Data Reservoir: Utilization of Multi-Gigabit Backbone Network for Data-Intensive Research

Kei Hiraki, Mary Inaba, Junji Tamatsukuri
University of Tokyo
7-3-1, Hongo, Bunkyo-ku, Tokyo, Japan
{hiraki, mary, junji}@is.s.u-tokyo.ac.jp

Ryutaro Kurusu, Yukichi Ikuta
Fujitsu Program Laboratories
1313, Miwata, Nagano-shi, Nagano, Japan
{ryu, ikuta}@fpl.fujitsu.com

Hisashi Koga, Akira Zinzaki
Fujitsu Laboratories
4-1-1, Odanaka, Nakahara-ku
Kawasaki, Kanagawa,Japan
{koga, zinzin}@flab.fujitsu.com

Abstract

We propose data sharing facility for data intensive scientific research, "Data Reservoir"; which is optimized to transfer huge amount of data files between distant places fully utilizing multi-gigabit backbone network. In addition, "Data Reservoir" can be used as an ordinary UNIX server in local network without any modification of server softwares. We use low-level protocol and hierarchical striping to realize (1) separation of bulk data transfer and local accesses by caching, (2) file-system transparency, i.e. interoperable whatever in higher layer than disk driver, including file system. (3) scalability for network and storage.

This paper shows our design, implementation using iSCSI protocol[1] and their performances for both 1Gbps model in the real network and 10Gbps model in our laboratory.

1. Introduction

Computing systems have been played a big role for progress or advancement of scientific research from the very beginning of its history. EDSAC, the first stored-program computer was used for numerical computation in chemistry, meteorology, and radio astronomy[2]. As the computation power grows by, successive supercomputers are used for numerical computation for scientific simulation, such as CFD, FEM, QCD, MD, or First-Principle, ab-initio methods. And now, besides numerical computation as above, "Data Intensive Computation" has come, which is, in short, to mine or find out scientific truth from huge amount of data by executing computational analysis. High energy accelerators, huge telescopes, scientific satellites, large electron microscopes and earth observation equipments are examples of huge data supplier. Size of available data is increasing because of the rapid improvement of precision or downsizing of detectors and growing capacity of storage devices. For example, the SUBARU optical-infrared telescope in Hawaii now generates 10G bytes data every day with 0.5GB/sec on peak time. And, for these kinds of researches, it's quite popular that many research groups share huge observation facility or observed data in collaboration. On the other hand, it's not so easy for ordinary computers to transfer a single huge file with the speed over 300Mbps locally and 100Mbps globally. As the result, even now, physical transfer of DLT tapes by automobiles or planes is performed.

Now, domestic high-speed backbone network has been prepared and inter-continent networks are also being prepared. Hence, general scheme of seamless handling of huge data via high-speed backbone network is eagerly awaited, which is the key to achieve data intensive researches.

The objective of our research is to propose a general, non-project-specific, file sharing facility to support data intensive scientific research projects, spread in distant locations but connected each other by high-speed backbone networks. Our target is a simple distributed shared file system that has scalability in (1) network bandwidth, (2) file size and (3) latency of global networking with a reliable protection mechanism, and can be also used as an ordinary server such as NFS, SAMBA, FTP, or HTTP server without any modification.

This paper describes "Data Reservoir" system[3, 11, 12,

0-7695-1524-X/02 $17.00 (c) 2002 IEEE

Project	Group Leader	Domestic Connection	International Connection	Data Size
High Energy Polarimeter SMART	H. Sakai	RIKEN, RCNP	CERN, BrookHaven	50 DLT/month
Radio Telescope (VLBI)	Y. Sofue	Nobeyama Radio Observatory	Max Plank Observatory	VLBI data: 200 GB
Slone Digital Sky Survey	S. Okamura	National Astronomical Observatory	Fermi Lab	Survey Data: 10TB
Satellite observation of early universe	K. Makishima	ISAS, Hiroshima U.	NASA, European Space Agency	Current Satellite: 1GB/day
Simulation of Global Change	T. Yamagata	Frontier Research System for Global Change	N/A	10TB per 1 time simulation. Current data archive: 50 PBytes
JC ATRAC Experiment	T. Kobayashi	KEK, Kyoto U.	CERN	CERN LHC: 100MB/sec
Infra-red observation Satellite	T. Onaka	IRIS, Nagoya U	ESA receiving site (Sweden)	Downlink :200MB, Data exchange within a minutes
Astronomical Simulation by GRAPE-6	J. Makino	National Astronomical Observatory	Princeton U., Indiana U.	100MB/s, 10TB per 1 time simulation
KEK b-factory	H. Aihara	KEK, Nagoya U	Princeton U.	Raw Data:600GB/day, Data exchange 10GB/day
SUBARU telescope	K. Shimasaku and S. Okamura	National Astronomical Observatory	Hawaii Observatory	100GB/day, peak BW 4GB/sec

Table 1. Research Projects with Data Reservoir

13, 14, 15]; a novel approach for multi-gigabit file sharing facility using low-level protocol and hierarchical disk striping technique for rapid data transfer, system scalability, and file-system transparency. We show outline of our projects in Section 2. The basic design and implementation of the Data Reservoir is described in Section 3 Then experimental result, and comparison with other systems is shown in Section 4 and Section 5, and we conclude this work in Section 6.

