A Cost-Effective, High Bandwidth Server I/O network Architecture for Cluster Systems

Hsing-bung Chen, Gary Grider, Parks Fields
Los Alamos National Labs
Los Alamos, New Mexico 87545 USA
hbchen@lanl.gov

Abstract

In this paper we present a cost-effective, high bandwidth server I/O network architecture, named PaScal (Parallel and Scalable). We use the PaScal server I/O network to support data-intensive scientific applications running on very large-scale Linux clusters. PaScal server I/O network architecture provides (1) bi-level data transfer network by combining high speed interconnects for computing Inter-Process Communication (IPC) requirements and low-cost Gigabit Ethernet interconnect for global IP based storage/file access, (2) bandwidth on demand I/O network architecture without re-wiring and reconfiguring the system, (3) Multi-path routing scheme, (4) reliability improvement through reducing large number of network components in server I/O network, and (5) global storage/file systems support in heterogeneous multi-cluster and Grids environments. We have compared the PaScal server I/O network architecture with the Federated server I/O network architecture (FESIO). Concurrent MPI-I/O performance testing results and deployment cost comparison demonstrate that the PaScal server I/O network architecture can outperform the FESIO network architecture in many categories: cost-effectiveness, scalability, and manageability and ease of large-scale I/O network.

Keywords: Parallel and Scalable I/O, Cluster computing, Global Storage/File system, Multi-path routing

1. Introduction

Computing, server I/O, and storage systems are the three most critical elements to build a very large-scale cluster system. Without these three elements being well organized and balanced, we cannot fully utilize a cluster system [1][2][3][4][5][6]. The proposed PaScal[1] (**Pa**rallel and **Scal**able) architecture is designed as a highly scalable server I/O network to meet constantly increasing computation power and storage capacity.

The main goal of PaScal is to provide cost-effective, high performance, efficient, reliable, parallel, and scalable I/O capabilities for data-intensive scientific applications running on very large-scale clusters. Data-intensive scientific simulation-based analysis normally requires efficient transfer of a huge volume of complex data among simulation, visualization, and data manipulation functions.

PaScal adopts several scale-up (parallel) and scale-out (scalable) networking features such as

1) Bi-level switch-fabric interconnected systems by combining high speed interconnects for inter-processes message passing requirement and low-cost Gigabit Ethernet interconnect for IP based global storage access,
2) A bandwidth on demand linear scaling I/O network architecture without re-wiring and reconfiguring the system,
3) Equal Cost Multi-path routing scheme,
4) Improve reliability through reducing large number of network components in server I/O network, and
5) Support for global storage/file systems in heterogeneous multi-cluster and Grids computing environment.

The rest of this paper is organized as follows: In Section 2, we present the requirements and problems of building a fully functional Cluster Computing System. In section 3, we provide the system view of the proposed PaScal server I/O architecture. In Section 4, we present performance evaluation results from parallel MPI-IO benchmark testing and deployment cost comparison of PaScal I/O vs. Federated I/O. In Section 5, we conclude and describe our future works.

1-4244-0910-1/07/$20.00 ©2007 IEEE

[1] This work was carried out under the auspices of the National Nuclear Security Administration of the U.S. Department of Energy at Los Alamos National Laboratory under Contract No. DE-AC52-06NA25396. LA-UR-06-7462

2. Requirements and Problems

Generally a cluster system requires high speed computing with big-memory server nodes and a high bandwidth global storage/file system. A global storage/file system employs a single global namespace [9][10], removes physical and logical boundaries, provides parallel data paths to compute nodes, scales client network capacity, grows seamlessly with a single global namespace, and supports dynamic load balancing. High bandwidth server I/O networking serves as a data “superhighway” to bridge heavy data traffic workloads between server nodes and global storage/file systems.

2.1. Related works

Server I/O networks are used by servers to connect to I.O devices, client, and other servers. Various server I/O architectures have been proposed and deployed in large-size clusters [2][23][24][25][26]. Most of them are using Federated switch fabrics [23][24][25] or reduced mode Federated switch fabrics [26] to address the I/O scaling issues when growing a global storage network.

2.2. Networks used in cluster systems

We classified the interconnection network used in cluster systems into two categories.

1) Level-1 interconnect uses high speed interconnect systems with non-blocking (Fat tree or Clos tree) tree topology such as Quadrics, Myrinet, or Infiniband for fulfilling requirements of low latency, high speed, high bandwidth cluster Inter-Process Communication (IPC), and
2) Level–2 interconnect uses IP based storage network to support latency-tolerant I/O communication and global storage/file systems.

Level-1 interconnect is a “must” in a large-scale cluster system. We only focus on the study of Level-2 interconnect in this paper.

2.3. Level-2 interconnect architectures

There are two different server I/O network networking architectures used in Level-2 IP based interconnection network [9]:

1) Federated Ethernet Server I/O network (FESIO) networking architecture connects the data storage system to a common Constant Bi-section Bandwidth (CBB) based Federated Ethernet I/O Network (Level-2). It then connects all cluster compute server nodes both to the level-2 network and to a high-performance interconnect system (Level-1) such as Quadrics, Myrinet, or Infiniband
2) Linear scaling Ethernet Server I/O network (LESIO) networking architecture connects the data storage system to a linear-scaling Ethernet I/O Network. It uses server I/O network nodes as area border routers to route data traffic for multiple compute nodes and connects all cluster server nodes (compute and I/O) to a high-performance interconnect systems (Level-1) such as Quadrics, Myrinet, or Infiniband.

2.4. Federated Ethernet Se rver I/O (FESIO)

FESIO architecture, shown in Figure-1, normally uses "Federated" Ethernet switches in the Level-2 network that are configured as non-blocking, CBB tree topologies (Fat-tree or Clos-tree). In multiple Gigabit-switches cluster I/O configurations, CBB from cluster compute nodes to global storage/file system is achieved when the aggregate uplink bandwidth is equal to or greater than the aggregate nodal bandwidth on each access switch and the uplinks are evenly distributed among the non-blocking core switches. FESIO is typically proposed and used in a multi-cluster environment for supporting scalable I/O networking. Every server node is equipped with two network interface cards (NICs). One NIC is connected to the Level-1 network for latency sensitive IPC communication and the other NIC is connected to the Level-2 network for data I/O traffic. A single-path routing policy is normally adopted in FESIO’s network [18][19][20].

