Designing a Low Cost and Scalable PC Cluster System for HPC Environment

Laxman S. Naik
Department of Computer Engineering,
RMCET, University of Mumbai.

ABSTRACT
In recent years many organizations are trying to design an advanced computing environment to get the high performance. Also, the small academic institutions are wishing to develop an effective computing and digital communication environment. It creates problems in getting the required powerful hardware components and softwares because the high level servers and workstations are very expensive. It is not affordable to purchase the high level servers and workstations for the small academic institutions. In this paper we have proposed a low cost and scalable PC cluster system by using the Commodity off the Shelf Personal Computers and free open source softwares.

Keywords
PC cluster, parallel virtual file system, High performance computing.

1. INTRODUCTION
As the cost of today's Commodity off the Shelf Personal Computers and Workstations decreases and the performance of network hardware increases, there has been an increasing trend toward personal computer cluster systems for network services. The reasons for this trend are the good performance price ratios these systems offer, the availability of these systems, and the broad range of applications suitable for these systems.

High performance, distributed computing and computational sciences require large data sets, fast and efficient ways of getting to that data, and a security model that will protect the integrity of the stored data. In order to create enough usable space without spending large amounts of money for storage, multiple storage servers need to be used in groups. We can use some computers which are used for teaching or administration in the institution to substitute the purchase of an expensive high-level storage server.

In case of many educational institutions it is observed that most of the resources of personal computers are not utilized fully. If the personal computer is only used for web browsing, some light weight programming, and word processing, the space of its hard disk would only require 30% to 40% of its total capacity in use. The remaining unused 60% to 70% space will become idle. In this case, we can connect those computers dispersed in the institution and combine unused storage spaces by the use of Cluster technology and PVFS (Parallel Virtual file system) [2,3]. In this way, it is possible to obtain a storage capacity equal to that of a more expensive storage device and to integrate the computing power of those personal computers.

In this work, I build small-scale cluster-based network system as a low-cost alternative to traditional high-performance computing systems. The design and development of my cluster system for scalable network services is characterized by its high performance, high availability, high scalable file system, low cost, and broad range of applications. The parallel virtual file system version 2 (PVFS2)[2,7,8,10] is deployed in the system to provide a high performance and scalable parallel file system for PC clusters. In addition with PVFS2 the MPICH2 (MPI-IO implementation)[1,9] is combined for message passing.

Clusters of servers, connected by a fast network are emerging as a viable architecture for building highly scalable and available services. This type of loosely coupled architecture is more scalable, more cost effective, and more reliable than a tightly coupled multiprocessor system. However, a number of challenges must be addressed to make cluster technology an efficient architecture to support scalable services [4]. The parallel file systems are designed to provide cluster nodes with shared access to data in parallel. They enable high performance by allowing system architects to use various storage technologies and high-speed interconnects. Parallel file systems also can scale well as an organization's storage needs grow. And by providing multiple paths to storage, parallel file systems can provide high availability for HPC clusters.

2. RELATED WORK
The need of large-scale scientific and enterprise computing applications has motivated significant research in distributed file systems that can efficiently and reliably support processing large quantities of data. Distributed and parallel file systems, such as PVFS [12], GPFS [13], and AFS [14], have been developed to address the needs of a range of large- scale applications. A number of independent evaluation studies have explored various aspects of these parallel file systems.

Previous studies have shown that high-end computing workloads feature data accesses by multiple processes as well as high metadata rates, and frequent and concurrent creates and deletes. Metadata operations make up over half of some workloads. As ever larger machines with more disks are being deployed, a single metadata server is no longer sufficient to handle the workload. Some of the metadata issues can be ameliorated by using collective interfaces, such as MPI-I/O [1], at the clients. These techniques limit accesses to one client, with results broadcast outside of the file system to the other cooperating clients. Unfortunately, not all usage scenarios can limit their metadata accesses in this way, and non-collective I/O interfaces, such as POSIX, are still prevalent.

A common distributed file system architecture consists of many IO servers that store data contents, one or more metadata servers that store information about the data, and many clients, all of which are connected by a shared network. Clients typically communicate with both IO and metadata servers to

perform file system activities, and servers may or may not communicate among themselves. This architecture is used in many current parallel file system implementations. Some systems delegate operations, such as create, from the requesting client to a single server that in turn contacts other servers as required to perform the metadata and data operations. This approach simplifies the client implementation at the expense of scalability. It also requires an architecture where all servers can communicate with each other.