2. Our Scientific Research Projects and Network Environment

With the rapid progress of network technology such as WDM optical network and high-speed switching technology, multi-gigabit backbone networks are being arranged for scientific research projects in Japan.

In the beginning of 2002, Super-SINET (Super Science Information Network [4]) starts to connect universities and research institutes in Japan. Currently Super-SINET covers about half area of Japan(Figure 1 thick lines), and in 2003, will be expanded to cover almost whole of Japan (Figure 1 thin lines). Super-SINET consists of 10Gbps global IP network and bunches of 1 Gbps project lines. Main objective of Super-SINET is to support scientific research projects as astronomical observation, seismic expectation, genome informatics, and high energy physics experiments.

Table 1 shows our partner project groups in Faculty of Science, the University of Tokyo. They share huge data with observatories or institutes in distant places such as the

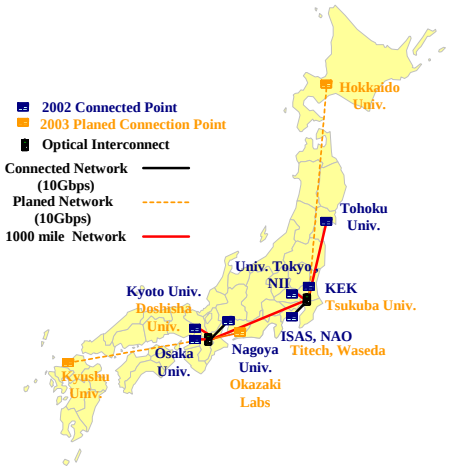

Figure 1. Super-SINET network in Japan.

High Energy Accelerator Research Organization or the Institute of Space and Astronomical Science. Note that most groups also strongly eager to share data with oversea observatories such as accelerator of Brook Haven or CERN in Europe, or SUBARU telescope in Hawaii.

However, given long distance high-speed network, it is not easy for scientific researchers to utilize it well. Technically, control of transfer window and buffer management for large-latency high-speed network, bandwidth bottleneck at disk drives, I/O bus, and memory bus, overheads of an operating system and run time libraries, and protection are major difficulties that prevent high speed data transfer for long distance. Our System aims to conceal these difficulties

from end-users' eyes.

3 Our Design

3.1 DSF(Distributed Shared File) Architecture

We propose Distributed Shared File(DSF) architecture to construct a general, non-project-specific, data sharing system for data intensive research.

From the view point of users, desirable system is to make users feel as if they had their observation equipments in their own local area network without any additional operation. In most cases, location of observatories, i.e., data-suppliers, are fixed as one or a few places by each project, and, location for analysis such as universities or research centers, i.e., data-consumers, are tend to be also fixed by each project for some time period. And, we assume these observatories and universities or research centers are being connected each other by high-speed project line.

We concentrate on the case that one or a few data-supplier, such as telescope or high energy accelerator, and several data-consumers, such as universities or institutes, are spread in limitted number of distant locations but connected each other with very high speed network, such as Super-SINET. We mainly consider data intensive research projects but this scheme can be also applied to companies which have high speed intranet infrastructure inside, treate big data, and have demands for disaster recovery. Technically, interesting point of huge data transfer on over gigabit netowork is that we are just on the balancing point; for local transfer, main bottle neck is disk I/O or I/O bus speed, in turn, for long distance transfer, the bottle neck is network, especially when TCP/IP is used, where network latency has a big effect on its performance.

In this setting, system should fulfill the following requirements.

- separation of high bandwidth high latency long distance communication and local communication
- use of standard API from user program
- scalability in both network bandwidth and storage size

We propose a system of Distributed Shared File(DSF) architecture which separately treats local file access by caching and long distance bulk data transfer internally.