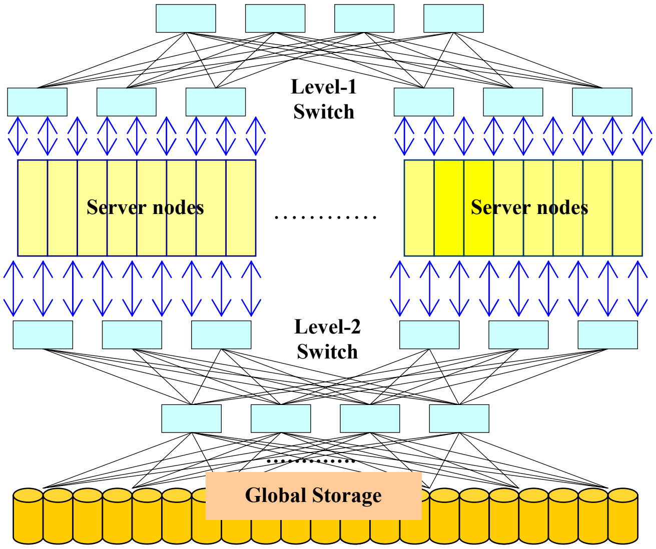


Figure-1: FESIO based Server I/O network architecture

The advantage of this server I/O network architecture is that it can be scaled to support thousands of cluster nodes while continuing to provide high bandwidth connectivity to the rest of the network. The FESIO architecture has many scaling problems such as (1) it has redundant network, (2) it is very expensive and complex to grow this type of federated network, (3) it forces compute server nodes to get involved in “single-path I/O routing”, (4) single-path routing used in FESIO provides no load balancing between the Level-1 and the Level-2 networks (i.e. Level-1(Myrinet/Infiniband) vs. Level-2(Gigabit Ethernet)), (5) it cannot scale well in a cost effective manner, (6) it needs a very complicated Ethernet routing/switching configuration in the Level-2 network, and (7) it has error prone NIC cable installation and complicated cable management overhead.

2.4.1. Reduced FESIO network architecture

Without using a fully connected Level-2 IP based network, FESIO's Level-1 network vendors (Quadrics, Myrinet, and Infiniband) provide so called Ethernet concentrator (Gigabit Ethernet or 10-Gigabit Ethernet Connection Modules) in Level-1's switches. Those Ethernet Concentrators are used to provide additional accessing connectivity from Level-1 network to IP based network. Due to the limited number of concentrator modules supported inside a Level-1 switch, generally those Ethernet Concentrators can not scale well to support high-volume of data access bandwidth (100GB/sec, 200GB/sec…, Multiple TB/sec, etc) for a Petascale data intensive computing.

2.5. Linear scaling Ethernet Server I/O network

The LESIO (Linear scaling Ethernet Server I/O) architecture is shown in Figure-2. In LESIO, we separate server nodes into two categories based on their operational purposes; (a) Compute server nodes are only used for cluster computation and (b) I/O server node are only used for I/O routing and access to global storage systems. The ratio of I/O node vs. Compute node is normally less than 5% dependant on the bandwidth requirement between computing and storage. The LESIO architecture provides a bandwidth on-demand linear growing path in terms of deployment cost and data accessing performance. More I/O server nodes and Ethernet switches can be added linearly to fulfill the requirement of increasing storage capacity and I/O accessing bandwidth.

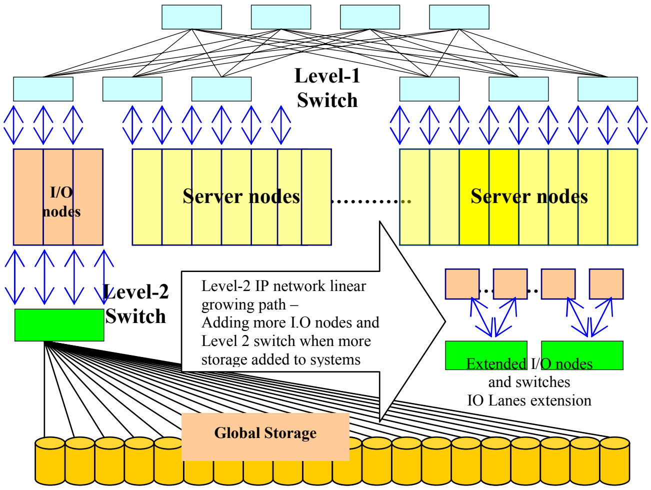

Level-1 Switch
I/O nodes
Server nodes
....
Server nodes
Level-2 Switch
Level-2 IP network linear growing path – Adding more I.O nodes and Level 2 switch when more storage added to systems
Extended I/O nodes and switches IO Lanes extension
Global Storage

Figure-2: LESIO based Server I/O network architecture

3. OVERVIEW OF PASCAL ARCHITECTUIRE

PaScal, LESIO based, adopts several hardware and software components to provide a unique and scalable server I/O networking architecture. Figure-3 has shown the system components used in PaScal.

3.1. Hardware Components used in PaScal

Level-1 High Speed Interconnection Network

The Level-1 interconnect uses (a) high speed interconnect systems (scale up) such as Quadrics, Myrinet, or Infiniband for fulfilling requirements of low latency, high speed, high bandwidth cluster IPC communication and (b) aggregating I/O-Aware multi-Path routes (scale-out) for load-balancing and failover.

Level–2 IP based Interconnection Network

The Level-2 interconnect uses multiple Gigabit Ethernet switches/routers with layer-3 network routing support (scale out) to provide latency-tolerant I/O communication and global IP based storage systems. Without using the "Federated network" solution, we can linearly expand the Level-2 IP based network by employing a global host domain multicasting feature in metadata servers of a global file system. With this support we can maintain a "single name space" global storage system and provide a linear cost growing path for I/O networking.

Compute node

A Compute node is equipped with at least one high-speed interface card connected to a high-speed interconnect fabric in Level-1. The node is setup with Linux multi-path equalized routing to multiple available I/O nodes for load balancing and failover (high availability). A Compute node is used for computing only and is not involved with any routing activities.

I/O node

I/O node: An I/O routing node has two network interfaces. One high-speed interface card is connected to the Level-1 network for communication with Compute nodes. One or more Gigabit Ethernet interface cards (bondable) are connected to the Level-2 linear scaling Gigabit switches. I/O nodes serve as the routing gateways between Level-1 and Level-2 network. Every I/O has the same networking capability.

