A High-Performance Storage System for the LHCb Experiment

Juan Manuel Caicedo Carvajal, Jean-Christophe Garnier, Niko Neufeld, and Rainer Schwemmer

Abstract—The LHCb experiment uses a single, high performance storage system to serve all kinds of storage needs: home directories, shared areas, raw data storage and buffer storage for event reconstruction. All these applications are concurrent and require careful optimisation. In particular for accessing the raw data in read and write mode, a custom light weight non-POSIX compliant file system has been developed. File serving is achieved by running several redundant file servers in an active/active configuration with high availability capabilities and good performance. In this paper we describe the design and current architecture of this storage system. We discuss implementation issues and problems we had to overcome during the hitherto 18 months running period. Based on our experience we will also discuss the relative advantages and disadvantages of such a system over another composed of several smaller storage systems. We also present performance measurements.

Index Terms—Data acquisition, data storage, RAID, storage area networks (SANs).

I. INTRODUCTION

THE Large Hadron Collider beauty (LHCb) experiment [1] is one of the four experiments around the Large Hadron Collider (LHC). The aim of the project is to measure the decay of B-hadrons produced during the collisions of protons in the accelerator.

The events registered by the detector are obtained by specialized Data Acquisition boards and are then sent to a high-level trigger (HLT) farm, where they are processed and reconstructed. In the HLT farm the events are filtered in order to select only the interesting ones for physics analysis. After this selection, the event rate is reduced considerably. From a rate of 1 MHz of events received, the trigger algorithms in the HLT produce accepted events at a rate of 2 kHz, with fluctuations up to 5 kHz. These chosen events are written to the storage system that is found at the experiment site. The final part of the data acquisition process consists of moving the data to the remote storage center at the main CERN's computing facilities.

Alongside with the event recording, the experiment data storage is used for handling other storage needs, such as home directories, shared areas for software and libraries, event reconstruction and backups of the database used by the internal applications. All these different use cases lead to the need for

careful configuration and optimisation in order to guarantee the requirements in terms of bandwidth, latency and high availability.

In this paper we discuss the architecture and the performance of the storage system for the LHCb experiment. Sections II–V contain the requirements for this system, the current architecture, performance measurements and possible improvements.

II. REQUIREMENTS

The storage system must fulfill all the storage needs of the LHCb experiment: home directories, shared areas, raw data storage and buffer storage for reconstruction under the following requirements of throughput, high availability, fault tolerance and scalability.

- 1) For the event data acquisition process, the storage system must be able to sustain a writing throughput that ranges between 70 MB/s and 175 MB/s, since the size of an event is approximately 35 kB. The event data are sent from the HLT farm to a group of streaming servers. Here the events are grouped into “streams” according to some physical characteristics of the collision and then written using a custom lightweight file system to achieve the required performance. The available bandwidth to the remote storage center must be at least twice the writing rate. This is necessary in order to guarantee that there is enough free space available while the data acquisition and moving processes are executed in parallel.
- 2) The system must be used for storing the software used in the LHCb cluster, along with the home directories and group files for all the users of the LHCb collaboration. The current setup includes approximately 1100 Linux servers, farm nodes and desktops, and about 100 Windows servers and desktops. The system must guarantee the performance necessary for executing the applications and low latency and access times for handling a larger number of operations on small files, concurrent access and also several file system metadata operations, such as getting the attributes of a file. For the nature of these operations, it is hard to specify the requirements in more precise terms of throughput and latency.
- 3) The system must store the backups of the databases of the experiment. We perform these backups daily and we move them periodically to the remote storage center. The total space required is approximately 4 TB.
- 4) The system must be tolerant to failures in the network infrastructure, the servers and the storage components in order to guarantee the required availability levels. Unexpected downtime meaning data loss is not allowed, not only for

Manuscript received May 22, 2009; revised July 31, 2009. Current version published April 14, 2010. This work was supported by a Marie Curie Early Stage Research Training Fellowship of the European Community's Sixth and Seventh Framework Programmes under Contract Nos. MEST-CT-2005-020216-ELACCO and PITN-GA-2008-211801-ACEOLE.

The authors are with CERN, European Organization for Nuclear Research, Geneva CH-1211 23, Switzerland (e-mail: juan.manuel.caicedo.carvajal@cern.ch).

Digital Object Identifier 10.1109/TNS.2009.2037816

Fig. 1. Architecture of the storage system.

the experiment data but also for the reconstruction of the events, for the backup process and for handling the user home directories and the files in the shared areas. The goal for the availability during data taking is 99.99%, taking into account that the duty cycle of the detector is 12 hours per day on average.

- 5) It must be possible to upgrade the system in order to support the growth of the experiment and meet its new requirements of capacity and throughput.