Similar to software DSM(Distributed Shared Memory), DSF architecture coalesces local file accesses and global coarse-grained bulk data transfer. For bulk data transfer, an important thing is to fully utilize network bandwidth. Network bandwidth utilization by multiple streams and disk I/O speed up by striped files over several disks are now rather orthodox methods in either high-speed file transfer software

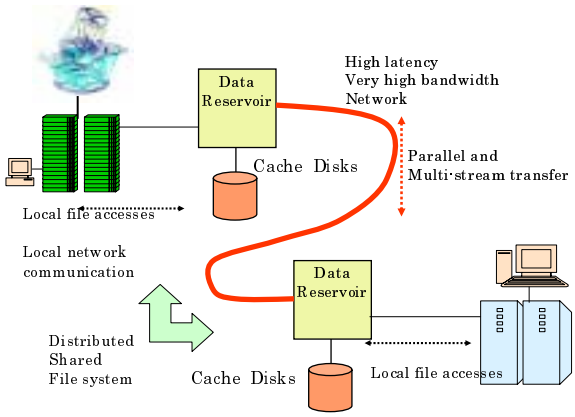

Figure 2. DSF Architecture

such as bbftp[5], sfcp[6] and GridFTP[7] or global network file systems such as AFS[8], DFS[9] or CODA[10]. For local access, the most imporatnt thing is user's convenience, in other words, interoperability or file system transparency.

We adopt low level protocol to realize multi-stream and data striping, so that block level data transfer is available on each disk, and direct transfer from storage device to storage device in distant locations can be easily done. Using low-level protocol also enables interoperability to user-programs and operating systems, which we call "filesystem transparency", and gives a single file image for local data transfer, so, existing software not only client systems, but also NFS server, SAMBA, FTP or HTTP daemon, can be used without any modification.

A combination of several computer nodes and a network switch forms a node of DSF system. We call it "Data Reservoir". A pair of Data Reservoir systems are placed in the entrances of the local area networks, and, connects local area network and very high speed network.

A Data Reservoir system consists of "*file servers*" and "*disk servers*". File servers are computer nodes to provide services for cliant computers in local area network. Disk servers are intermediate computer nodes with either SCSI disks or lower level computer nodes. Disk server gives us more flexibility on scalability and security issue such as SSH like authentification or packet filtering.

For local access, file servers treates disk servers as their own local disks, and, data are supplied to client machines in local area network by existing server software, such as NFS, SAMBA, or HTTP servers. Once long distance burst transfer starts, disk servers start data transfer by themselves to corresponding disk servers placed in distant location, so that multi-stream data transfer is performed(Figure 3).

To realize this scheme, we use iSCSI as low level protocol. which has high similality to conventional SCSI protocol, and is already a standard. For speed and scalability, we

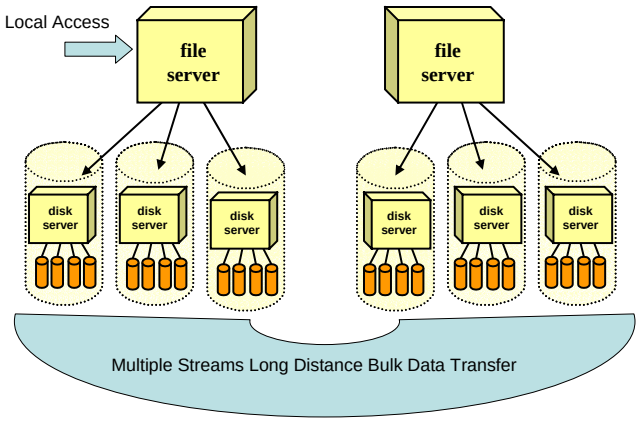

Figure 3. Data Reservoir System

take hierarchical striping.

3.2 iSCSI — Low level protocol

We adopt low level protocol iSCSI[1] so that

1. Block level data transfer is available on each disk,
2. Direct transfer from storage device to storage device can be easily done
3. System is interoperable for both user programs and operating system, that is, any file system or higher level system software, or user application can work without any modification.

SCSI(Small Computer Systems Interface) is popularly used protocol for communication to I/O devices. Using SCSI, in many cases, a computer communicates with several I/O devices, but, sometimes a few computers share I/O devices or communication between I/O device and I/O device is executed. In the terms of SCSI, we call an *initiator* that makes I/O requests and we call the specified device *target* that responses to the requests to proceed.

iSCSI (intenet SCSI) protocol is a transport protocol for SCSI that operates on internet, and is already a standard. As for storage device, we use computer nodes with several hard disk drives, disk server, instead of using iSCSI disks, so that storage device can play the role of both target while local access and initiator while burst data transfer.

3.3 System and Implemenatation

Data Reservoir System consists of file servers and disk servers. For local access, a file server works as an initiator, and, disk server works as an target.

For long distance transfer, file server is quiet, and, source disk server works as an initiator and distination disk server works as a target.