PVFS uses striping across IO servers to achieve high data rates, and fully distributes file, directory, symbolic link, and other metadata objects across one or more metadata servers. Operation is optimized for the case of cooperating clients, avoiding the need for mandatory defensive locking, but also not allowing for the use of client-side data caches. The server is stateless from the protocol point of view, although network connections are cached for performance. Distributing metadata among multiple servers ensures good scalability [5] but at a cost of an increased number of transactions required to perform a single operation from the client point of view. Furthermore, PVFS is designed with correctness in mind, so each of the multiple transactions is followed by a disk synchronization to ensure data stability, and relationships between the transactions forces serialization at times. In addition, these parallel file systems, as well as I/O optimizations on them, are mostly research prototypes, while our work is done on PVFS, a production-ready parallel file system widely used on Linux clusters.

3. PROBLEM DESCRIPTION

Today many organizations are trying to design an advanced computing environment to get the high performance and good results. Also, the small academic institutions are wishing to develop an effective computing and digital communication environment. It creates problems in getting the required powerful hardware components and software because the high level servers and workstations are very expensive. It is not affordable to purchase the high level servers and workstations for the small academic institutions.

The high level servers, workstations, and storage area networks are cost expensive; hence the small academic institutions are required alternative solution. These institutions must use the personal computers and the low cost hardware components to design high performance computing environment. The utilization of the available hardware components for developing the low cost alternative of HPC is one of the fundamental issues in a cluster computing technology. The large-scale scientific computation often requires significant computational power and involves large quantities of data. Scientific simulations of various physical processes, data mining of large data sets to extract business intelligence, and enterprise-level operations such as e-mail services for large organizations, are examples of modern large-scale applications that require computing infrastructure comprising hundreds of processors.

However, the performance of I/O subsystems within high-performance computing (HPC) clusters has not kept speed with processing and communications capabilities. Inadequate I/O capability can severely degrade overall cluster performance. Therefore there is a need of a low cost and scalable PC cluster system for high-performance computing environment with parallel file systems.

When building a high-performance computing cluster, the system architect can choose among three main categories of file systems: the Network File System (NFS); storage area network (SAN) file systems, and parallel file systems. In a cluster environment, large files are shared across multiple nodes, making a parallel file system well suited for I/O subsystems. Generally, a parallel file system includes a metadata server (MDS), which contains information about the data on the I/O nodes. Metadata is the information about a file —for example, its name, location, and owner. Some parallel file systems use a dedicated server for the MDS, while other parallel file systems distribute the functionality of the MDS across the I/O nodes.

4.PROPOSED SYSTEM SETUP

The small-scale cluster-based network system is designed by using off the self personal computers and some free open source software. The design and development of cluster system for scalable network services is characterized by its high performance, high availability, high scalable file system, low cost, and broad range of applications. The parallel virtual file system (PVFS2) is deployed in the system to provide a high performance and scalable parallel file system for PC clusters. In addition with PVFS2 the MPICH2 is combined for message passing. PVFS2 and MPICH2 are available at free of cost.

Our experimental setup (Fig. 1) consists of the 8 nodes cluster of Pentium IV PC's with 512MB of RAM and 40GB HDD running Red Hat Enterprise Linux 4 and connected through a Fast Ethernet switch. Each file server node uses an underlying Linux file system of type ext3. In all MPI experiments we used the MPICH2-1.0.8 implementation and PVFS-2.7.1.

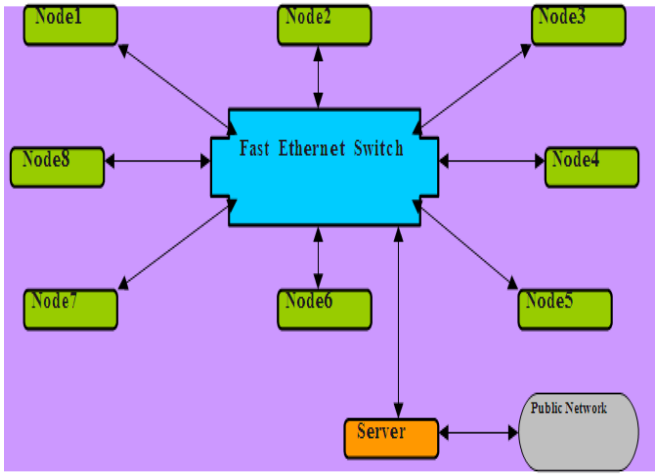

Fig 1: Experimental Setup of the system

In this work, we have used some benchmarks as well as some designed test programs to evaluate the parallel file system and the cluster. Several of the benchmarks use the MPI (Message Passing Interface) programming model, which forms the basis for many parallel programs developed for the scientific domain. MPI provides the mechanisms to communicate between and synchronize processes executing on the processors of a computing cluster.

Objective of our proposed systems is to provide an enhanced high-performance computing environment for the small organization by utilizing commodity hardware and free open source software.