The LHCb experiment uses an independent network within CERN. In addition to the data taken by the data acquisition (DAQ) process, the software used for the trigger and the control of the experiment is recorded in the storage system. The members of the collaboration start sessions on the servers located on the network, which allows collaboration members remote access. As data becomes available at the CERN main site, it can be analyzed using CERN's computing resources.

III. CURRENT DESIGN

The storage system for the LHCb experiment is composed of various elements. First, a storage area network (SAN) is configured over fibre channel with a high performance storage appliance connected to it. Then there is a cluster of servers that has access to the share-disk SAN. Finally, specific areas of the storage are shared using network file systems (NFS and CIFS), and a service is provided for writing event data directly.

In the rest of this section, we present an overview of each of these components. Fig. 1 illustrates the existing architecture.

A. Storage Appliance

We use a couplet of high performance storage controllers, two S2A 9900 from DataDirect Networks. The controllers are connected together to guarantee high availability. Each controller has access to 7 disk chassis with 10 disks each. One of the chassis consists solely of small fast SAS disks while the other 6 are built from large SATA drives. Currently, the total capacity of the system is approximately 44 TB, and more disks can be added when the system grows.

The RAID groups are configured with a scheme called *directRAID*, using 8 disks for data, 1 for parity and the last one as a hot spare disk. This scheme combines the advantages of RAID 3 and RAID 0. Like RAID 3, a dedicated disk is used for storing parity information and, like RAID 0, data are striped across the disks in the same group and also across different groups when

Fig. 2. Example of the LUN partitioning scheme.

a *logical unit* of the storage array (LUN) is created across multiple groups.

One important advantage of *directRAID* scheme and the storage controller is the resilience in the reconstruction of a disk in the event of a failure. When there is an error in one of the physical disks, the storage controller rebuilds the disk's data in a spare disk using the parity information. In this case the controller must dedicate a part of its throughput to perform this process, nevertheless its effect over the general performance is negligible [2].

All LUNs on one RAID group have to be owned by one of the two controllers. If a LUN is accessed through the non owning controller, there is a small performance penalty in the form of latency to pay. This does not have any serious effects on continuous, large data streams, but severely cripples metadata and small file operations.

In order to improve the overall performance of the system, the LUNs have been partitioned as follows. The DAQ area uses 4 RAID groups, managed by both controllers. These groups contain the LUNs for storing the data and the file system metadata as well, and they are not used by any of the other storage areas.

The rest of the RAID groups are used to store the data of the other areas, such as home directories, software, backups, etc. In these cases, the file system metadata is stored in the special RAID group that consists of SAS drives.

In both cases, the ownership of the RAID groups is distributed between the two controllers, which means that each one is in charge of the half of the RAID groups. This separation is designed to avoid that operations in one controller can affect the performance of the other. This can be a problem when, for instance, the throughput of the DAQ process is reduced significantly because of a high number of operations in the user or software areas. Fig. 2 presents an example of the LUN partitioning scheme.

B. Storage Cluster

A cluster of servers has access to the disks shared by the storage controllers. This cluster is connected to the SAN and provides services to the clients for reading and writing data to the different areas.

The cluster is composed of by 4 Dell PowerEdge 2950 servers with 8 GB of RAM, running Scientific Linux CERN 5 (SLC5). The servers are configured in an active/active setup, this means that all the members of the cluster are operating and can process requests from the clients at the same time. The configuration of the cluster is managed using the Linux High-Availability

package [3]. The servers have access to the SAN via redundant connections to the fibre channel network. The details about the network infrastructure are explained below.

The servers access the storage appliance using StorNext, a shared-disk file system developed by Quantum [4]. This file system provides a high level of performance along with ease of maintenance. The GFS [5] file system was also considered, but we did not find it superior in terms of robustness and manageability; nevertheless one important feature that GFS has over StorNext is the symmetry in the roles of the nodes of the cluster: all of them can be used for data and file system metadata operations as well.

In order to be able to have better organizational separation, several file systems were set up. Each file system is used for a specific storage need, such as the raw data, home directories, the shared files, among others; and they have been configured to access the specific areas of the storage array following the partitioning scheme described above. In the store nodes these file systems run as independent processes that are controlled by the StorNext management daemons. All the file systems are enabled in all the nodes, although they can be controlled individually.

The main advantage of having different file systems is that they can have independent configuration parameters that can be used for improving the performance or for setting different administration policies on the storage areas. For example, the usage quotas may be different for the home directories than for the shared areas or disabled for the raw data. Further, the separation of file systems allows us to perform maintenance tasks and data integrity checks on a specific storage area.

C. Network

The storage appliance and the cluster of servers are connected by a high speed Fibre Channel Network. This configuration allows high speed transfers and provides high availability by enabling redundant connections between the storage controllers and the cluster nodes.

