An Overview of a Large-Scale Data Migration

Magnus Lübeck
Magnus.Lubeck@cern.ch

Dirk Geppert
Dirk Geppert@cern.ch

Krzysztof Nienartowicz
Krzysztof.Nienartowicz@cern.ch

Marcin Nowak
Marcin.Nowak@cern.ch

Andrea Valassi
Andrea.Valassi@cern.ch

Database group, IT Division,
European Organization for Nuclear Research (CERN)

Abstract

We present an overview of the migration of the data collected by the COMPASS experiment at CERN. In less than three months, almost 300 TB will be migrated at a sustained data rate approaching 100 MB/s, using a distributed system with multiple nodes and data servers.

This project, which is being carried out by the Database Group of CERN's IT Division and is expected to be completed by early 2003, involves both a physical media migration, from StorageTek 9940A to StorageTek 9940B tapes, and a data format conversion, from an Object Database Management System to a hybrid persistency mechanism based on flat files referenced in a Relational Database Management System.

1. Introduction

High Energy Physics (HEP) is an area of science requiring a large amount of data storage. A single HEP experiment may demand storage for hundreds of terabytes of data per year of operation. The data is written out to disks and tapes using different formats and algorithms, ranging from Database Management Systems to home made solution optimized on a case by case basis.

Most of the data accumulated by an HEP experiment, collectively referred to as "event data", records the response of complex electronic devices to an interaction of a high-energy beam of particles against a fixed target, or against another beam traveling in the opposite direction (an "event"). A smaller fraction of the data, which can be referred to as "event metadata", corresponds to measurements of the experimental conditions under which a set of events was recorded (such as the energy of the beam, the type of target used and so on), and/or to a numbering scheme allowing to navigate among the millions of events recorded by the experiment.

While event metadata is usually kept permanently on disks, most of the event data is generally archived on tapes and retrieved relatively infrequently. Given the statistical nature of the interaction between elementary particles, in fact, the set of observations corresponding to one event is completely independent from that which refers to another event. Different events can thus be processed and analyzed individually, making it possible to store most of the event data on tapes and to only retrieve that referring to the set of events that one wants to analyze.

The physics community expects all the data collected by the HEP experiments to remain available for reprocessing and analysis for many years after the end of data taking. HEP is not the only field of science with a need for archival. Studies and monitoring of meteorology, earth science, and genetics are also collecting a vast amount of data, having the same problem of keeping it intact for the future.

Long-term storage and archival is not only the matter of moving data from one medium to another to keep the bytes on the medium in a readable form, there must also be a way of interpreting their logical format to extract the value of the data. Large-scale data migrations must therefore be performed whenever support for the physical medium where the data is stored, or for the algorithms

used to stream it out as a sequence of bytes on the medium, is about to be discontinued.

This paper gives an overview of one such project, the migration of the data collected at CERN by the COMPASS experiment, Common Muon and Proton Apparatus for Structure and Spectroscopy [1,2]. COMPASS is one of the most data intensive HEP experiments in the world, having accumulated almost 300 TB of data since it started data taking in 2001. The typical size of a COMPASS event being 30kB, this corresponds to an order of magnitude of ten billion events. The average sustained input data rate at which all this data has been collected over two years was limited by filtering to about 35 MB/s during each period of operation of the detector in continuous data acquisition mode. Over 80% of the total data volume has been collected during 2002 alone.

The COMPASS event data and metadata has all been stored, mainly on StorageTek 9940A tapes, using an Object Database Management System (ODBMS). Because of a change in the long-term strategy for HEP data persistency at CERN, support for this ODBMS will soon be discontinued. At the same time, 9940A tape drives are about to be phased out as the more recent StorageTek 9940B tape technology is being deployed. To ensure its long-term availability for future reprocessing and analysis, all the data already collected by COMPASS must therefore be migrated to a new system before support for the older tape and DBMS technology is dropped. The new storage system, which will be used by COMPASS to take more data from 2003 onwards, has already been designed and involves a Relational Database Management System (RDBMS) for event metadata, and flat files stored on 9940B tapes, in the "DATE" format [4] used by the experiment data acquisition system, for event data.

