CASTOR Project Status

J.-P. Baud[1], *O. Bärring*[1], *J.-D.Durand*[1]

CERN, CH - 1211 Geneva 23, Switzerland

Abstract

In January 1999, CERN began to develop CASTOR* ("CERN Advanced STORage manager"). It is an evolution of SHIFT, which has been used at CERN for all experiments in the 1990's. CASTOR is more scalable than SHIFT and should be able to handle data for the overlapping runs of the NA48 and COMPASS experiments, the latter starting in 2000. The sustained data rate for NA48 is $>$ 25 MB/s while for COMPASS it will be $>$ 35 MB/s. Scalability issues are solved by using multi-threading and database technology instead of a flat file for the stager catalog. The first tests show that we obtain the full bandwidth of the drives and of the network even using PCs running Linux as low cost disk/tape servers. The software can be installed on any platform running Unix or Windows/NT. The CASTOR user interface is fully backward compatible with SHIFT and the product itself is very modular, so we will be able to put it in production component by component over the winter 1999/2000. Note that the modular and layered nature of CASTOR should permit easier sharing of software components between HEP sites. The functionality has been also greatly improved by providing a *Hierarchical Storage Manager* facility: a name server has been implemented, it is capable of using different databases (Cdb, Raima or Oracle) and a Tape Volume Allocator is currently being developed. The first tests of the name server show that we get excellent response time even for administrative tasks like listing large directories. A "Mock Data Challenge" test (7 days at 100MB/s sustained, i.e. around 60TB) will be attempted for ALICE in February 2000 using HPSS and CASTOR. Preliminary tests while setting up part of the configuration for the ALICE Data Challenge reached close to the required data rate.

Keywords: Mass Storage, Backend Store, Hierarchical Storage Manager (HSM)

1 Introduction

In January 1999, CERN began to develop CASTOR ("CERN Advanced STORage manager"). It is an evolution of SHIFT (Scalable Heterogeneous Integrated FaciliTy) which has been used for tape I/O by all the CERN experiments in the 1990's.

CASTOR is more scalable than SHIFT and should be able to handle data for the overlapping run of the NA48 and COMPASS experiments, the latter starting in 2000. The sustained data rate for NA48 is $>$ 25 MB/s while for COMPASS it will be $>$ 35 MB/s. CASTOR would also be a prototype for the software to be used as a cache and a backend storage manager in the LHC era.

After one year development, it is being put in production at CERN and will be stressed by the yearly ALICE data challenge at the end of February. The required data rate is 100 MB/s sustained for one week (1000 STK Redwood cartridges).

The following sections will describe more precisely the objectives of CASTOR and the main components of the system. The performance achieved in the early tests will be shown. Then the hardware configuration used for the ALICE Data challenge will be presented.

* `http://wwwinfo.cern.ch/pdp/castor/`

Finally the status of the project will be given.

2 CASTOR objectives

CASTOR is a disk pool manager coupled with a backend store: the data residing on tertiary storage (magnetic tapes now, but commodity devices like DVDs could be used also) is accessed via a disk cache. The name space, the tape pool and the migration between secondary storage and tapes is normally handled by the *Hierarchical Storage Manager* of CASTOR (explicit migration and recalls in the first version), but the physicist can also stage to/from explicit tapes to optimize data throughput in certain cases like Central Data Recording. The major objectives are: high performance, good scalability and easy to clone and deploy to support in the future thousands of clients, excellent resilience and reliability. It is focussed on HEP requirements.

The system is highly modular but not fully compliant with the IEEE Mass Storage Reference Model. It is available on major Unix systems like AIX, HP-UX, Irix, Linux, Solaris and Digital Tru64 and will also be available under Windows/NT.

3 CASTOR major components

The physicist's client application uses two components: the stager to trigger the migration and recall of data and RFIO to access the data on a remote disk pool.

Behind the scene, the major components are the Name Server, the Volume Manager, the Volume and Drive Queue Manager and the Mover called RTCOPY. The stager, the name server and the volume manager use a database and the interface has been designed in a very clean way, so they can either use a commercial database like Oracle and Raima or the CASTOR database system called Cdb. All components are client-server applications written in C and use TCP/IP sockets. They provide a command line as well as a thread-safe C programming interface. Most of the servers use pools of threads to handle the requests. The critical servers are replicated and an automatic fail-over is being implemented.

3.1 stager

It manages disk pools, using Cdb to keep track of the data in the pool. The filesystem selection is done using a round-robin algorithm. The purge policy is very flexible and has been in production in SHIFT for a long time. The migration policy is being implemented. It interfaces to the Name Server, the Volume Manager and RTCOPY.

3.2 RFIO

It implements a remote version of most standard POSIX calls like open, read, write, lseek and close using a very light weight protocol. The control and data streams are separated. To optimize the data throughput by overlapping network and disk I/O, a circular buffer and two threads are used for each connection. Software striping is not implemented, but of course system implementations of RAID or parallel file systems can be used.

