Development of Fibre Channel Disk Clusters

Final Report
for Period September 2, 1998 – March 17, 1999

RECEIVED
APR 04 2000
OSTI

W.L. Dunn, J.R. Justice, T.D. Stockert, A.R. Barker, and A.M. Yacout

Quantum Research Services, Inc.
5410-W Apex Highway
Durham, NC 27713-9434

August 1999

We have no objection from a patent
standpoint to the publication or
distribution of this material.

Office of Intellectual
Property Counsel
NRC Field Office, Chicago

3/9/00
Date

Prepared for
The U.S. Department of Energy
Under Award No. DE-FG02-98ER-82648

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Department of Energy, nor any of their employees, nor any of their awardees, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product or process disclosed or represents that its use would not infringe privately-owned rights.

Abstract

This report documents the accomplishments of a Phase I project whose purpose was to demonstrate feasibility of developing inexpensive and fast data storage using multi-host Fibre Channel disk clusters. In Phase I, a working file system called ZFS was developed and tested. The ZFS approach was designed to be suited for high energy physics applications, but is general and flexible enough to be useful for other high-volume applications. The ZFS approach, which borrows from the networking concept of cut-through routing, uses Linux boxes and disk clusters in a Fibre Channel – Arbitrated Loop architecture. In ZFS, file locking and other meta-data level operations are carried out over the primary data network, after which all data are sent directly over a Fibre Channel between the workstation and the disk cluster. No intermediate server is required. Substantially higher throughputs than in traditional networked disk architectures have been demonstrated. The ZFS architecture is described and tests of the first implementation of ZFS at Fermilab are discussed. The current system is implemented for Linux and is being optimized for Fermilab's needs, but extensions to other operating systems and other data-intensive applications are clearly foreseen.

Introduction

Upcoming experiments at Fermilab, SLAC, Brookhaven, and elsewhere in the US and at CERN in Europe (where US scientists are heavily involved in international collaborations) will generate datasets as large as a Petabyte (PB), i.e., 10^{15} Bytes or 1,000 terabytes (TB), in the near future. Large data volumes are also encountered in the Human Genome Project, in satellite imagery and space exploration, in digital Geographic Information System (GIS) databases, in electronic patient records, and in other data warehouse applications. Efficient, high-speed, distributed access to these data are required from a variety of platforms and applications. Throughput from data storage must be consistent, and data must be accessible from any platform in current use or from platforms that may be adopted in the coming years.

As an example, we summarize in Table 1 anticipated specifications of the CDF and D0 collaborations for the upcoming Run II at Fermilab. Together the experiments will generate over 750 TB of raw data per year. This vast amount of data cannot be stored on-line and sophisticated data handling systems must be implemented. CDF has elected an object-oriented persistent data format based on the ROOT I/O (Murat, 1997; Brun, Murat, and Rademakers, 1998) with about 2 GB file sizes. D0 intends to upgrade the Sequential Access Model (SAM) used in past experiments (Kowald, et al., 1997), with slightly smaller (0.5 – 2 GB) file sizes. The raw data will be processed into event data units (EDUs) of various sizes; the smallest EDUs are expected to occupy about 5-10 kB per event and will be stored on-line for distributed access. In addition, the appropriate meta-data (data about the data), which is slightly smaller in the object-oriented ROOT model, also must be kept on-line. This translates into a need for almost 30 TB on-line data storage for CDF and almost 15 TB for D0, per year.

Table 1. Fermilab Run II general specifications.

	CDF	D0
Number of events	1.8×10^9	1.2×10^9
Raw data size	250 kB/event	250 kB/event
Annual raw data store	500 TB	280 TB
File size	2 GB	0.5 – 2 GB
Smallest EDU size	10 kB	5-10 kB
Meta-data	10 – 50 B	50 B
Annual on-line data store	28 TB	14 TB

A need exists for efficient, heterogeneous, high-bandwidth disk file systems. We sought to construct in Phase I the rudiments of a robust file system for access to 1-10 TB of data for Fermilab's Run II experiments by a heterogeneous assembly of computing platforms (Linux, DEC, UNIX, AIX, IRIX, SunOS, and others). Our successful Phase I project looked at the use of Fibre Channel (FC) disk technology running with commodity hardware as a means to achieve low-cost, high-bandwidth data access. As we will describe, we developed a functional Linux-based FC file system called ZFS, which achieves much higher bandwidth than traditional network file systems such as NFS.