The network hardware used is the following:

- host adapters QLogic ISP2432 based 4 Gb controllers;
- Brocade Silksworm 200E Switches, with 16 4 Gb ports;
- Brocade Silksworm 300 Switches, with 8 8 Gb ports and 16 4 Gb ports.

The network is configured with redundant paths from the storage appliance to the cluster. This provides the high availability required to access the storage, avoiding interruptions caused by failures in the network infrastructure, the cluster of servers or the storage controllers.

D. Event Writer Software

For accessing the events of the Data Acquisition process in read and write mode, a custom light weight non-POSIX compliant file system has been developed. For recording data the *Event Writer Software* is running on a cluster of *streaming servers*, where the event data are categorized into streams and then recorded in the storage system. There is one inclusive stream which contains the data from all the detectors and, additionally, a low rate special *express* stream, which contains events suitable for calibration purposes.

This software provides services of load balancing and fail over across all the writing processes and thus guarantees that the event data are written persistently to the storage system [2], and it is implemented using the *Gaudi* framework [6].

The Event Writer maintains the required throughput of the data acquisition process by aligning the data and writing data with Direct I/O. The event data must be aligned to precise boundaries, such as the page, sector or the stripe boundary, in order to access the storage devices efficiently. Although the file system provides mechanisms for deferring writing operations and grouping them into more efficient units, in data acquisition systems these algorithms do not adapt well to the kind of required workload. Additionally writing data directly into the devices, bypassing the operating system cache, improves the throughput of these operations. This is mainly because of the use of the *O_DIRECT* flag, which allows to write data directly from userspace memory to the disk, instead of copying it to the kernel space in every write operation [2].

E. Clients

The other areas of the storage, such as the home and group directories and the shared areas for software, are exported via NFS. This file system is installed on all the servers of the storage cluster and the clients are set up with the corresponding configuration for accessing the shared directories.

There are approximately 1100 Linux servers, including farm nodes, servers, and desktops. Additionally, about 100 servers and desktops running Windows also need access to the storage system.

F. Discussion

The main motivation for using one integrated storage system is the ease of maintenance. The storage system uses several kinds of components, which can be used for different purposes and that share the redundancy and the high availability mechanisms. Our experience has shown that it is easy to manage one disk array, a single fibre channel network and one type of file system for accessing the storage. Further, an upgrade to the system can benefit all the storage areas; this is convenient when it is necessary to augment the disk capacity or the network bandwidth.

Nevertheless, the major disadvantage is that a general outage affects all the storage areas possibly causing data loss and preventing to run the software for controlling the experiment. This risk has been reduced by using redundant high performance components and implementing a high availability schema in order to isolate failures and allow to run the system even when some storage areas are not accessible.

During the period since the storage system has been running, the problems that we have experienced were limited to one storage area, without affecting the whole system or other parts of it. In these situations, the failover mechanisms proved to be useful, although manual intervention was still necessary to restore the configuration of the system.

IV. PERFORMANCE MEASUREMENTS

The following tests were performed with the aim of validating the requirements of the storage system. The tests were made

Fig. 3. Performance of read, write and mixed operations while accessing sequentially a file in the DAQ area.

using the hardware described above, except that only one server of the storage cluster was used. The tools used were IOZone [7] and Bonnie++ [8].

A. Access to the DAQ Area

In the first test we simulate the workload produced by the data acquisition process, where the events are recorded sequentially in large files and then those files are read to be moved to the remote storage center. The size of the files can be configured in the Event Writer software; the current size used is 2 GB, which is an appropriate size for transferring the files later to the remote storage center in the main CERN's computing facilities.

We measure the performance of the storage system with IOZone configured to sequentially read and write a 2 GB file and using different values of the record size parameter, ranging from 64 kB to 16 MB; disk controllers require using powers of two as record sizes so only those values were used. The record size parameter is used to specify the amount of bytes that are written in every operation. Fig. 3 presents the results of this test. The performance of reads improves significantly with large record sizes, starting from 1 MB. Nevertheless, this parameter does not affect writes, this may be the result of data alignment and internal buffers used by the file system.

Further, this test also evaluates the *write-back cache* of the controllers. When enabled, the controller sends a successful reply as soon as the data are written into cache, delaying the real writing operation which is later performed in the physical discs. The performance increase is clearly visible in Fig. 3, specially with large record sizes. As expected, the write-back cache does not have an effect on other write operations.

However, the use of the write-back cache can lead to data corruption problems if the required techniques for redundancy

Fig. 4. Concurrent processes performing read and write operations in the DAQ area.

are not used. The controllers are connected to uninterruptible power supplies (UPS), even though data loss or corruption can occur if there is a power failure before the data have actually been written to the disks. To prevent this situation, storage controllers provide redundancy mechanisms like cache coherency among other controllers and battery backed memory.