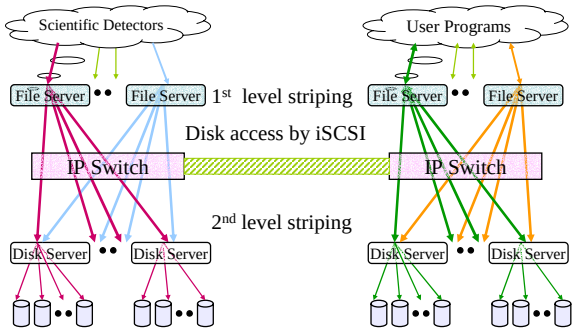

Figure 4. Local Data Transfer

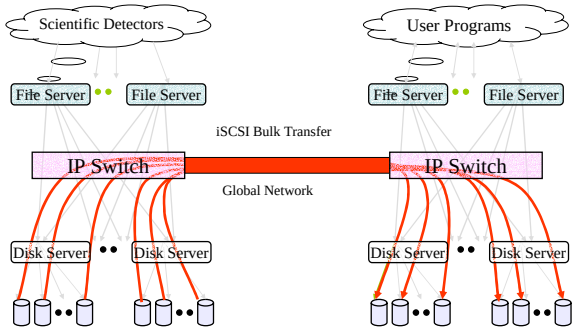

Figure 5. Burst Data Transfer

We implement softwares on Linux 2.4.18. For initiator, we implement iSCSI driver. When read/write system call is called, iSCSI driver gets SCSI command block from SCSI driver and sends PDUs (iSCSI Protocol Data Units) to a port of a target machine. For target, we implement iSCSI kernel daemon. Each target has two types of ports, one is for playing the role of RAID device for local access, and the other is for the role of raw disk device, for burst transfer.

To tackle with disk I/O bottle neck and to maintain scalability, we use hierarchical striping using disk servers.

A file servers recognize disk servers as its local disks, and, by using RAID 0 of Linux software RAID, first level striping is proceeded. Each disk server has several local disks, and, when disk server gets I/O request, disk server again stripes data, and, store or retrieve striped data.

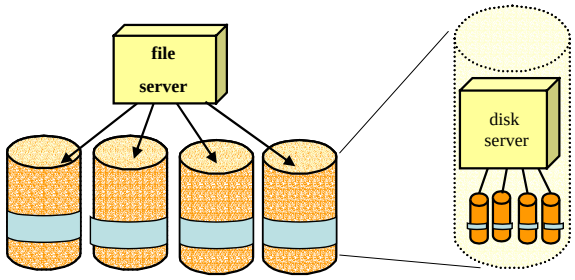

Figure 6. Hierarchical Striping

Figure 6 shows how striping is done. Each computer nodes, either file server or disk server, cares its local striping independently.

Figure 7 shows software of disk server, and Figure 8 shows that of file server in case that file server is used as an NFS server with EXT2 local file system. Yellow text boxes in these figures indicate existing softwares, and it shows any application and many system software can work without any modification.

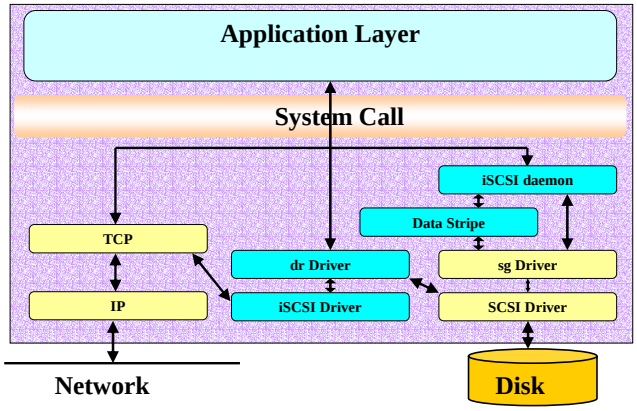

Figure 7. Software for Disk Server

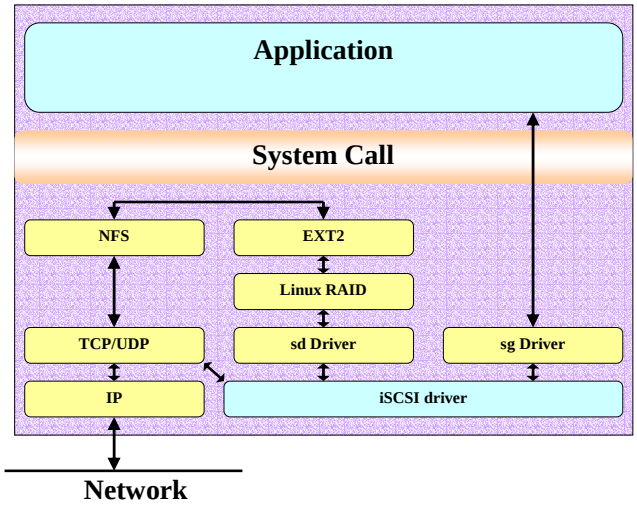

Figure 8. Software for File Server

To omit the unused data transfer, we utilize Data Block Bit Map of EXT2. As for security, we use ssh with RSA for the communication for exchanging information of partition or other settings. For more implemenation details or tunings around iSCSI see [16].