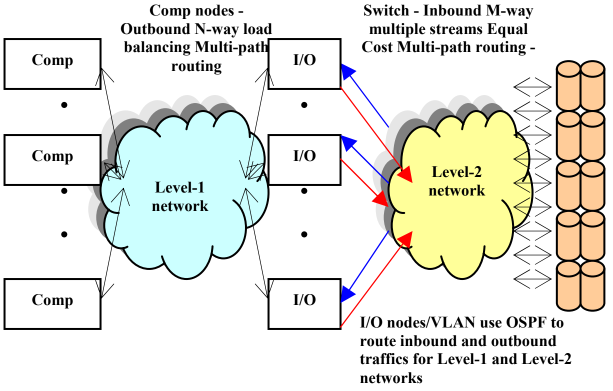

Comp nodes - Outbound N-way load balancing Multi-path routing
Switch - Inbound M-way multiple streams Equal Cost Multi-path routing -
Comp
Comp
Comp
I/O
I/O
I/O
Level-1 network
Level-2 network
I/O nodes/VLAN use OSPF to route inbound and outbound traffics for Level-1 and Level-2 networks

Figure-3: Components used in PaScal

3.2. System Software Components used in PaScal

3.2.1. Equal Cost Multi-path Routing for load balancing

Multi-path routing is used to provide balanced outbound traffic to the multiple I/O gateways. It also supports failover and dead-gateway detection capability for choosing good routes from active I/O gateways. Linux Multi-Path routing is a destination address-based load-balancing algorithm. The selection of an ideal candidate I/O node is based on the hashing results of Layer-2 MAC addresses and Layer-3 IP addresses from a source Compute node and destination storage targets. Multi-path routing (RFC2991, RFC2992) is applied to utilize several routing paths to distribute traffic from a source to a destination [11] (Figure-3). Multi-path routing should improve system performance through load balancing and reduce end-to-end delay. Multi-path routing overcomes the capacity constraint of "single-path routing" and routes through less congested paths.

Each Compute node is setup with N-ways multi-path routes thru "N" I/O nodes. Multi-path routing also balances the bandwidth gap between the Level-1 and the Level-2 interconnects. Table-1 shows the advantage of using multipath routing in Level-2 network. We use the Equal Cost Multi-path (ECMP) routing strategy on compute nodes so compute nodes can evenly distribute traffic workloads on all I/O nodes. Best path routing strategy is not used here because every I/O node is with the same networking capability.

With this bi-direction multi-path routing we can sustain parallel data paths for both write (outbound) and read (inbound) data transfer. This is especially useful when applied to concurrent socket I/O sessions on IP based storage systems. PaScal can evenly allocate socket I/O sessions to routing available I/O routing nodes. We used an I/O-aware load balancing multi-path routing policy on both load balancing initiators from Compute node writing/sending traffic and the Level-2 Gigabit Ethernet switch reading/receiving traffic.

Network Interface Card	Gigabit Ethernet	Myrinet D card	Infiniband 4x HCA
IP over NIC (Ethernet, Myrinet, Infiniband)	125MB/sec	250MB/sec	500MB/sec
{bandwidth of IPoverNIC} vs. {Bandwidth of Gigabit Ethernet}	1:1	2:1	4:1
Bandwidth from Single path routing used in Level-2 network: FESIO cannot balance the bandwidth bias between Level-1 and Level-2 network	125MB/sec	125MB/sec	125MB/sec
Bandwidth from Multipath routing used in Level-2 network: LESIO can balance the bandwidth bias between Level-1 and Level-2 network	125 MB/sec	250 MB/sec 2-way Multipath routing	500MB/sec 4-way Multipath routing

Table 1: Multipath routing advantages

3.2.2. OSPF routing used in I/O nodes

OSPF routing capability in I/O nodes and Level-2 Ethernet switches is used to efficiently manage the inbound and outbound traffic for bi-direction load balancing. We evenly assign I/O nodes into multiple subnets and create corresponding VLANs in Level-2's Ethernet switches to work with each I/O node subnet. Instead of advertising the whole compute node community from each I/O node, I/O nodes sub-netting will drastically reduce compute node route advertising overhead. Each I/O node will just advertise routes within its subnet. The number of I/O nodes per Subnet/VLAN is dependent on the capability from Level-2's Ethernet switches. In general it could be 4-ways, 8-ways or 16-ways ECMP. OSPF routing overhead is not seen here due to the limited number of routing hops used in PaScal server I/O network. We also designate each I/O Subnet/VLAN in a "stub area with no summary". This prevents the I/O node and switch's VLAN from advertising route summary (external routes, optional inter-are routes) into I/O node's subnet stub area. Using "Stub area with no summary" can significantly reduce the size of the routing tables in I/O nodes and provide some isolation in the area/subnet from changes in topology outside the area/subnet. This also eliminates type-3/type-4/type-5 LSA-Summary messages and reduces 70%~75% of OSPF routing overhead. With OSPF dynamic routing capability, we can gradually grow I/O node subnets to accommodate a multi-cluster environment without any impact on the existing compute node community [22].

3.3. PaScal I/O on Multi-clusters Environment

With (a) network Layer-2 and Layer-3 fail-over support from Linux kernel routing implementation and Ethernet switch capabilities and (b) a global multicast domain support from scalable metadata servers, the PaScal I/O networking architecture can support a global storage system in a heterogeneous multi-cluster and Grids environment.

Figure-5 illustrates the top-level view of PaScal in a heterogeneous multi-cluster and Grids environment. We can apply PaScal to support a heterogeneous multi-clusters environment that is consisted of several independent large-scale cluster systems. These systems are possibly managed by separate research organizations. PaScal provides an ability to mount a single name space global file system across all clusters. Each cluster maps its I/O routing paths through multiple "IO-Lanes".

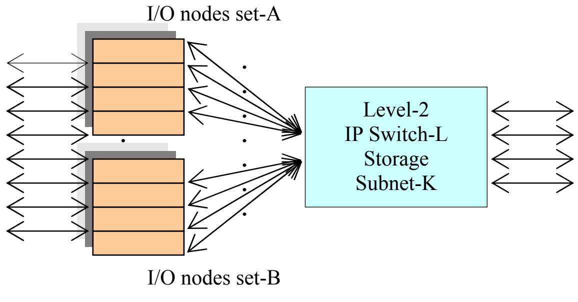
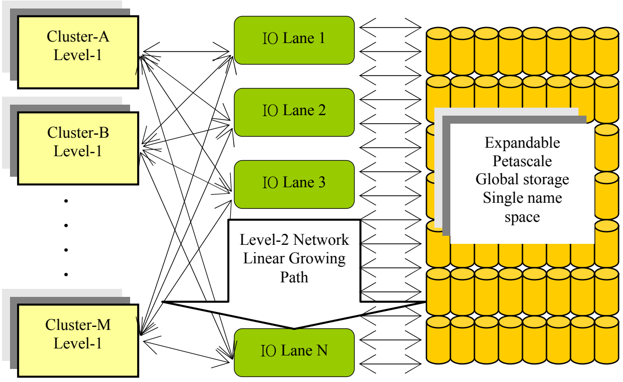