3.3 Name Server

It implements an hierarchical view of the name space with files and directories, supports the standard Unix permissions and provides a POSIX API. The metadata are stored in a database as well as in user labels together with the data on tapes for recovery purpose (the tapes are ANSI labelled and self describing). Utilities are provided to create directories, change ownership, list and remove files and directories.

3.4 Volume Manager

It handles pool of tapes either private to an experiment or public. Tapes are allocated for storing files according to file size, media cost, experiment name, filename... File spanning over several volumes is supported: in a given pool of tapes, either the filling of the volumes can be optimized or the number of mounts can be minimized. When the database provides an SQL interface (Oracle, Raima), catalogued procedures and dynamic SQL queries are used for tape allocation.

3.5 Volume and Drive Queue Manager

VDQM maintains a global queue of tape requests, provides tape server load balancing and optimizes the number of tape mounts. Tape servers can be dedicated to certain applications like Central Data Recording and in the future individual drives could also be dedicated.

3.6 RTCOPY

RTCOPY is a complete rewrite of the SHIFT tape mover. The server is multithreaded and uses large ($\geq$ 100MB) circular buffers to overlay tape and network I/O. It also attempts to cache in/out as much data as possible in memory, while it is blocked in tape device operations (mount, position and unmount). Sophisticated error reporting allows the client full insight in case something goes wrong so that the client can chose to retry if it makes sense. RTCOPY is delivered with a complete backward compatibility (with SHIFT) command line interface as well a new thread API, which is primarily used by the stager.

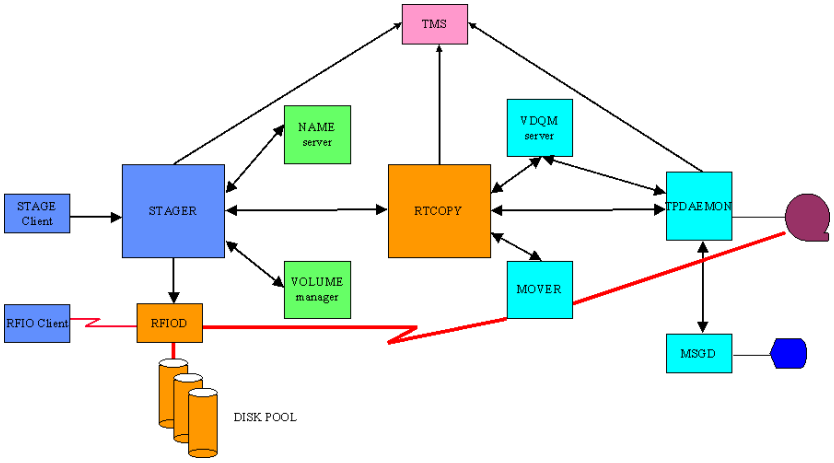

Figure 1: CASTOR layout

3.7 Cdb

The CASTOR Database is a multithreaded, concurrent and flexible database, not yet transactionnal, based on the hash model, which provides the facility to open multiple databases within multiple sessions, each database being defined with a C-like description file by any client having admin privilege. A database is composed of tables, themselved composed of fields which can belong to multiple compound keys. All standard non-SQL access methods are available, including persistent locks in the datafile, partial key search, and move to the next or previous row vs. a given order using partial or full key(s). Cdb provides a thread-safe API.

Tape device	Read from tape (KB/s)			Write to tape (KB/s)		
	1 stream	2 streams	3 streams	1 stream	2 streams	3 streams
STK Redwood	13110	21870	24260	11690	23400	28040
STK 9840	11630	22850	24530	11420	22490	32700
IBM 3590E	13500	21200		17400	31900	

Table I: RTCOPY test: aggregate rates with 1–3 streams per Linux tape server.

4 Early tests

Here follows the first results from the performance tests with RTCOPY and the CASTOR name server.

4.1 RTCOPY

The performance results for the CASTOR tape mover (RTCOPY) are summarised in table I. All numbers are rates with data compression on the tape drives and should be compared with the native speed for each of the tape drives multiplied with a factor 1.2–1.3. The natives speeds are 11MB/s (Redwood), 10MB/s (9840) and 13.5MB/s (3590E). These results have been obtained with one SUN E450 4 CPU disk server and Linux tape servers (each with 3 drives) connected over gigabit ethernet. Tape write performance on Redwood is degraded by about 10 seconds per transfer due to the slow write of tape marks. Tape read performance does not easily exceed 13MB/s per stream because the disk write process was not yet implemented with multiple threads. The *3 streams* rate could not be measured for 3590E because of lack of drives. In a small scale test 120GB was transferred over Gbit ethernet between three disk servers and three tape servers using in total nine tape drives (6 Redwood and 3 9840 drives) during 30 minutes at a sustained rate (including tape mount/unmount) of ~ 70MB/s (~ 90MB/s if tape mount/unmount is not taken into account).

