yunhong Gu and Robert L. Grossman
National Center for Data Mining
University of Illinois at Chicago
www.ncdm.uic.edu

Joel Mambretti
National Center for Advanced Internet Research
Northwestern University
www.icair.org

ABSTRACT

This paper presents Sector, a distributed environment that was created specifically to address the challenges inherent in accessing, exploring, analyzing and transporting extremely large scientific datasets over high performance wide area networks. To date, the effective utilization of such datasets has been limited because accessing and transporting large remote data sets in traditional distributed computing environments is often a challenge. Sector was designed and developed to eliminate these barriers. The Sector architecture incorporates specialized communications services and specialized data services that are designed for high volume data flows over wide area high performance optical networks. This design employs several innovative techniques to ensure that data flows are maximized at all times and at all locations required regardless of distances. This architectural design has been implemented on a prototype international experimental testbed. The results and implications of these preliminary tests are described.

Categories and Subject Descriptors

C.2.4 [**Distributed Systems**]: Distributed Applications

General Terms

Management, Performance, Design

Keywords

Distributed Data Storage, High Performance Networks

GridNets 2007, October 17-19, 2007, Lyon, France
Copyright 2007 ICST 978-963-9799-07-3

1. INTRODUCTION

In this paper, we describe a distributed system called Sector that has been designed to enable scientists to access, explore and analyze very large scientific datasets. Sector was designed as a distributed, programmable platform that can effectively utilize extremely large data sets as a basis for analysis, decision making, and new knowledge discovery. Sector is a virtualized environment comprised of several key resource components, including Sector Clients, Sector Core Nodes, Sector Middleware, and high performance transport protocols and network services. Sector has been implemented on a large scale, international testbed. This paper also describes several recent experimental studies using Sector on this international testbed. One such experiment provided access to the Sloan Digital Sky Survey (SDSS) astronomy dataset to scientists in Moscow via the GLORIAD network [GLORIAD]. The results of these experiments are presented in the final sections of this paper.

Sector is a distributed environment comprised of several basic, integrated resource components. Sector Clients are edge processes and devices that utilize services from one or more Sector Core Nodes. Sector Core Nodes, are multi-processor computational clusters with large storage systems. All datasets are stored on Sector Core Nodes. These Nodes are dedicated exclusively to science projects requiring the management of large-scale data. All Sector Core Nodes are interconnected with high performance communication services. Sector can be implemented exclusively on any individual network service layer, e.g., layer 3, layer 2, or layer 1 services supported by wide area high performance optical networks. Also, it can be implemented to rely simultaneously on several layers using co-existent complementary services. The experiments described in this paper relied on wide-area layer 2 services.

Sector Clients also can be provided with access to Sector Core Nodes using any layer of network service, including high performance layer 2 networks. Sector Clients can also use the commodity Internet to access data from the Sector system. However, these clients will experience lower performance. In general, Sector is configured so that data managed by Sector is available to any Sector core node. Additional access policy restrictions can be applied through Sector Middleware.

The rest of the paper is organized as follows. Sector 2 describes the International network architecture for the Sector system.

Sector 3 discusses the software architecture in detail. Section 4 introduces the experimental studies that use Sector to transfer large scientific datasets. Sector 5 discusses the related work. Section 6 concludes the paper with a brief look at future work.

2. SECTOR NETWORK ARCHITECTURE

The unprecedented, and continuing, success of the Internet in communications is based in part on its capabilities for efficiently transmitting very large amounts of small pieces of information. However, its ability to manage extremely high volumes of data is a well known challenge. As the number and size of data sets required by applications increase, these restrictions are becoming increasingly problematic. The future Internet must be able to support the sharing of data, even the largest data sets, including large-scale multimedia objects. Furthermore, such communication services must be closely integrated with other distributed resources, such as compute clusters and storage devices, which are also optimized for supporting large scale data.