Figure 4: IO Lane

An "IO-Lane" is consisted of a group of I/O nodes managed and routed by an individual Gigabit/10gigabit Ethernet switch. Each IO-Lane provides accessibility to a set of storage subnets. We then use a global domain multicasting to maintain a global file with a single name space. With this we can support a Peta-scale global file system accessible for multi-cluster environments using the PaScal I/O architecture. We can linearly add more "IO-Lanes" into the PaScal I/O architecture to meet the increasing bandwidth demand of global parallel file systems. Remote Grids computing facilities, with "PaScal's IO Lanes using multi Gigabit Ethernet links or multiple 10-Gigabit Ethernet links", can participate the sharing of a distance-less remote/global storage/file system through long-haul optical links. The purpose of using IO Lane is to mitigate single switch bandwidth limitation and provide a linear growing path for IP storage network.



Global Domain (single name space) = $\sum$Subnet (Lane$_i$ (i=1..M)),
Level-2 Linear scaling IP routing network

Figure-5: PaScal I/O on Multi-Cluster environments

4. Evaluation of PaScal Server I/O Network Architecture

We have implemented the PaScal Server I/O network Architecture on several large-scale Linux Clusters (over 9000+ nodes, 7 large-size Linux clusters) at LANL in the past two years. The following describes the PaScal's performance/cost evaluation on (1) LANL's 1024-node Pink Linux cluster (dual Xeon 2.4 GHZ CPUs), (2) LANL's 256-node BlueSteel cluster (dual AMD Opteron 2.0GHZ CPUs), and (3) Ptest Cluster 12-node AMD-Opteron machines.

4.1. Performance evaluation – Parallel MPI-IO benchmark testing using PINK Cluster

LANL's Pink cluster equipped with (1) 960 Compute nodes with one Myrinet PCI-X D-card, (2) 64 I/O routing nodes with one Myrinet PCI-X D-card and one Gigabit Ethernet interface, 8GB/sec bandwidth from 64 I/O routing nodes, (3) Myrinet switch fabric (1024 node Clos tree) for the Level-1 interconnect, (4) One Extreme Network Black-Diamond 6808 Gigabit Ethernet switch with Layer-3 routing capability for the Level-2 I/O interconnect for accessing IP based global storage, and (5) Eight shelves of Panasas ActiveScale Storage system [12].

4.1.1. Synthesized parallel I/O Access Patterns

We have used two synthesized parallel I/O access patterns referred to as "N-to-1" and "N-to-N". "N-to-1" and "N-to-N" parallel I/O access patterns are the two most used in message-passing based scientific application programs such as BLAST, FLASH, BTIO, Parallel Ocean Program Model, Quantum Chemistry, Terrain rendering, Electron scattering, and LANL's SAGE code [13][14][15][16][17].

In the "N-to-N" I/O access pattern shown in Figure-6, each of N processes concurrently reads/writes from/to a corresponding file. Each file is striped across multiple disk drives.

In the "N-to-1" I/O access pattern shown in Figure-7, all N processes concurrently read/write from/to one single file. Every process accesses a sequence of non-overlapping regions within a file. This single file is striped across multiple disk drives.

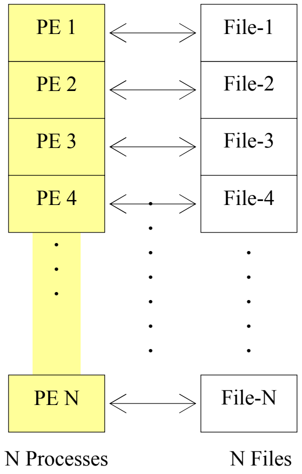


Figure-6: "N-to-N" Read and Write

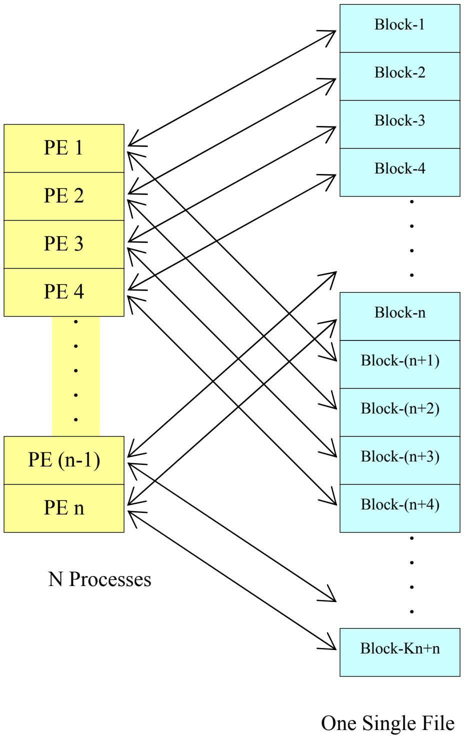


Figure-7: N-to-1 Read and Write

4.1.2. Panasas File System

The Panasas ActiveScale Storage system is composed of five major components in the Panasas Storage system. These are:

1) The primary component is the object, which contains the file data and enough additional meta data for the OSD to autonomously manage it. An object can be viewed as an offset and length in a file, with each object located on a different OSD,
2) The network attached Object Storage Device (OSD) (a.k.a. Storage Blade), which is a more intelligent evolution of today's disk drive that autonomously manages the object, including its layout,
3) The Panasas File System (PanFS) client module, which accepts POSIX file system commands and data from the client operating system, addresses the OSDs directly and stripes objects across multiple OSDs. It is currently available as a Linux kernel module,
4) The PanFS Metadata Server (MDS) (a.k.a. Director Blade), which is distinct from the OSD, intermediates amongst multiple clients allowing them to share data while maintaining consistency on all nodes. The MDS provides clients with signed capabilities, that the clients then present to an OSD for accessing object contents, and
5) A high bandwidth network fabric that ties the clients to the OSDs and the MDSs.

4.1.3. Parallel MPI-IO benchmark testing

We have conducted a sequence of parallel MPI-IO benchmarks on Pink to assess the impact on performance and scaling of the PaScal framework. We have developed a parallel MPI-IO software benchmark that uses the MPI-IO API to write and then read files in a variety of patterns that mimic the I/O profiles of scientific simulation codes.