One solution for providing shared access to large datasets is to attach disk arrays to large SMP machines; this solution, however, is expensive and has limited scalability. Our approach is to build a file system that integrates commodity hardware (Linux boxes) and FC drives; this is relatively inexpensive and has flexibility and scalability. Multiple hosts can simultaneously attach to disk clusters via FC and each host sees Native disk access rates. Fibre Channel – Arbitrated Loop (FC-AL) technology allows up to 126 distinct hosts and disks to be so attached. In the future, FC switching fabrics will significantly expand this limit.

Unfortunately, though current FC hardware fully enables multi-host attachment, existing file systems such as ext2 in Linux intrinsically assume single-host access to the disks and thus cannot be used without modification. Network file systems, such as NFS, rely on a native file system on a single host server attached to a disk. Concurrent access by multiple clients to such an NFS file system is possible because all modifications to data on disk are serialized through the single host server, and all data to and from the clients flow over an Ethernet network. Users of NFS are well aware of the data-rate limitations that this scheme imposes. By taking a distributed approach to storage using high-speed FC, ZFS can provide the necessary bandwidth, fault tolerance, and platform independence.

Purpose of the Phase I research

As stated in the Phase I proposal, the objective of the Phase I project was "...to demonstrate that multi-host Fibre Channel clusters can provide an efficient means of data storage and access." This objective was fully achieved during Phase I. Quantum developed an operational file system – ZFS – that was tested at Fermilab. The Phase I system was developed on a two-node Linux-box system but was then performance tested on a three-node system. The Phase I ZFS software contains file system utilities (fdisk, makefs, fsck.zfs, and mount), kernel interfaces, a volume manager, and a lock manager. Both client and server are multithreaded applications and integrate with the Linux 2.2 development kernel.

Description of research carried out in Phase I

We coordinated our work closely with colleagues in the Computing Division and the D0 and CDF collaborations at Fermilab. In meetings with our Fermilab collaborators, we developed a set of requirements for ZFS. These can be succinctly stated as follows:

- Traditional UNIX style file access
- Files can span multiple drives
- Failures contained (single disk failure affects only one logical volume)
- Distributed lock manager (two machines guard every resource)
- Single writer, multiple readers per file

Although not all of these requirements were fully incorporated in Phase I, the Phase I ZFS was designed so that all requirements would eventually be realized. Design and construction of a functioning, testable file system within a six-month Phase I project was a major accomplishment.

Comparison to traditional approaches

To appreciate the ZFS Phase I approach we contrast it with the traditional network approach, which is schematically represented in Fig. 1. A user (client) communicates through a standard or enhanced Ethernet connection with a server, who then routes the user's request to the appropriate drive, from where the desired data are retrieved and sent back to the user over the network. The advantages of this approach are that it is fully tested and functional now and it exhibits interoperability among diverse clients and servers, particularly with NSF as the client/server protocol. The primary disadvantage, however, is that the network throughput, which is much slower than disk throughput (even for fast Ethernet), limits total bandwidth. The high native disk speeds are compromised by the fact that the data must be transmitted over the much slower Ethernet network.

Figure 1. A traditional network file system architecture. Typical fast Ethernet rates are of order 5 MB/sec.