Sector takes advantage of several emerging trends that are shaping the future direction of digital communication services. The Sector design incorporates recent innovations in multi-layer network architecture. The traditional Internet provides for a single, undifferentiated "best effort" service, based on layer 3 packet routing. As a result, quality and performance vary highly in response to different network traffic conditions over time. Therefore, no real performance guarantees are provided. Although this approach has been highly successful for general Internet services, it has become increasingly problematic for more specialized requirements, such as those related to large-scale data and digital media. Transporting large-scale data streams over the commodity Internet can degrade performance for both those streams and for the lower volume streams sharing the same network resources. Consequently, network researchers have been developing new architectural frameworks and technologies that enable more fine grained, high performance differentiated services, not only at layer 3, but also at layers 1 and 2 [GN06].

In part these new techniques relay on enhanced abstraction levels for network services and subsequent reduced dependencies on specific communication infrastructure implementations. This approach allows for a high degree of separation between communication services and low level infrastructure. As a result, communication services can be more flexible, and they can be highly customized to meet a far wider range of requirements than the traditional Internet. Also, this increased level of abstraction enables even foundation communications infrastructure to become a programmable facility, or platform. To date, almost all Grid implementations have been based on communication services that have been used as external, non-deterministic resources. This new approach allows networks to participate in Grid environments as "first class," addressable resources.

Using this approach, Sector is able to provide large scale data flows that require particularly high performance with communications resources that are not available through the general Internet, including guaranteed high performance services on point-to-point paths among continents. As noted, Sector can be implemented with services based on any individual network layer or on any combination of layers. However, for the series of experiments described in this paper Sector was provisioned on an international experimental testbed using high performance layer 2 services exclusively. In part, this approach was used to examine the potential for path optimization by using end-to-end light paths over many thousands of miles.

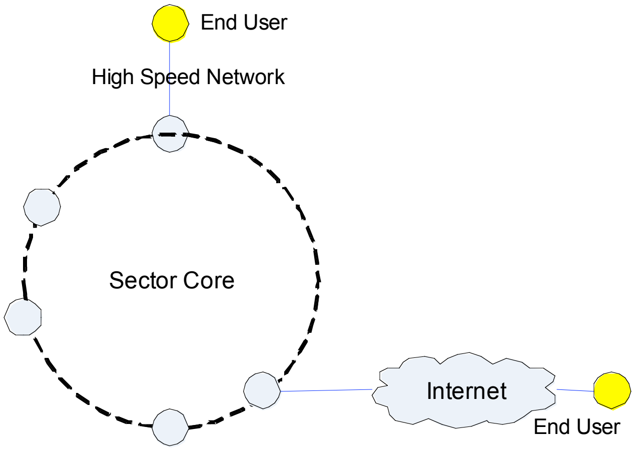

Figure 1. Network Illustration for Sector system.

Figure 1 illustrates the Sector network architecture. The Sector core consists of servers with high speed networks. This is an open network that any server can connect and contribute data and storage. End users within the high speed networks can access to the data served by Sector by connecting to any server in the Sector core. Meanwhile, users with a regular Internet connection can also access to the data with the help of UDT protocol.

3. SECTOR SOFTWARE ARCHITECTURE

3.1 Overview

As noted, Sector is a distributed infrastructure for accessing, exploring, analyzing, and communicating large scientific datasets. Sector is comprised of multiple resource components, including distributed computational clusters connected by lightpaths on high-speed optical networks. Usually these clusters are geographically distributed, and they are generally managed by different organizations. Sector can be implemented to allow these clusters and lightpaths to be dynamically allocated as required.

The Sector Core network consists of participating nodes running the Sector servers. These servers communicate using a P2P routing algorithm and do not rely on a centralized control node. The Sector Core automatically manages the joining and leaving of Sector server nodes, locates files and related services, and runs the services requested by the users.

The Sector Client is responsible for decision making and job scheduling for a particular application. This design was employed for two reasons. First, because the Sector Core is decentralized, it is very difficult for the Sector Core network to make optimal global decisions or to schedule optimally. Second, the client has specific knowledge about the application and thus it can make better decisions. This design principle leads to very simple server side software. This architecture follows the basic Internet principle of placing intelligence at the edge while maintaining simplicity in core resources.