All of the parallel MPI-IO testing runs were against

1) Four shelves of Panasas Active Scale Storage (20TB) with an estimated maximum bandwidth (EMaxB) of 1600MB/sec for both read and write access [12], or
2) Eight shelves of Panasas Active Scale Storage (40TB) with an EMaxB of 3200MB/sec for both read and write access [12].

For all tests, the minimum effective bandwidth, or the speed of the slowest process, is reported. By "effective" we mean that the files create/open and close times are all factored into the calculation of bandwidth. Even with open and close times factored into the bandwidth calculation, initial results are very promising in terms of scaling and performance.

Here is the definition for effective read and write used in performance studies.

$N\ task{:}\ Task_{i,\ i=1..n}$ // Number of Task
$FileCreateTime(Task_{i,\ i=1..n})$ // File creation time
$FileOpenTime(Task_{i,\ i=1..n})$ // File Open time
$FileCloseTime(Task_{i,\ i=1..n})$ // File Close Time
$FileReadTime(Task_{i,\ i=1..n})$ // File read time
$FileWriteTime(Task_{i,\ i=1..n})$ // File write time
$FinishReadTime(Task_{i\ i=1..n})=FileCreateTime(Task_i)+FilOpenTime(Task_i)+$
$FileCloseTime(Task_i)+FileReadTime(Task_i)$
$FinishWriteTime(Task_{i\ i=1..n})=FileCreateTime(Task_i)+FilOpenTime(Task_i)+$
$FileCloseTime(Task_i)+FileWriteTime(Task_i)$
$EffectiveReadBandwidth= SizeOfFile\ /\ Max(FinishReadTime(Task_{i,\ i=1..n}))$
$EffectiveWriteBandwidth= SizeOfFile\ /\ Max(FinishWriteTime(Task_{i,\ i=1..n}))$

4.1.4. N-to-N Concurrent Write

Figure-8a shows the effective write bandwidth for the "N-to-N" case using 2 to 512 processes with the write message size between 512KB and 64MB. Each process wrote a single 4 GB file with sequential (not random) write access and all N files were created under one subdirectory. You can see that the write bandwidth is steadily approaching/passing 1 GB/sec as we include more and more processors. The peak write bandwidth is about 1309MB/sec. Remember that at 256 processors, 256 files are all being created and opened at approximately the same time. The results also show that the maximum bandwidth achieved does not decrease much with larger message sizes.

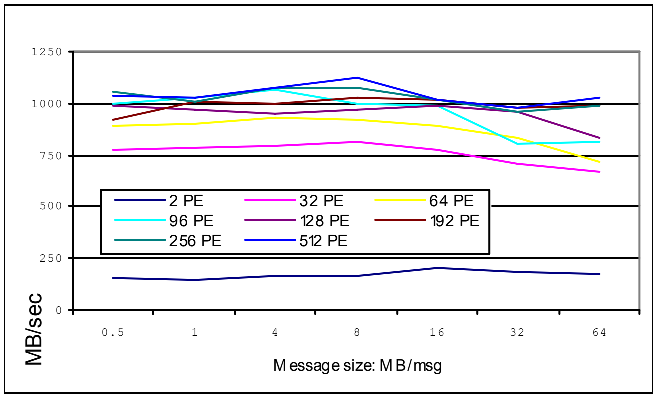


Figure-8a: N-to-N Effective Write Bandwidth

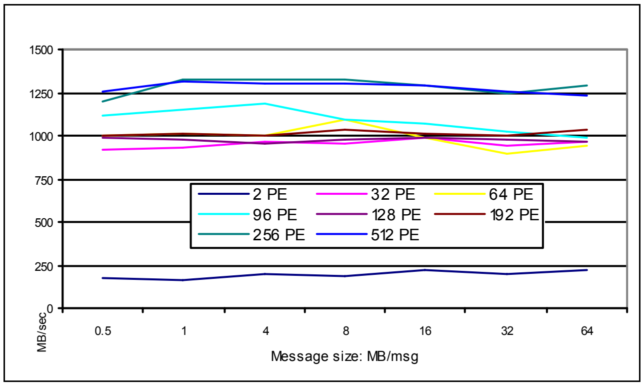


Figure-8b: N-to-1 Effective Read Bandwidth

4.1.5. N-to-1 Concurrent Write

For the "N-to-1" write bandwidth, shown in Figure-8b, we are using 2 to 512 processes with the write message size between 512KB and 64MB and a single 512 GB file. In the "N-to-N" case, we see fairly consistent performance across a large range of message sizes and an increase in overall bandwidth as more processes are writing. The maximum write bandwidth peaks at approximately 1125 MB/sec, which is less than the "N-to-N" case. Essentially all processes are opening, writing, and closing a single file at the same time. This creates contention from accessing the control path in the file system and explains the reduced peak bandwidth.

4.1.6. "N-to-N" and "N-to-1" Concurrent Read

In the "N-to-N" concurrent read test case, we launch N read processes. Each process reads a separate 4GB file using message size ranges from 512KB to 64MB. In the "N-to-1" read case, we enable N read processes access to a single 512GB file using message size ranges from 512KB to 64MB. 4 to 512 processes are used in both "N-to-N" and "N-to-1" read testing. The scaling read result of "N-to-N" case in Figure-9a shows that we can reach the peak bandwidth about 1477MB/sec across four Panasas storage shelves as the message size is approaching 4MB. This can be explained for the reason that larger size messages imply a few number of transactions, resulting in the reduced overhead of packet transmission. The overall result of the "N-to-1" read shown in Figure-9b is slightly worse (10~15 %) than the "N-to-N" case due to root causes mentioned in the "N-to-1" write case.