4. Evaluation

In this section, we show the experimental result of

1. 1Gbps model basic evaluation,
2. 1Gbps model connected by 1Gbps Super-SINET between 25 mile(40km)distance for real usage
3. 1Gbps model for 1000 mile(1600km) round trip in Japan, and
4. 10Gbps model basic evaluation.

All the performance figures described in this section is the average data transfer rate calculated from the amount of the size of transfered files. Therefore, physical network transfer rate is slightly faster than data shown in tables because of TCP/IP or iSCSI header.

For network analysis, we use Sniffer-PRO and SmartBits network analyzer.

4.1 Basic Experiment of 1Gbps model

Basic 1Gbps model consists of 5 IA servers; a file server and 4 disk servers, (DELL Power Edge 1650, Pentium III 1.4 GHz, dual CPU, 1GBmemory, Linux 2.4.18, a 36GB 10,000 rpm system disk, 4 of 73GB 10,000rpm Ultra3 SCSI hard disk in PowerVault for disk servers, and Netgear GA620 1000BASE-SX NIC) and a gigabit ethernet switch(Summit 5i) for 1000BASE-SX optical fiber connections.

From now, we denote $i \times j \times k$ DR system a Data Reservoir system of i file servers, j disk servers for each file server, and k disks for each disk server. And when we use 2nd SCSI bus, we denote $i \times j \times (k_1 + k_2)$ where k_1 and k_2 are the numbers of disks for each SCSI bus. The basic 1Gbps model above is denoted as $1 \times 4 \times 4$.

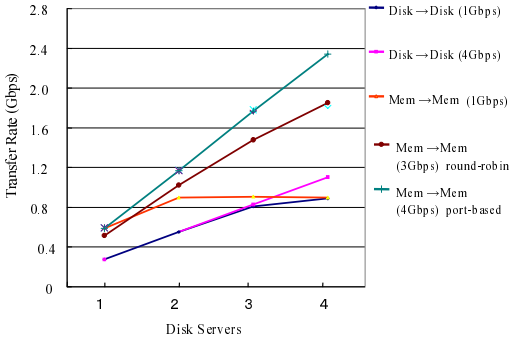

Figure 9. Basic Performance.

Figure 9 plots the basic ability of data transfer without file system overhead. Each disk server is treated as a raw device of its file server and we observe the relationship between the transfer rate and the number of streams, i.e., the number of raw devices. We examined disk to disk raw device copy and memory to memory copy on 1 Gbps and

4Gbps network, where 4Gbps connection is realized by port trunking of the switch with round-robin distribution. Figure 9 shows that transfer rate for raw device disks are proportional to the number of streams until network is saturated.

Since we check and exchange the information of Data Block Bit Map between file server and disk server, burst data transfer is efficiently performed when the usage of the partation is high.

File Size	32GB	200GB	533GB
Usage	6%	37%	100 %
time	4m58s	21m51s	54m27s
rate	0.85Gbps	1.22Gbps	1.31Gbps

Table 2. File Size and transfer rate

Table 2 shows that as far as usage of partition is more than about 40%, performance drop is almost negligible, but, usage of partition is less than 10 %, file system overhead causes performance drop.

We also examine the peformance of the Data Reservoir system as an NFS file server in local area network. We found that from the view point of NFS client, there seems no difference either the server has local disk or iSCSI disk.

4.2 1Gbps for middle distance(25mile)

A pair of Data Reservoir System is installed at the Institute of Space and Astronomical Science (ISAS) and Univ. of Tokyo, whose distance is about 25 mile(40km), using 1Gbps Gigabit Ethernet on private project line of Super-SINET. Latency is 1.7 msec for one way, and 3.5msec for round trip. No transmission error except jaber is observed during the experiment. While data transfer, max 98.7% of 1Gbps bandwidth was utilized, and result of data transfer was same as 1Gbps experiment as above. Hence, large data transfer for 25 mile distance, network bandwidth for multiple streams or local I/O speed for a single stream is still the bottle necks, and, latency of the network does not effect much.

protocol	bandwidth(Mbps)	configuration
iSCSI	337.52	8 iSCSI queue, window size optimized
ftp	222.08	window size optimized
ftp	93.12	default setting

Table 3. Performance comparison of ftp and iSCSI

We also make a comparison between ftp and iSCSI. Table 4.2 shows data bandwidth comparison of ftp(1GB file transfer) and iSCSI. From Table 4.2, sequential data transfer by iSCSI protocol is about 50% faster than data transfer by ftp whose window size is optimized for long distance data transfer.