Figure 2 shows the server side architecture. The distributed storage system is based on a P2P routing protocol for maintaining metadata. A file locating service and a file access service are the main services provided by the current version of Sector.

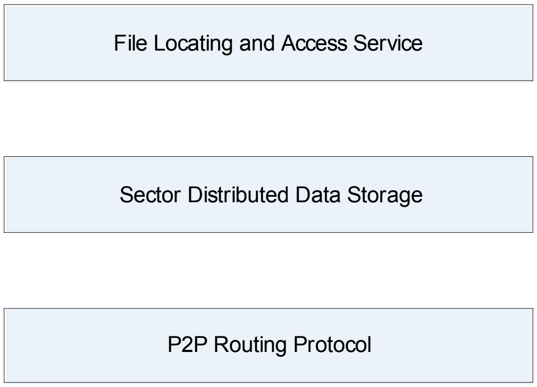
File Locating and Access Service
Sector Distributed Data Storage
P2P Routing Protocol

Figure2. Sector server architecture.

When users want to download a data file, they can start the Sector client software and connect it to any Sector server. The server will locate the metadata information first using a P2P routing protocol and then send it back to the client. The metadata includes information about the file (e.g., size, timestamp) and its locations. The client will then decide which copy it will download, will send a request to the server that has the copy, and then start downloading the copy. A more sophisticated client may download portions of the dataset concurrently from multiple servers.

3.2 Routing

The P2P routing protocol in Sector is responsible for maintaining the organization of the Sector core nodes, and it is also responsible for locating files in Sector networks. The current routing protocol used in Sector is Chord [SM+01], a P2P routing protocol based on a distributed hash (DHash). In Chord, each server is assigned a 128-bit ID by using a SHA digest on a string consisting of the server IP and server port. Note that because port information is also used, it is possible to start more than one Sector server on the same node. The servers are ordered by their IDs.

The files that stored in the Sector Core are also assigned a 128-bit ID according to the file names. The Chord protocol specifies that the metadata information of a particular file is stored on the server whose ID is greater than the one that is smaller or equal to the file ID. More information about Chord can be found in [SM+01].

The routing layer is an independent layer in Sector. Many types of routing algorithms can be used here, as long as they can provide a service that is able to locate files. In fact, a centralized routing server can also be used for this purpose. The reason that we chose a P2P protocol is due to the nature of non-centralized distributed resources.

3.3 Distributed Data Storage

Sector is not a native file system. Instead, it utilizes a storage system that relies on native file systems. Each Sector server specifies a local file directory that holds the required the required Sector files.

The Sector server maintains two file indexes. The local file index contains metadata information about all files stored in the local Sector directory. The remote file index contains metadata information about files on other servers. The remote file metadata that is stored is determined by the routing protocol as described above. Both indexes are periodically updated.

Once a request comes, the remote file index is checked and the file location information is returned. Note that there may be transient errors due to Sector nodes joining or leaving the Sector network. Such errors are inevitable, but Sector employs a strategy to keep them transient: the error will be fixed during the next time the index is checked and the user can submit the request again if the first try fails.

Sector uses a flat file name space. Because there is no directory structure, every Sector file must have a unique file name. Each Sector server has a local file management system to manage the file IO, file creation and deletion. Note that Sector does not split files into blocks so Sector files can be accessed outside the Sector network. For example, a user may manually copy a new file into the Sector directory.

Finally, Sector automatically maintains a specified minimum number of copies of a file in different locations, if sufficient additional disk space is available. In this way, Sector can serve as a long term repository for large scientific datasets.

3.4 Sector Services and Processes

For each Sector client, the Sector server starts the required service in order to run the client's request. A typical service requested is file access. In this case, the server starts a file service thread, which connects to the client via a UDT [GG06] connection. All the related commands and data are transferred using this connection. For example, if the user needs to download an entire file, it will send a specific command via UDT to the service thread. After receiving the request, the service thread, will start to transfer the file to the client.