The ZFS approach borrows from the networking concept of cut-through routing. It uses Linux boxes and disk clusters in a FC-AL architecture, as shown in Fig. 2. In ZFS, file locking and other meta-data level operations are carried out over the primary data network, after which all data are sent directly over a Fibre Channel between the workstation and the disk cluster. No intermediate server is required and substantial improvement in throughput is possible. As before, a user communicates with the server over the Ethernet connection. However, once the server recognizes the client, the client is free to access the data directly over the high-speed FC-AL interconnect. Thus, very little traffic goes over the slower Ethernet connection and most of the data transfer takes the high-speed FC-AL path. Much higher bandwidth is achieved and

throughput in principle approaches Native disk speed. Another advantage is that most of ZFS is implemented in software and so can be used with hardware from almost any vendor. This means, for instance, that low-cost Linux PCs can be used, as was done in Phase I. Other platforms will work as well, once appropriate interfacing is performed.

Figure 2. The ZFS system architecture. In principle, this architecture can achieve native disk rates of over 10 MB/sec.

Overview of ZFS

ZFS is a set of programs running under Linux OS that allow clusters of machines fast, concurrent access to FC disk drives. ZFS provides a locking mechanism on top of a traditional local file system to ensure data and metadata coherency in a multiple-access environment. This locking is independent of actual data access, enabling the use of fastest available data path. In our case, we allow access to the FC ring from any machine on the cluster, not just the server. Below, we describe the ZFS system in its current prototype state, summarize performance tests that compare it with the ubiquitous NFS file system, and briefly describe the envisioned completed ZFS system.

System Architecture

The server exists entirely in user space, with the main program maintaining state for the cluster, arbitrating and coordinating connections. The client consists of kernel modifications, a kernel module, and a user-space network client. We used the QLogic QLA2000 PCI interface card, , ISP2100 SCSI-FCP Linux driver. Basic tests to confirm operation were conducted on the development system, shown in Fig. 3; additional performance tests run on a three-node system at Fermilab are described below.

Figure 3. The Phase I development system. The two FC drives were linked through a hub loaned by Emulex and a 4-bay chassis purchased from Storage Path.

Performance Tests

Our initial Phase I technical objectives were to eventually obtain transfer rates of 80% of Native disk speed or better, demonstrating the feasibility of our approach. In fact, we obtained rates in excess of 97%. We benchmarked reading rates using 'dd bs=1024k' commands against 200 MB files. The average sustained rates are tabulated below:

Number of nodes	Native (MB/sec)	ZFS (MB/sec)	ZFS/Native
1 node	9.7	9.6	99%
2 nodes concurrently	7.7	7.5	97%
3 nodes concurrently	6.1	6.3	100%

The uncertainties in the rates are estimated to be of order 5% or more, and so these rates are approximate. Nevertheless, ZFS consistently exceeds 90% of Native rates.

Results obtained using the "Bonnie" benchmark are tabulated below. Bonnie provides an estimate of both random and sequential access rates in both block and character modes. We only list block transfer rates from Bonnie because the character access rates are known to be difficult to reproduce and sensitive to the version of libc. For these tests we used three nodes attached via fast Ethernet to a hub. All nodes were attached via a FC hub to a "JBOD" FC disk chassis with Seagate Barracuda drives (ST39173FC, 9 GB). All nodes were running the 2.1.131 Linux kernel.

	Block Output (KB/sec)	Block Input (KB/sec)
Native ¹	12,559	12,417
NFS	7,519	1,257
ZFS – 1 node ²	8,884	9,880
ZFS – 1 node ³	7,066	10,724
ZFS – 2 node ⁴ ; node 1	8,564	10,042
ZFS – 2 node; node 2	6,720	2,341

Even though ZFS has not yet been optimized for performance, it generally outperforms NFS in these tests. We note that the sum of the 2-node block output tests is slightly higher than Native; we suspect this results because the various Bonnie sections on the two nodes do not overlap exactly. We ran more definitive stripe set tests and achieved the following impressive block I/O results.