4.2 Name Server

We compare Cdb with Raima[1]/Velocis2.1 database performance for (figure on the left) successive creations (left axis) and listings (right axis) of one directory of 5000 entries and the concurrent listing of one directory of 5000 entries by up to three clients (figure on the right). Our preliminary conclusion is that Cdb and Raima are of the same order of magnitude of performance, the samll differences explained by the database model behind (Hash for Cdb, Btree for Raima). Current Cdb performance scales well for CASTOR needs.

5 ALICE Data Challenge

The ALICE Mock Data Challenge is a feasibility test to run at a sustained rate of 100MB/s during one week (~ 60TB). The hardware configuration is depicted in Figure 2. The data producer processes run on 17 PowerPC nodes in the ALICE test beam area. The data is written directly over a dedicated Gbit ethernet link onto disk servers in the CERN computing center, about 8km from the data producer nodes. The CASTOR stager and HSM manage the disk pools and migrates the files to tape.

[1] `http://www.raima.com/`

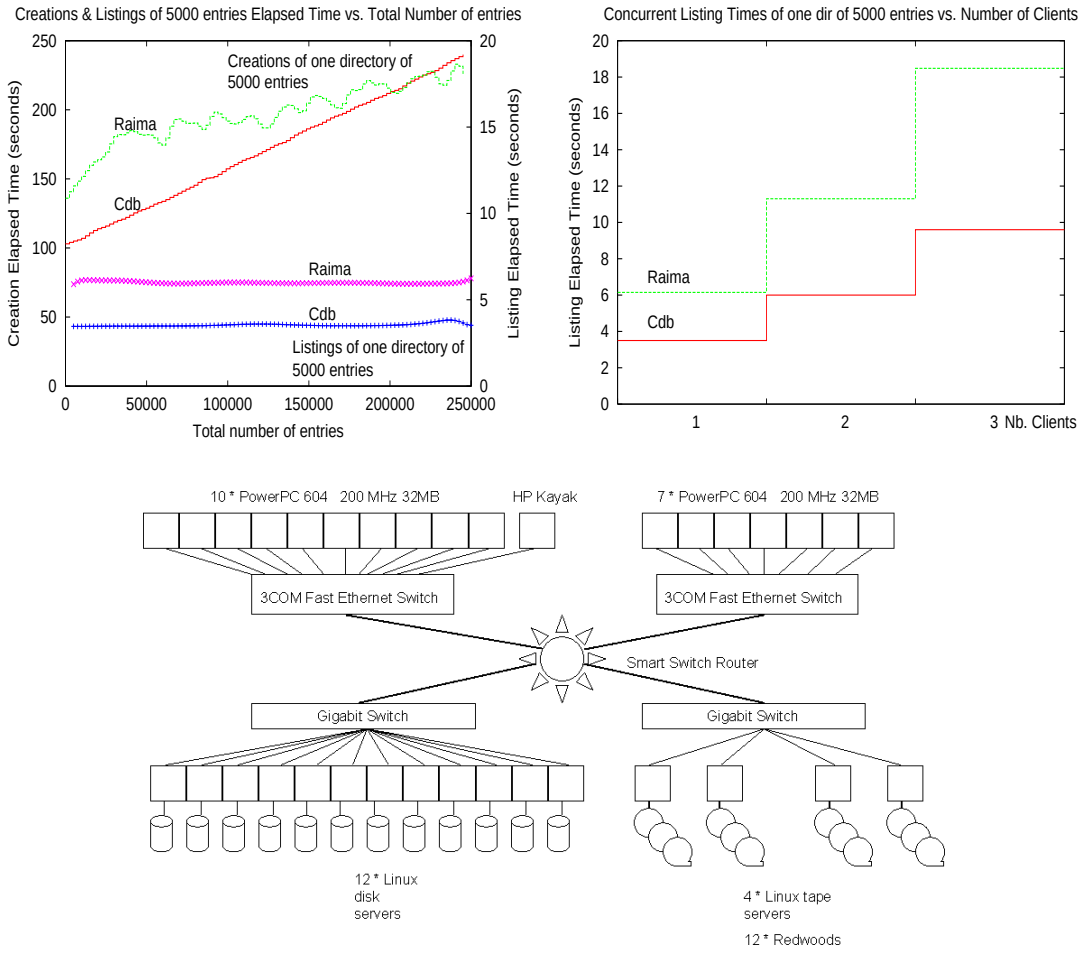

Figure 2: Upper: Name server performance tests. Bottom: ALICE Data Challenge Configuration

6 Current status and possible enhancements

All components described above have been developed and are being extensively tested. As CASTOR provides at least one user interface fully backward compatible with SHIFT and the system is very modular, we will be able to put it in production over the winter 1999/2000 and the goal is to have all experiments using the new software before the accelerator startup. The possible developments are: GUI and WEB management tools, transparent migration/recall, intelligent disk space allocation taking disk activity into account, automatic migration between media types, quotas, undelete and repack functions, import/export of data and metadata from/to other HSM systems...

7 Conclusion

After about two man years of design and development the CASTOR project has now reached its deployment phase almost within schedule. The deployment is facilitated by the high modularity and the full backward compatibility with SHIFT. We see a large improvement over SHIFT and the performance seems to be limited only by the hardware configuration.