The client cannot only read or write data files, but it can also request certain computations on the files. The service thread may run the specific computation (e.g., a SQL query on the data file) and transfer the result back to the client.

3.5 Security

Sector was designed to allow for flexibility in implementing levels of security. Sector can be implemented as a fairly open public system or as a highly secure environment, depending on the access policies selected. Sector employs one security model for those accessing data and another for those who write, update or upload data.

Data served by Sector can be open to public, in the same way that documents served by the web are open to the public. Also similar to general web services web, Sector allows the addition of transport layer security. However, Sector can also be implemented within a private network or within an organization as a highly secure closed system.

Write access to Sector Core is limited. While clients may download a file from any available server, someone wishing to

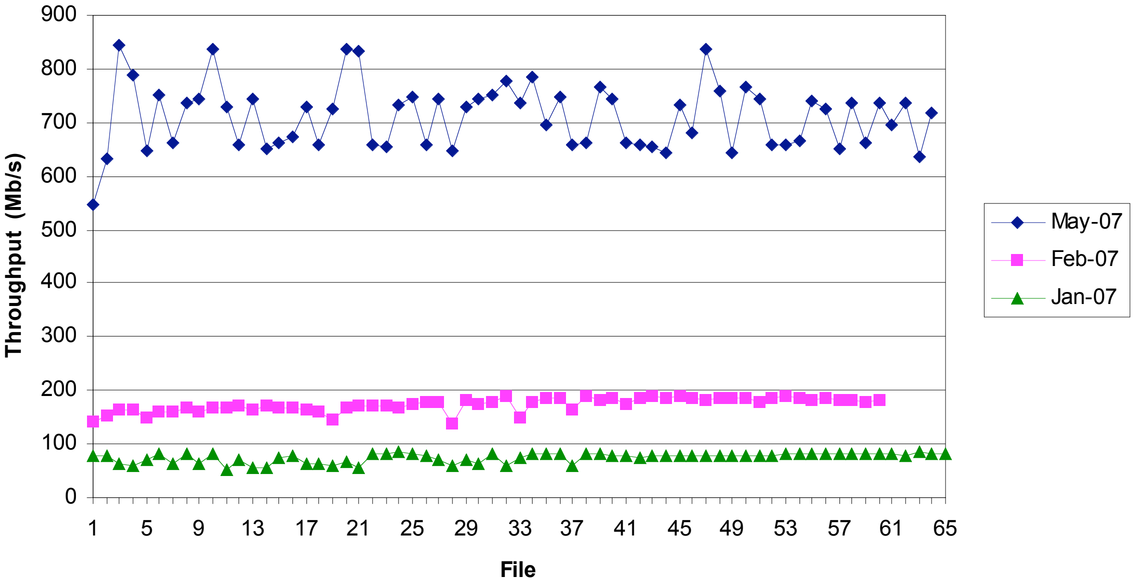

Figure 3: Downloading SDSS data to Moscow using Sector.

add data to the Sector network must have an account on a specific server that supports uploading of Sector files. Further, uploads are restricted to computers whose IP addresses have been added to the allow list of that server. The allow list can be configured by the system administrator who owns the server. All such tasks can be completely governed by any additional use policies required.

4. EXPERIMENTAL STUDIES

We have been using Sector to store and distribute Sloan Digital Sky Survey (SDSS) data [SDSS] over Teraflow Network [TFN]. We have developed a SDSS downloading tool that automatically chooses the nearest SDSS Sector server to download the specified files. The system has been online since July 2006. During the last 8 months, we have had about 3000 system accesses and a total of 100TB of data has been transferred to end users.

Recently, we set up a layer 2 path service from Chicago to Moscow and tested Sector on this network. This path service connects the StarLight international communications exchange facility in Chicago and Geophysical Center at the Space Research Institute, Russian Academy of Sciences (GC) in Moscow via the GLORIAD network [GLORIAD]. Three 1Gb/s lambdas are provisioned with the 10Gb/s WAN. A scientist from GC used Sector to download 1.34TB compressed SDSS DR5 catalog, consisting of 64 files. Figure 2 shows the performance per file. The bottleneck was the disk performance, while network bandwidth was 1Gb/s (only one pair of machines was used). Previously as a comparison trial, GC also downloaded the same data twice via the public Internet. See Figure 3 for a comparison of these three transfers of the SDSS DR5 catalog data.