4.3 1Gbps, 1000mile(1600km) round trip on low quality network

Another pair of 1Gbps model was settled up at Univ. of Tokyo. We made 1000 mile(1600 km) round trip circuit using Gigabit Ethernet on Super-SINET; the circuit connection is, Data Reservoir A – Univ. of Tokyo(C6509, Cisco) – Kyoto Univ.(fiber patch) – Osaka Univ.(Black Diamond, Extreme) – Tohoku Univ.(fiber patch) – Univ. of Tokyo(C6509) – Data Reservoir B.

The round-trip latency of the network is 36msec. And, 950Mbps is the maximum utilization of the bandwidth which we check by sending ether frame using SmartBits. Since the basic 1Gbps DR model $1 \times 4 \times 4$ attains only 812Mbps speed, slower than results above, we check effects of the number of disk servers and the number of disks for each disk server.

$1 \times x \times y$	4, 6+2	4, 2+2	4, 4	2, 6+2	2, 4	2, 2+2
Rate(Mbps)	870	824	816	736	704	696
Rate (%)	91	87	86	77	74	73
Streams	32	16	16	16	8	8

Table 4. Transfer Rate and Composition

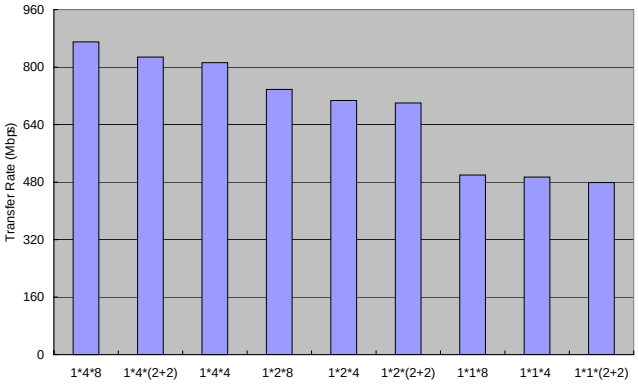

Figure 10. Composition examined on 1000 mile circuit

Here, as the number of streams incrases, transfer rate grows, however, expected linearity of the number of disk servers and transfer rate disappears. This may be caused by processors of disk servers can have some rest because of network latency.

4.4 10Gbps model, on 8Gbps

10Gbps model consists of 30 IA servers; 6 file servers and 24 disk servers, $6 \times 24 \times 2$, (DELL Power Edge 1650, Pentium III 1.4 GHz, dual CPU, 1GBmemory, Linux 2.4.18, a 36GB 10,000 rpm system disk, 2 of 73GB 10,000rpm Ultra3 SCSI hard disk for disk servers, and Netgear GA620 1000BASE-SX NIC) and a 10Gbps Ethernet switch.

Unfortunately, real 10Gbps ethernet switch based on IEEE802.3ae is not yet available in Japan, we examined Foundry Big Iron 8000 with 10Gbps ethernet module whose internal bus speed is 8Gbps max, and RiverStone RS 16000 (12 ports and 8 ports) Extreme BlackDiamond 6808 (8 ports), for which, 10Gbps ethernet port is not available, and we use trunking function of the switch.

Table 5 shows the performance of file transfer between two 10 G models of 777.24GB data. In the table, FSpeed denotes file transfer speed, and NSpeed denotes throughput of the network including headers.

	RS(12)	BigIron	B-D(8)	RS(8)
Time	12m09s	14m56s	15m00s	15m47s
FSpeed	8.51Gbps	6.94Gbps	6.91Gbps	6.56Gbps
NSpeed	9.13Gbps	7.31Gbps	7.27Gbps	6.90Gbps

Table 5. Switch and Transfer Rate

According to our measurements, bandwidth loss by trunking is about 5 to 10%. Here, 300GB files are transferred. According to Table 5, 6.9Gbps file transfer rate on 8Gbps network was attained with 24 disk servers.

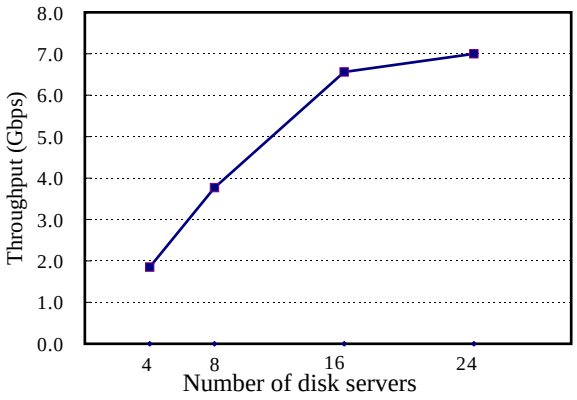

Figure 11. Transfer rate and number of disk servers

Figure 11 shows the relationship between the number of disk servers and attained performance.

There are several issues to further increase performance. (1) load balancing and scheduling of disk servers. Currently, scheduling is determined by the balance of disk access speed and port scheduling of the Ethernet switch. We observed about 3% performance loss due to load imbalance between disk servers. (2) number of disk server can be reduced to 20 for this configuration.