4. CONCLUSION & FUTURE WORK

We can design an enhanced high-performance computing cluster system with PVFS2 and MPICH2. Each cluster node can be marked by its efficiency in communicating with the

clients and its fast response. In order to sustain the system's performance, one could raise the network bandwidth to avoid performance deterioration. While increasing the number of compute nodes is, perhaps, one of the best ways for load balancing, increasing the I/O node may also improve the writing and reading performance of the PVFS2. To optimize the performance of PVFS2, the physical memory of cluster nodes must be adjusted appropriately when the compute node has to handle many large-scale files.

Increasing the compute node to balance the connections with I/O nodes will improve the performance of PVFS2. The results also revealed that parallel access resulted in better performance and was more efficient than only with PVFS2. Both MPICH2 and PVFS2 are free software and are easy to be included in the construction of the environment for cluster network service. The overall system performance can be further improved by using powerful network devices such as Giga Ethernet.

In the future, this system can be applied for the research and development in the large range of applications at small organization. We may design the new scheduling algorithm that considers the system load in real time and the sizes of partitioning and stripping of the files for each network service.

4. ACKNOWLEDGMENTS

I am thankful to Shri. Ravindraji Mane (Chairman, RMCET, Ambav, Devrukh.) and **Dr. G. V. Mulgund** (Principal, RMCET, Ambav, Devrukh) for being very generous with their advice and encouragement. I am also thankful to my department people for their support.

For future, further improvement of Lifetime will be necessary while improving power efficiency. If a device of longer Lifetime is realized, the foot of the application spreads out greatly.

We hope that the development discussed in this paper opens up a course to practical use of OLED as lighting sources for illumination use, backlights and others.

5. REFERENCES

[1] MPICH2. http://www.mcs.anl.gov/research/projects/mpich2.

[2] P. H. Carns, W. B. Ligon III, R. B. Ross and R. Thakur, "PVFS: A Parallel File System For Linux Clusters", pp. 317-327, Proceedings of the 4th Annual Linux Showcase and Conference, Atlanta, GA, October 2000.

[3] Chao-Tung Yang, Chien-Tung Pan, Kuan-Ching Li, and Wen-Kui Chang, "On Construction of a Large File System Using PVFS for Grid," Parallel and Distributed Computing: Applications and Technologies: 5th International Conference, PDCAT 2004, Lecture Notes in Computer Science, Springer, vol. 3320, pp. 860-863, Dec. 8-10, 2004

[4] [Yi-Hsing Chang J. Wey Chen. "Designing an Enhanced PC Cluster System for Scalable Network Services". Proceedings of the 19th International Conference on Advanced Information Networking and Applications, 2005.

[5] Yifeng Zhu, Hong Jiang, Xiao Qin, Dan Feng, and David R. Swanson "Improved Read Performance in a Cost- Effective, Fault-Tolerant Parallel Virtual File System (CEFT- PVFS)", Department of Computer Science and Engineering University of Nebraska - Lincoln, NE, U.S.A Email: jiang@cse.unl.edu Department of Computer Science and Engineering Huazhong University of Science and Technology, Wuhan, China Email: dfeng@hust.edu., 2003, IEEE.

[6] Weikuan Yu Shuang Liang and Dhabaleswar K. Panda, "High Performance Support of Parallel Virtual File System (PVFS2) over Quadrics", Network-Based Computing Laboratory Dept. of Computer Sci. and Engineering The Ohio State University, 2005, ACM.

[7] F. Haddad, "PVFS: A Parallel Virtual File System for Linux Clusters", pp. 74-82, Linux Journal, December 2000.

[8] Parallel Virtual File System,Version 2, http://www.pvfs.org/pvfs2/

[9] MPICH, http://www-unix.mcs.anl.gov/mpi/mpich

[10] Parallel Architecture Research Laboratory http://parlweb.parl.clemson.edu/.

[11] Avery Ching, Alok Choudhary, Wei keng Liao, Robert Ross, and William Gropp. "Noncontiguous I/O through PVFS". In Proceedings of the IEEE International Conference on Cluster Computing, 2002.

[12] LIGON, W. I., AND ROSS, R. "Overview of the Parallel Virtual File System". In Proc..of Extreme Linux Workshop (1999).

[13] SHMUCK, F., AND HASKIN, R. "GPFS: Shared Disk File System for Large Computing Clusters". In Proc. of 2nd USENIX Conference on File and Storage Technologies (FAST) (2002).

[14] HOWARD, J., KAZAR, M., MENEES, S., NICHOLS, D., SATYANARAYANAN, M., SIDEBOTHAM, R., AND WEST, M. "Scale and Performance in a Distributed File System". ACM Transactions on Computer Systems 6, 1 (February 1988), 55-81.

[15] PVFS2 Development Team, Parallel Virtual File System, Version 2, September 2003.

[16] PVFS2 Development Team, A Quick Start Guide to PVFS2, Last Updated: July 2007.