The downloading trial using the public Internet in January 2007 and February 2007 achieved a maximum performance of 75Mb/s and 180Mb/s respectively. The recent data transfer in May 2007 via GLORIAD achieved more than 700Mb/s average throughput, with peak speed at 844 Mb/s.

Sector also supports parallel data access, which greatly improves data I/O because multiple Sector servers can be accessed concurrently. During SC 2006, we demonstrated downloading the same SDSS dataset from Chicago to Tampa from six Sector servers concurrently. In this way, we achieved an average performance of over 8Gb/s.

5. RELATED WORK

As muli-layer communication services based on high performance inexpensive optical networks spread, the Internet will be able to support high performance distributed data storage and exploration at a global level. Sector serves as a prototype to explore this vision. The Sector initiative described here complements many other advanced distributed infrastructure research projects.

Such systems have been developed in large enterprises, such as Google's GFS [GGL03], MapReduce [DS04], and BigTable [CD06], and Amazon's S3. Compared to Sector, these systems use dedicated distributed clusters, while Sector does not require dedicated hardware. On major reason is that these systems are managed by single organization, while Sector encourages collaboration among multiple organizations.

To the end, Sector is also related to Grid systems including Globus [Globus] and Condor [Condor]. However, grid systems were designed to aggregate computing resources rather than data bandwidth, because network bandwidth in the last decade was still scarce compared to the system bus. The Grid architecture incorporates considerations of desktop systems as well as high-end computational clusters. In contrast, currently Sector is being designed to be used only with high end systems and clusters configured with large storage and high performance network connections. Although the Sector servers do not necessarily have to be dedicated, they are assumed to have to have high end performance capability.

Recently, as P2P technology becomes popular, several P2P based Internet storage systems have been proposed (e.g., PAST [RD01] and OceanStore [KB+00]). Such P2P storage systems rely on extensive duplication of information in order to cope with unstable user nodes and network connections. The target use scenario is quite different that is shaping the design of Sector.

IBP [BMP02] is a centralized Internet storage that uses a data structure similar to the UNIX inode file system. IBP shares a similar goal to Sector of providing a platform for distributing large datasets. On the other hand, IBP organizes manages storage at the block level, while storage in Sector is managed at the file level. Files are not further split, except of course by the underling file system.

Many large scale optical networking research testbed projects have been established, in part to assist in address the requirements of managing data intensive applications in distributed environments. A number of innovations are beginning to emerge from several of these testbeds that are creating new methods for providing layer 1 and layer 2 services on advanced optical networks, including those using dynamic lightpath provisioning. A recent report provides a description of many of these advanced testbeds, including OMNInet, StarPlane, OptIPuter, EnLIGHTened, Phosphorus, UltraScience Net, DRAGON, and Viola. [JM2007] Sector is being designed to take advantage of the innovations in dynamic network resource provisioning and reconfiguration that are emerging from these testbeds.

6. CONCLUSION AND FUTURE WORK

The Sector architecture used in the experiments described here as well as in other research trials has demonstrated preliminary positive results. Future research will focus on extending Sector functionality, especially with regard to enhancing abstraction layers for implementing and monitoring processes, protocol design and development, and resource integration, with a special focus on integrating large scale high performance services based on global layer 2 services and dynamic lightpath provisioning. Future research will utilize additional large scale science datasets and it will be conducted on a wide range of international testbeds.

Sector is an experimental distributed environment that was designed to meet the needs of managing extremely large scientific datasets distributed across the world. Currently, managing such datasets has been restricted because of the limitations of traditional distributed architecture and technology at multiple levels. Sector is being developed not only to eliminate such restrictions for large datasets but also to provide new functionality not possible today. The Sector architecture places particular emphasis on integrating resources with specialized communication services implemented on international high performance optical networks. Initial implementations on prototype international experimental testbed have been positive. Further Sector developments and testbed testing are being planned.