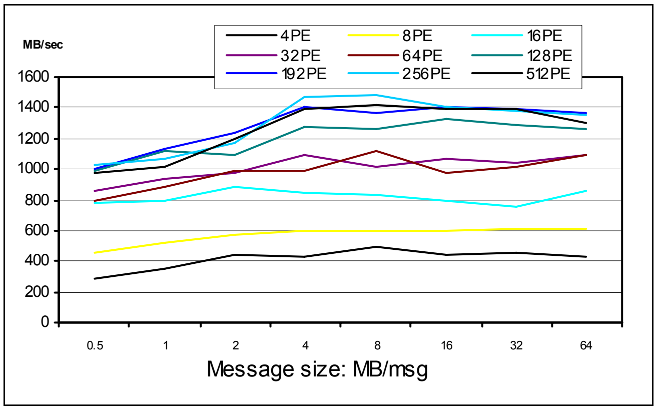


Figure-9a: N-to-N effective read bandwidth

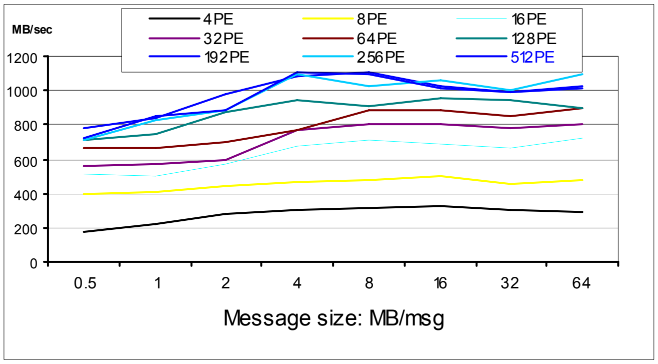


Figure-9b: N-to-1 effective read bandwidth

4.2. Evaluation of Single Gigabit Ethernet Link Vs. Multiple Gigabit Ethernet Links bonding

We use Link Aggregation on I/O routing nodes. Link Aggregation is a common practice of bonding multiple physical links into a single link for increased bandwidth. We can increase the capacity and availability with Link Aggregation. Link Aggregation provides (1) load balancing so that no single link is overloaded and (2) fail-over so that no single link prevents a disruption of the communication between the interconnected devices. We used a 256-node BlueSteel Cluster equipped eight I/O routing nodes with dual Gigabit Ethernet bonded links to, launch an "N-to-N Read/Write from/to 4 Panasas shelves" testing using 128 PEs. Figure-10a and Figure-10b show the summarized results from 16 runs of testing.

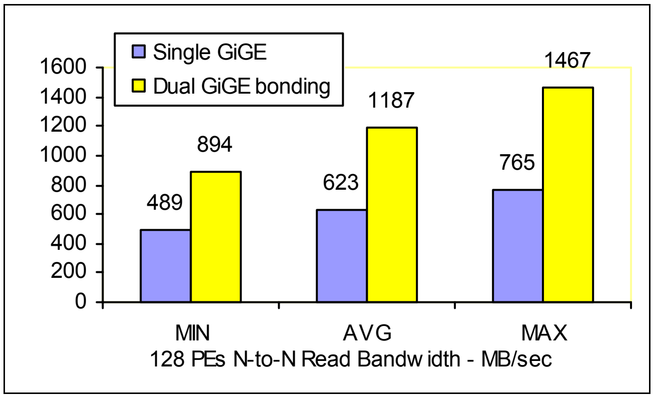


Figure-10a: Bonding N-to-N Read

We can obtain about 180% bandwidth improvement using dual-Ethernet-link bonding compared to a single Ethernet link. Applying Link Aggregation with multiple NICs we can (1) reduce the amount of I/O routing nodes and maintain the same bandwidth capacity or (2) increase the bandwidth capacity and maintain the same amount of I/O routing nodes.

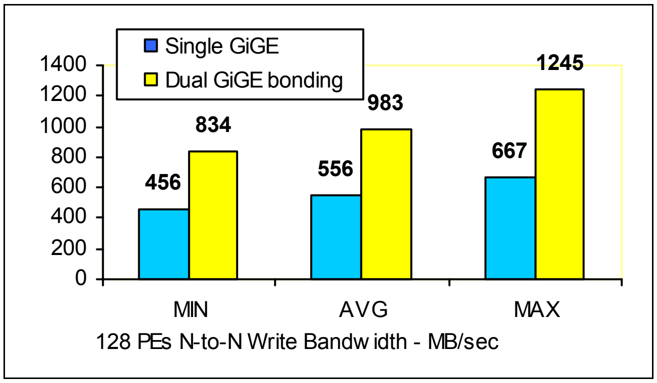


Figure-10b: Bonding N-to-N Write

4.3. Evaluation of Single Client's Performance

The performance evaluation of a single client's Single-Path routing in FESIO (SP/FESIO, Single Path route to level-2 IP network used in compute server node) vs. Multi-path routing in PaScal (MP/PaScal, N-way Multipath routes used in a single compute server node) was executed on (1) a 12-node IBM E-server-326 AMD 2Ghz Opteron machines, (2) one 16-port Myrinet switch & one 24-port TopSpin Infiniband switch for Level-1 network, and (3) one Extreme Networks 6808 Gigabit Ethernet switch for Level-2 network. In the SP/FESIO testing, each server node is equipped with one PCI-X Myrinet D card or PCI-X Infiniband 4X HCA card and one PCI-X Gigabit Ethernet card. In the MP/PaScal testing, each compute node and each I/O node is equipped with one PCI-X Myrinet card or PCI-X Infiniband 4X HCA card and each I/O node is equipped one/two/four (bonding) PCI-X Gigabit Ethernet cards. Results in Figure-11 show that (1) a single client using SP/FESIO can only get a maximum bandwidth from a single Gigabit Ethernet link and (2) a single client using MP/PaScal can obtain scaling bandwidth when the number of Gigabit Ethernet bonding increased in I/O nodes.

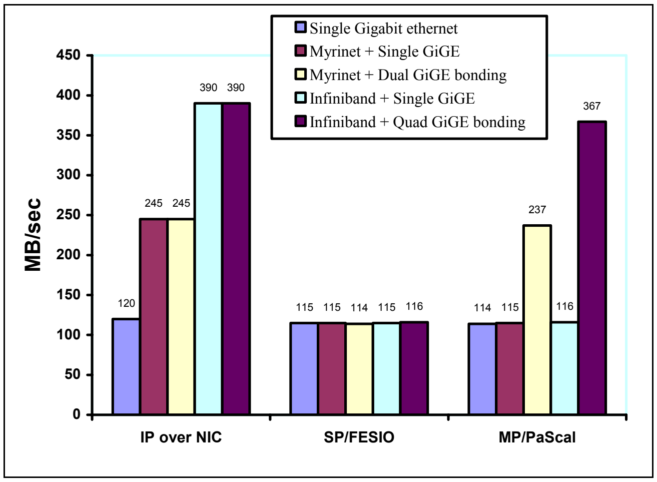


Figure-11: Single Client performance comparison

4.4. Network consolidation and cost reduction

The FESIO architecture is used widely in several large-scale clusters such as LLNL's (Lawrence Livermore National Laboratory) Multi-Cluster Global Scalable Storage System [18], and a previous configuration of LANL's Lightning cluster. We have done a cost/performance comparison between FESIO architecture and the PaScal architecture on the deployment of LANL's Pink 1024 node cluster equipped with eight shelves of Panasas File System.