	Output	%CPU	Rewrite	%CPU	Input	%CPU
Native	64,767	61.0	19,946	45.9	52,518	40.8
Native SMP	64,468	62.3	19,799	49.4	54,818	48.4
ZFS ⁵	33,283	34.8	17,685	38.7	48,915	35.6
ZFS SMP	57,124	67.5	19,478	48.1	54,475	47.6

In summary we can say that (SMP) ZFS achieves 90% of Native writing rates and 100% of Native reading rates on striped file systems tuned to deliver 64 MB/sec

¹ Node with local ext2 file system

² Node running both ZFS client and server

³ No Node running ZFS client, remote server

⁴ Node #1 runs ZFS client and server, node #2 runs client; nodes 1 and 2 had different CPU speeds and memories

⁵ one machine, server on a different node

Budget and Schedule Issues

The project was completed on-time and within budget; there are no unexpended funds at the end of the project.

Conclusion

The Phase I objectives were fully met; in fact they were exceeded in the sense that transfer rates of 97% of Native disk speed were achieved whereas the Phase I goal was 80%. A presentation (Justice, et al., 1999) was made after the end of the project at the Extreme Linux track of Linux Expo. This presentation was very well received and has generated interest in ZFS from several groups. The general benefit of ZFS file system is availability of efficient (relatively fast and inexpensive) mass storage that is accessible from a wide variety of platforms. This benefit accrues to high energy physics laboratories, but also to other users who must maintain on-line access to large databases. The immediate beneficiaries are likely to be reconstruction and/or analysis farms for the Run II D0 and CDF experiments at Fermilab. Run II is currently scheduled to be initiated in early 2000 and will run for several years. By the year 2002 datasets of order 1-10 TB will need to be stored for heterogeneous access and analysis by collaboration physicists.

Since the technology is for data storage and access, it has very broad application to many other applications involving massive databases. These include but are not limited to GIS, medical data records, satellite imagery, video-on-demand, and DNA sequencing of the human and selected model organism genomes (Collins, et al., 1998). As one example, GIS technology provides the capability to capture, store, manage, analyze, and display multiple layers of data that can be associated with geographic locations. Government agencies and large commercial entities use GIS for land use, land planning, epidemiology studies, construction project management, census purposes, and so forth. Ortho-photography of selected geographic regions is digitized and stored with other related data for ready access as needed by GIS users. Since various agencies seek access to the data, heterogeneity is an important issue.

Once a Linux-based version of ZFS is developed, it is all but certain that other vendors will cooperate with Quantum to develop versions for their operating systems. This confidence is justified by the fact that, as we describe in the next section, ZFS achieves almost an order of magnitude higher data transfer rates to the user than do traditional Ethernet network file systems. Thus, the ultimate benefit will be a new paradigm for distributed file systems, which will have very broad reach throughout the fast-growing information technology world. It is noted that Quantum has already received funding from Emulex Corporation to build a Linux SCSI driver for their Fibre Channel Host Bus Adapter. This funding is a direct result of the Phase I fibre channel cluster project.

References

- Brun, R., P. Murat, and F. Rademakers (1998), ROOT-based data handling system for CDF in Run II, CDF-note 4497, February 26, 1998.
- Collins, F.S., A. Patrino, E. Jordan, A. Chakavarti, R. Gesteland, L. Walters, and the members of the DOE and NIH planning groups, New Goals for the U.S. Human Genome Project: 1998-2003 (1998), *Science* **282**(5389):682-689.
- Justice, J.R., T.D. Stockert, A.D. Barker, A.M. Yacout, W.L. Dunn, D. Holmgren, and D. Petravick (1999), Early Experience with the ZFS Fibre Channel Disk Cluster File System, Proceedings of the ExtremeLinux Track, Linux Expo, 18-22 May 1999, Raleigh, NC.
- Kowald, K., L. Lueking, K. Ruthmansdorfer, J. Schellman, and V. White (1997), Sequential File Access Model Evaluation and Conceptual Design for Run II Data Management, Draft D0 and CDF note, October 13, 1997.
- Murat, P. (1997), On the Possibility to Use ROOT Data Model for the CDF Data Storage Needs in Run II, CDF-note 4412, December 8, 1997.