7. ACKNOWLEDGMENTS

The authors would like to thank the many international partners who have supported the implementation of the global paths that made possible the Chicago-Moscow experiments described in this paper, including Bill St. Arnaud, Pieter de Boer, Michael Boyarsky, Natalia Bulashova, Maxine Brown, Greg Cole, Tom DeFanti, Freek Dijkstra, Bos Erik-jan, Rene Hatem, Wouter Huisman, Alexander Y Ilin, Paola Grosso, Veniamin Konoplev, Anton Korotin, Veniamin Konoplev, Cees de Laat, John Lankford, Ravil Nazirov, Kees Neggers, Platonov Alexey Pavlovich, Bram Peeters, Alexei Soldatov, Olga Starostina, Thomas Tam, JP Velders, Alan Verlo, Linda Winkler, and Mikhail Zhizhin.

This work was supported in part by the National Science Foundation through grants SCI-0430781, CNS-0420847, and ACI-0325013.

8. REFERENCES

[CD06] Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, and Robert E. Gruber, Bigtable: A Distributed Storage System for Structured Data, OSDI'06: Seventh Symposium on Operating System Design and Implementation, Seattle, WA, November, 2006.

[Condor] Condor. http://www.cs.wisc.edu/condor/, retrieved on Jan 19, 2007.

[DS04] Jeffrey Dean and Sanjay Ghemawat, MapReduce: Simplified Data Processing on Large Clusters, OSDI'04: Sixth Symposium on Operating System Design and Implementation, San Francisco, CA, December, 2004.

[GG06] Yunhong Gu and Robert Grossman, UDT: UDP-based data transfer for high-speed networks, Computer Networks (Elsevier). Volume 51, Issue 7. May 2007.

[GGL03] Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung, "The Google File System", pub. 19th ACM Symposium on Operating Systems Principles, Lake George, NY, October, 2003.

[Globus] Globus Toolkit. http://www.globus.org/toolkit/, retrieved on Jan 19, 2007.

[Gloriad] Global Ring Network for Advanced Applications Development, http://www.gloriad.org, retrieved in May 2007.

[GN06] Franco Travostino, Joe Mambretti, Gigi Karmous-Edwards, eds, "Grid Networks: Enabling Grids with Advanced Communication Technology," Wiley, 2006

[BMP02] Micah Beck, Terry Moore, James S. Plank. An End-to-End Approach to Globally Scalable Network Storage, ACM SIGCOMM 2002, Pittsburgh, PA, USA, August, 2002.

[JM07] Joe Mambretti, Tomonori Aoyama "Report of the Interagency Optical Networking Testbed Workshop 3," Networking and Information Technology Research and Development's Large Scale Network Coordinating Group, May 2007.

[KB+00] Kubiatowicz J, Bindel D, Chen Y, Czerwinski S, Eaton P, Geels D, Gummadi R, Rhea S, Weatherspoon H, Weimer W, Wells C, Zhao B. 2000. OceanStore: An architecture for global -

scale persistent store. In: Proceedings of ASPLOS'2000; 2000; November; Cambridge, MA; Pages 190-201.

[RD01] Rowstron A, Druschel P. 2001. Storage management and caching in PAST, a large-scale, persistent peer-to-peer storage utility. In: Proceedings of ACM SOSP'01; 2001; October; Banff, Canada; Pages 188-201.

[SDSS] Transporting SDSS data using Sector and UDT, http://sdss.ncdm.uic.edu, retrieved in May 2007.

[SM+01] Stoica I, Morris R, Karger D, Kaashoek M. F, Balakrishnan H. 2001. Chord: A scalable peer-to-peer lookup service for Internet applications. In: Proceedings of ACM SIGCOMM'01; 2001; August; San Diego, CA; Pages 149-160.

[TFN] Teraflow Network, http://www.teraflownetworktestbed.net, retrieved in May 2007.