Table-2 lists the I/O equipment/price used to build both architectures. PaScal can save up to 95.65% on networking equipment. PaScal can reduce Gigabit Ethernet port count by 96.88%. PaScal can reduce Gigabit NIC cards by 93.75% and decrease Gigabit Ethernet cable management complexity ratio from 32 to 1.

Table-3 lists the comparison results of PaScal vs. FESIO architectures. The major reduction of cost from the PaScal I/O architecture is a result of eliminating superfluous high-end/high-port count Gigabit switches while maintaining the same I/O bandwidth. The cost of Gigabit Ethernet Switches from Extreme Networks was based on the price quote from the vendor

Item\Architecture	FESIO		PaScal	
Extreme 6816 Switch	12	$2,688,000	0	
Extreme 6808 Switch	0		1	$116,000
GigE NICs	1024	$51,200	64	$3,200
GigE cables	2048	$10,240	64	#320
Used GigE ports	2048		64	
Total Cost	$2,749,400		$119,520	

Table-2: Cost comparison of FESIO vs. PaScal

Category	PaScal vs. FESIO
Cost saving	95.65%
Gigabit NIC card reduction	93.75% (1;16)
Gigabit Port count reduction	96.88%
Cable management complexity ratio	1:32

Table-3: Advantages of using PaScal vs. FESIO

4.5. Comparison of Reserved I/O Bandwidth vs. Wasted I/O Bandwidth and Cost growing index

We compare the reserved I/O bandwidth vs. wasted I/O bandwidth using FESIO and PaScal I/O architectures.

Reserved I/O Bandwidth (RIOB)=
*#switch * BandwidthCapacity(G-128switch, fully populated bandwidth)*

Wasted I/O Bandwidth(WIOB) = (Reserved I/O Bandwidth – Required I/O Bandwidth[2]*)/Reserved I/O Bandwidth → if (#switch > 1)*
or
0% → if (#switch = = 1) // no bandwidth is wasted, basic requirement

We grow a Pink-like cluster from 128 nodes to 8192 nodes and use G-256 (256 Gigabit port Ethernet switch with full populated bandwidth) switch to construct Level-2 Gigabit Ethernet network. FESIO wastes about 97.92% reserved I/O bandwidth (Table-4) and PaScal I/O has no waste in reserved I/O bandwidth (Table-5).

#Node	Pink like cluster		# Switch	Used ports	Used Cable	RIOB GB/sec	WIOB %
	Tera Flop	Required I/O GB/sec	Federated I/O CBB tree				
128	1	1GB	1	128	128	32GB	0%
256	2	2GB	1	256	256	32GB	0%
512	4	4GB	6	1536	1024	192GB	97.92%
1024	8	8GB	12	3072	2048	384GB	97.92%
2048	16	16GB	24	6144	4192	768GB	97.92%
4096	32	32GB	48	12288	8192	1536GB	97.92%
8192	64	64GB	96	24576	16384	3072GB	97.92%

Table-4: Reserved and Wasted bandwidth – FESIO

[2] Reference to [21]

#Node	Pink like cluster		# Switch	Used ports	Used IO nodes	RIOB GB/sec	WIOB %
	Tera Flop	Required I/O GB/sec	Federated I/O CBB tree				
128	1	1GB	1	8	8	32GB	0%
256	2	2GB	1	16	16	32GB	0%
512	4	4GB	1	32	32	32GB	0%
1024	8	8GB	1	64	64	32GB	0%
2048	16	16GB	1	128	128	32GB	0%
4096	32	32GB	1	256	256	32GB	0%
8192	64	64GB	2	512	512	64GB	0%

Table-5: Reserved and Wasted bandwidth - PaScal I/O

High port count Gigabit Ethernet switch is mainly the dominant cost factor to build a Level-2 server I/O network. We have defined a cost-growing-index.

Cost-growing-index = CGI(Cost(N nodes) =
price(#switches) + price(#ports) + price(#cable))
≅ CGI(Total-number-switch-used-in-Level-2-network)

In Figure-12, PaScal demonstrates a substantial benefit of cost-saving in terms of total number of high port count Gigabit Ethernet switch (256 ports) used in Level-2 server I/O network.

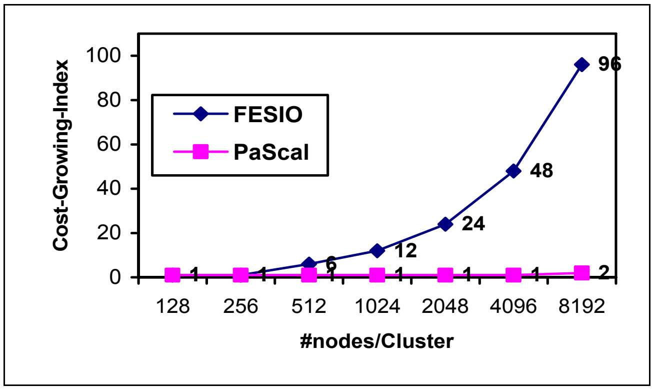


Figure-12: Cost growing index – PaScal vs. FESIO

5. Conclusion and Future works

Performance results from our parallel MPI-IO benchmarks on the Pink cluster demonstrate that the proposed PaScal I/O architecture is robust and capable of scaling bandwidth on large-scale Linux clusters. Scaling/aggregating test results prove that PaScal I/O architecture can preserve bandwidth on I/O nodes and meet the seamless capacity growth requirement of a global storage /file system.

More I/O routing nodes and Gigabit Ethernet switches can be linearly added to the Level-2 interconnect system when more storage access bandwidth is needed from the Level-1 Compute nodes.

Compared to the FESIO architecture, the proposed PaScal architecture clearly provides a great deal of network consolidation and cost saving in terms of reducing unnecessary (1) Ethernet NIC interfaces, (2) Ethernet cables, and (3) high-end/high port count Ethernet switches. Again, few networking components may help improving system reliability and manageability.

PaScal also provides a better solution for problems in FESIO architecture:

1. it has no redundant network and 0% wasted reserved-bandwidth,
2. it is very cost-effective and easy to grow Level-2 server I/O network,
3. it can scale very well as the system keeps growing,
4. it eliminates the I/O routing interferences on back-end Compute nodes and reduce significant amount of interactions between applications and operation system hence provides a "noiseless" operating system and allow applications to use as many cycles as possible,
5. it provides I/O-aware load balancing between the Level-1 and the Level-2 networks, and
6. it has much less NIC cable installation and complicated cable management overhead.