Figure 12. Photo of 10G model experiments

According to the results of experiments,

(1) Number of TCP connections is the dominatinc factor for the bandwidth in high-latency situation,

(2) Data Reservoir can efficiently transfer files under high-error-rate long-distance networks, and

(3) Sequential low-level data transfer is about 50% faster than ftp transfer, i.e. the use of sequential low-level protocol can reduce the required number of disks for the given bandwidth by 33%.

5. Related Works

Figure 13 shows mutual relation of existing data transfer facilities and Data Reservoir. Data transfer between two distant location can be achieved by either high-level, mid-level or low-level protocols. Here, high-level protocol means file transfer protocols over application program such as ftp and its extension programs with grid middleware, mid-level protocol means parallel and redundant file system file access level protocol over NFS or its successors, and low-level protocol means block level access protocols on storage devices such as iSCSI and iFC.

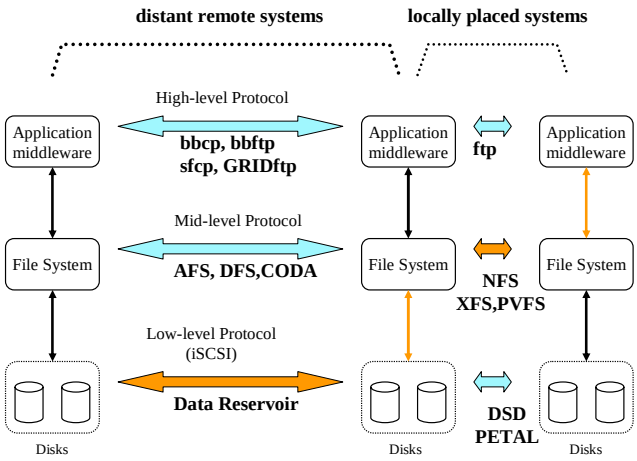

Figure 13. Related Works.

Petal[20] and DSD[21] are distributed file system that utilize low-level protocol. Using these systems, distributed file systems are achieved by virtualizing disk volumes within a cluster and realizing a storage area network on ATM or IP networks. However, these system can operate within a local cluster and cannot realize global distributed shared file system with caching capability through high-latency, high-bandwidth networks.

As for mid-level protocol approach, there are several extension of NFS to accommodate with global file sharing. AFS[8], DFS[9] and CODA[10] are examples of these file system. They have transparent interface to user programs but their performance cannot be scalable because it is very difficult to parallelize a file server and they suffer common overheads of file systems in global communication.

PVFS[17] and CXFS[19] are examples of parallelized file systems that can be used through conventional VFS calls. They implemented user transparent file system by a sophisticated mechanism. However, its implementation is limited to local cluster system and they suffer overheads of file system layer during data transfer between distant nodes and disk I/O bandwidth limitation due to randomness in disk access sequence.

Data transfer in recent cluster systems are achieved by extension of ftp. bbftp[5], sfcp[6] and GridFTP[7] are popular examples among scientific researchers. They can exploit communication speed by utilizing multi-stream, parallel file transfer. But they require special API of middleware [18] or explicit use of multiple files in user programs. They also suffer overheads and bandwidth limitation mentioned above.

The proposed Data Reservoir system utilizes performance from almost sequential disk accesses and efficiency of low-level protocol during data transfer between distant nodes. At he same time, they gives file system transparency to any file system that can have SCSI disk drives. Therefore, users can use VFS interface and cached file accesses to local user programs.

As for the separation of local and remote network connection, none of related systems shown here supports this capability.

6. Concluding Remarks

We are constructing a novel data sharing facility, Data Reservoir, that uses hierarchically striped disk files, iSCSI transfer protocol for both local accesses and global data transfer, and file caching that separate high-latency global communication to low-latency local file accesses. The use of low-level protocol (iSCSI) realizes file-system transparency, i.e. interoperability to any file system and highly efficient data transfer between disks.

We have built 1 Gbps models and 10 Gbps models. 1 Gbps models are currently used for transferring satellite data from the Institute of Space and Astronomical Science to Univ. of Tokyo(25 miles). We attain 960Mbps(95%) transfer bandwidth. Using the 1Gbps model, we also performed 1000 miles round trip measurements. Then, the 10Gbps model is examined.

Our approach has following advantages to existing file transfer or file sharing facilities:

(1) hierarchical striping of disk storage achieves scalability to network bandwidth, storage capacity and network latency even in transferring one huge file,

(2) Users of transferred data can use existing computing machines for analyzing data without modification of programs, installing middlewares or optimizing network drivers, and

(3) Sequential nature of disk accesses on global data transfer realizes much more efficient use of disk I/O bandwidth than data transfer through a file system.