B. Concurrent Access

To measure the performance of the system when multiple operations are performed concurrently, we ran a test with several processes of IOZone concurrently reading and writing a 2 GB file in the DAQ area. Other configuration parameters were not modified, the write-back cache was enabled and the record size chosen was 512 kB. All the following tests use this configuration.

Fig. 4 presents the performance of read and write operations being performed concurrently by several processes in the same machine. The concurrency level affects the total throughput of the operations. Although the aggregated performance improves when more processes are running this relationship is not linear, and for some cases little improvement in the performance can be seen.

C. Mixed Operations

The data acquisition operations must not be significantly affected by other activity performed in the storage, such as frequent file system metadata operations in the home directories and in the software area. To verify this requirement we performed a test running Bonnie++ configured for continuously creating files and directories in the storage areas for home directories and software, thus producing a high number of metadata operations. Simultaneously, a test is performed with IOZone for writing a file in the DAQ area in order to measure the effect of the concurrent operations. The results of this test are presented in Fig. 5.

The performance of the DAQ area is affected by a high number of metadata operations in other storage areas. Performance decreases for write and mixed operations as well,

Fig. 5. Concurrent operations on the DAQ area while simultaneously performing intensive file system metadata accesses on other areas.

although the number of concurrent processes does not seem to reinforce this effect. Nevertheless, the throughput of writing operations in the DAQ area is still above the requirements for the operation of the experiment.

V. FURTHER WORK

There are several aspects of the design which can be improved. For instance, a specialized storage controller can be used for handling the file system metadata operations. This could improve the performance for accessing data in the shared areas (software and home directories) and could also reduce the effect of mixed workloads in the throughput of the data acquisition process. Another alternative that is being considered is to use solid-state drives (SSD) for the RAID group used for storing the file system metadata information.

Additionally, the way how the data are shared among the different clients can be simplified by using a distributed file system like Lustre [9] and installing it in the storage nodes and in all the clients. This will eliminate the need to export the data using NFS, and the clients can have access to the high availability and performance features offered by Lustre.

Other experiments at CERN have implemented similar systems, although their requirements differ substantially, especially concerning the storage areas needed. For example, the ALICE

experiment [10] uses a SAN over Fibre Channel with redundant disk arrays for the transient storage of the data acquired from the experiment [11], but it is not used for other storage needs, such as shared areas for software or home directories. Their setup also includes the StorNext file system and it has been configured in a way that can be an improvement for the storage system for the LHCb experiment.

They take advantage of a feature of the StorNext file system called *affinity* which is used to physically separate reading and writing streams in the same file system, thus the disk operations can actually be performed on different devices, increasing the total throughput of the system. This technique can be evaluated to improve the performance of the storage system for the LHCb experiment, specially for the storage areas dedicated to the data acquisition.

VI. CONCLUSIONS

In this paper, we present the storage system for the LHCb experiment, its special requirements, its current architecture and performance measurements. The diversity in the access patterns of the use cases poses design challenges that are overcome by defining a special partitioning schema for the storage controllers and also by using a specialized file system for accessing the raw data from detector.

The architecture presented fulfills the requirements of scalability, high availability and performance needed by the LHCb experiment. The performance tests showed that the throughput of reading and writing data from the detector is within the required limits, even if it is affected under certain operations in other areas of the storage system.

REFERENCES

- [1] *LHCb: Technical Proposal*, ser. Tech. Proposal. Geneva, Switzerland: CERN, 1998.
- [2] S. Cherukwada and N. Neufeld, "A high performance online storage system for the LHCb experiment," *IEEE Trans. Nucl. Sci.*, vol. 55, no. 1, pp. 278–283, Feb. 2008.
- [3] Linux high-availability project. [Online]. Available: <http://www.linux-ha.org/>
- [4] StorNext file system. [Online]. Available: <http://www.quantum.com/stornext>
- [5] Red Hat global file system. [Online]. Available: <http://www.redhat.com/gfs/>
- [6] G. Barrand *et al.*, "GAUDI: The software architecture and framework for building LHCb data processing applications," in *Proc. CHEP*, 2000, pp. 45–55.
- [7] IOzone Filesystem Benchmark. [Online]. Available: <http://www.iozone.org/>
- [8] Bonnie++. [Online]. Available: <http://www.coker.com.au/bonnie++/>
- [9] Lustre, a network clustering file system. [Online]. Available: <http://www.lustre.org>
- [10] *ALICE: Technical Proposal for a Large Ion Collider Experiment at the CERN LHC*, ser. LHC Tech. Proposal. Geneva, Switzerland: CERN, 1995.
- [11] F. Carena *et al.*, "The ALICE DAQ online transient data storage system," in *J. Phys.: Conf. Ser.*, 2008, vol. 119, p. 022016.