It is known that challenging scientific computing requires deep data [8]. PaScal is a scaling parallel I/O architecture that provides networking capabilities to satisfy the constantly increasing computing power and the global storage/file needs. PaScal not only provides good I/O bandwidth at scale but it also scales with very little incremental cost. When federated-switch fabrics in Level-2 interconnect are scaled (FESIO architecture) the cost doesn't just increase by switch port count. It is much worse than that. PaScal is looking for a way to scale up that doesn't have the federated-switch effect on cost by taking advantage of the fact that there is non-blocking Fat-Tree or Clos tree (Level-1) in network internal the clusters. PaScal also provides a cost-effective solution to connect multiple clusters and remote Grids to a global storage in a linear scaling, load balancing, and fail-over way.

1. In order to leverage the PaScal capabilities we are currently conducting research and design in the following areas:

1. Dead Gateway detection and Recovery,

2. Applying 10-Gigabit Ethernet in I/O routing node group for supporting Petabyte-scale storage systems,

3. Using IPoIB, SDP, and SRP from OpenIB/OpenFabrics,

4. Applying PaScal to iSER based global storage,

5. Enhancing of Linux Multi-path routing with dead gateway detection capability,

6. Fault-tolerant application support,

7. Using SCTP Multi-homing fail-over [18] in file system client software for TCP/Socket connection migration,

8. Deployment of the PaScal I/O architecture using PVFS-2 , IBM's GPFS, and pNFS parallel file systems, and

9. Promoting PaScal Server I/O network design and implementation outside of LANL.

ACKNOWLEDGEMENTS

We are thankful to co-workers from LANL's HPC-3, CTN-5, and HPC-5 groups for their support on the design and implementation of PaScal Server I/O Infrastructure, Benjamin McCelland (LANL's HPC-3 Group) for reviewing and discussions, and engineers from Panasas File System for valuable discussions and sharing their storage/file system information.

7 REFERENCES

[1]. Brad Winett, "Building Fast, Scalable I/O Infrastructures for High-Performance Computing Clusters", Dell Power Solutions, Nov. 2005 Issues

[2]. Renato John Recio, "Server I/O Networks past, Present, and Future", Proceeding of the ACM SIGCOMM 2003 Workshop

[3]. Jonathan D. Bright and John A. Chandy, "A Scalable Architecture for Clustered Network Attached Storage", IEEEE/NASA 2003 conference on Mass Storage Systems & Technologies

[4]. Helen Chen, J. Decker, and N. Bierbaum, "Future Networking for Scalable I/O", Sandia national lab. Report CSIT-517-050, 2005

[5]. Andy D. Hospodor and Ethan L. Miller, "Interconnection Architectures for Petabyte-Scale High-Performance Storage Systems", Proceedings 2004 IEEE Goddard Conference on Mass Storage Systems & Technologies

[6]. Micah Beck, Terry Moore, and James S. Plank, "An End-to-End Approach to Globally Scalable Network Storage", SIGCOMM'02, August 2002

[7]. Xiao Qin and Hong Jiang, "Improving the Performance of I/O –Intensive Applications on Clusters of Workstations", Cluster Computing: The Journal of Networks, Software Tools, and Applications, Vol.8, No.4, Oct. 2005

[8]. W.T.C. Kramer, A. Shoshani, D.A. Agarwal, B.R. Draney, G. Jin, G.F. Butler, J.A. Hules, "Deep scientific computing requires deep data", IBM J Res & Dev, Vol. 48 NO. 2 March 2004

[9]. O.T. Anderson, L. Luan, C Everhart, M. Pereira, R. Sarkar, and J. Xu, "Global namespace for files", IBM System Journal, Vol. 43, No. 4, 2004

[10]. Roy S.C. Ho, Kai Hwang, and Hai Jin, "Single I/O Space for Scalable Cluster Computing", 1ST IEEE International Workshop on Cluster Computing, IWCC 1999

[11]. Henrik Abrahamsson, end etc, "A Multi Path Routing Algorithms for IP network Based on Flow Optimization", Proceedings of QofS 2002

[12]. David Nagle, Denis Serenyi, Abbie Matthews, " The Panasas ActiveScale Storage Cluster – Delivering Scalable High Bandwidth Storage", Proceedings of SC'04, Nov. 2004

[13]. Aaron E. Darling, Lucas Carey, Wu-chun Feng , "The Design, Implementation, and Evaluation of mpiBLAST", proceedings of ClusterWorld 2003

[14]. Jianwei Li, Wei-keng Liao, Alok Chounhary, etc., "Parallel netCDF: A High –Performance Scientific I/O Interface", SC'03

[15]. Heshan Lin, Xiaosong Ma, Praveen Chandramohan, Al Geist, Nagiza Samatova, "Efficient Data Access for Parallel BLAST" , proceedings of IEEE IPDPS 2005

[16]. Chris H.Q. Ding, Yum He, "Data Organization and I/O in a Parallel Ocean Circulation Model", SC'99

[17]. Phyllis E. Crandall, Ruth A Aydt, Andrew A. Chien, Daniel A. Reed, "Input/Output Characteristics of Scalable Parallel Applications", SC' 95

[18]. Ryan L. Braby, Jim E. Garlick, and Robin J. Goldstone, "Achieving Order through CHAOS: the LLNL HPC Linux Cluster Experience", LLNL document UCRL-JC-153559, May 2nd, 2003

[19]. Manish Gupta, Jose Moreira, "Overview of BlueGene/L System Software", BlueGene/L: Application, Architecture and Software Workshop, Oct. 2003

[20]. James Milano, Gary L. Mullen-Schultz, Blue Gene/L: hardware Overview and Planning, IBM RedBooks, Dec. 2005

[21]. Brent Welch, "OSD Technical Architecture – pNFS extension for NFSv4", IETF NFS-V4, IETF-62 Meeting , march 2005

[22]. Gyu Myoung Lee, Jin Seek Choi, "A Survey of Multipath rea routing for traffic engineering", Information and Communications University, Korea

[23]. ASCI Purple Statement of Work, Lawerance Livermore National Laboratory, 2002-PurpleSOW document

[24]. A. Gara and etc., "An Overview of the BlueGene/L System Architecture", IBM Journal of Research and Development, Vol49,No-2, 2005

[25]. Robin Goldstone, "The Roar of Thunder: LLNL Goes Itanium in a Big Way", LLNL UCRL-PRES-204277, May 2004

[26]. Jack Dongarra, "HPC Challenge Benchmarks and the TOP500", IEEE SC-2006, HCP Survey of Computer Architecture" talk. Nov 2006. Tempa, Florida, USA