We will install Data Reservoirs to all the location at Table 1 in two years for the efficient use of Super-SINET and other high-speed backbone networks.

Acknowledgements

The authors thank Akira Kato, Information Technology Center of Univ. of Tokyo for constructing network environments, Fumiaki Nagase, Akira Miura, and Yasumasa Kasaba of The Institute of Space and Astronautical Science for granting and helping to settle our system in their place, Hiroki Nogawa and Toyokazu Akiyama of Osaka University, Hideaki Sone of Tohoku University, and Hiroki Takakura of Kyoto University for settling 1000 mile round trip circuit on their GRID project network, Hideyuki Sakai,

Faculty of Science, Univ. of Tokyo for forming scientific research project groups, Masakazu Sakamoto and Toshitaka Yanagisawa from Fujitsu Program Laboratories LTD. for many constructive advices and coding. We also appreciate Foundry, Extreme, and Riverstone Networks. This study is supported by the Special Coordination Fund for Promoting Science and Technology, Ministry of Education, Culture, Sport, Science and Technology.

References

[1] http://www.ietf.org/internet-drafts/draft-ietf-ips-iscsi-12.pdf

[2] IEEE Annals of the History of Computing, Special issue on the University of Cambridge, Vol.14 (4), 1992

[3] http://data-reservoir.adm.s.u-tokyo.ac.jp/

[4] http://www.sinet.ad.jp/index-e.html

[5] http://ccweb.in2p3.fr/bbftp/

[6] http://www.slac.stanford.edu/ abh/sfcp/

[7] http://www.globus.org/datagrid/gridftp.html

[8] http://www.transarc.com/

[9] http://www.microsoft.com/NTServer/techresources/file print/dfs

[10] http://www.coda.cs.cmu.edu/

[11] K. Hiraki, M. Inba, J. Tamatsukuri, R. Kurusu, Y. Ikuta, H. Koga, and A. Zinzaki, “Data Reservoir: A 4Gbps Long Distance File Sharing Facility for Science Data Processing”, Poster session SC2001, Nov. 2001.

[12] K. Hiraki, M. Inaba, J. Tamatsukuri, R. Kurusu, A. Jinzaki, H. Koga, H. Sakai, Okamura, Y. Ikuta, Miyazawa, “Data Reservoir: Need of Network Infrustracture for Science” CPSY 2001. (In Japanese)

[13] K. Hiraki, et al, “Data Reservoir: A New Approach to Data-Intensive Scientific Computation,”, Proc. Int. Symp. on Parallel Architecture, Algorithm and Network, pp.269–274, 2002

[14] M. Inaba, R. Kurusu, J. Tamatsukuri, H. Koga, A. Jinzaki, Y. Ikuta, H. Sakai and K. Hiraki, “Data Reservoir: A very high-speed Long distance file sharing facility for Scientific data processing,” Proc. of the High Performance Computing Symposium 2002, IPSJ, pp. 81 – 88, Jan. 2002. In Japanese)

[15] M. Inaba, J. Tamatsukuri, R. Kurusu, A. Jinzaki, H. Koga, Y. Ikuta, and K. Hiraki, “Data Reservoir — Data Transfer facility, Our Approach and experimental result” CPSY 2001. (In Japanese)

[16] R. Kurusu, M.Sakamoto, Y.Ikuta, K. Hiraki, M. Inaba, J. Tamatsukuri, H. Koga, and A. Zinzaki, “Data Reservoir, Multi-Gigabit Data Transfer Facility, Its Design and Implementation” PDCAT industry session, Sept 2002.

[17] P.H.Carns, W.B.Ligon III, R.B. Ross and R. Thakur, “PVFS: A Parallel File System For Linux Cluster,” Proc. of the 4th Annual Linux Showcase and Conference, pp. 317-327, Oct. 2000.

[18] J. Huber, C.L.Elford, D.A.Reed, A.A. Chien and D.S.Blumenthal, “PPFS: A high performance portable parallel file system,” Proc. of the 9th ACM Int. Conf. on Supercomputing, pp. 385-394, July 1995.

[19] J. Mostek, B. Earl, S. Levine, S. Lord, R. Cattelan, K. McDonell, T. Kline, B. Graffey, and R. Ananthanarayanan, ”Porting the SGI XFS file system to Linux,” Proc of FREENIX Track, 2000 USENIX Annual Technical Converence, 2000.

[20] E.K.Lee and C.A.Thekkath, “Petal: Distributed Virtual Disks,” Proc. ASPLOS-VII, pp. 84-92, Oct. 1986.

[21] A. Savva, T. Kkashi, T. Shimizu, and M Kishimoto, “Distributed Shared Disk: Transparent and Efficient Disk Sharing in Cluster Systems,” Proc. JSPP98, pp. 335-342, June 1998.