This paper briefly describes the software and hardware components most relevant to the COMPASS migration, as well as the data volumes and timescales concerned. The project is under the responsibility of the Database Group of CERN's IT Division (IT-DB).

2. CERN's mass storage system

Storing, retrieving and writing data to tape at CERN are performed using a hierarchical storage system (HSM) called CASTOR [5,6], CERN Advanced Storage manager. CASTOR provides an API and a set of shell commands for staging and archiving files, designed to send large amounts of data to long-term (tertiary) storage via a disk cache. The system focuses on the data transfer needs of HEP experiments, i.e. sustained high transfer rates without any possibility of retransferring data.

CASTOR can be thought of as a file-system-like storage for archiving, removing the need for the application to know about the specific storage technology behind it. Information about files stored in CASTOR is retrieved from dedicated name servers. The CASTOR API allows to transparently access files using familiar system calls like open() or close(). The storage hardware is connected to network attached CASTOR tape servers. The tapes are stored in large tape libraries called "Silos", where each Silo can store up to 6000 tapes.

3. Data volumes and migration timescales

The most challenging aspects of this project are the data volume concerned and the short time frame available to complete its migration. The project involves reading 300 TB of event data from tape, converting it into a different format, and writing it on tape again. The tape drives are the most precious resource, as their availability has to be planned months in advance and in competition with other projects. COMPASS is only collecting data during about 100 – 150 days per year and the next data taking will start in the spring of 2003. Given that the migration should be completed before that date, and taking into account the availability of tape drives and other hardware resources, it is expected that the migration should be performed in a period of less than three months in total, between the end of 2002 and early 2003.

Table 1. Number and size of files in ODBMS

Period	Raw files	Raw size [TB]	Raw tapes
2001 P2B	21215	16.76	358
2002 P1A	39189	26.38	295
2002 P1B	28561	26.05	416
2002 P1C	15705	14.69	227
2002 P2A	32537	31.79	616
2002 P2B	21415	21.35	596
2002 P2C	23046	23.78	627
2002 P2D	22756	22.55	633
2002 P2E	26511	27.05	900
2002 P2F	11317	11.31	743
2002 P2G	24258	24.45	556
2002 P2H	17354	17.20	588
Total:	283864	263.35	3449

Table 1 lists the number of files and the size of the data stored by the COMPASS experiment in each of its many independent "periods" of data taking, as of before the start of the migration. As stated in the table, the data is spread over more than 3400 distinct tapes. The numbers of tapes used for the various periods actually add up to a larger

number, as files from different periods are sometimes stored on the same tape.

Before starting the actual migration of the data, a test was made on a single node, using the CASTOR system to estimate the number of drives and conversion nodes needed to complete the migration in less than three months. The tests showed that the conversion software was able to put up with a sustained data processing rate of 10 MB per second, while the transfer speed from 9940A tapes to disk and from disk to 9940B tapes were estimated in approximately 11 and 17 MB/s. These figures are lower than the specifications from the tape vendor because of the hardware and software setup of the test, where a single machine (a 1GB RAM, 1GHz, 2CPU Linux system running RedHat 7.1) was running the input staging, the migration conversion software and the output staging at the same time. The test was not intended to measure the optimal performance of each and every part of hardware in the system, but rather to provide input for the initial planning of the resources required by the migration project in a worst-case scenario. As a consequence, our estimates of the data rates from tape to disk (and vice-versa) should not be seen as a reference for other projects.

Assuming that the performance of the system scales linearly and aiming at completing the migration in less than 50 days to keep a large safety factor with respect to the three months allocated, it was estimated that nine STK 9940A input tape drives, five STK 9940B output tape drives and ten conversion nodes should be reserved for the migration period. These figures are summarized in table 2.

Table 2. Estimate of the required resources

Max total migration time	**50 days**
Total data volume	300 TB
Needed migration rate	70 MB/s
Needed # input drives	9 (at 90% of 11MB/s)
Needed # output drives	5 (at 90% of 17MB/s)
Needed # conversion nodes	10 (at 90% of 10 MB/s)

Our estimate of the resources required by the migration project takes into account that, in general, there are very few cases where throughput in a complex system is optimal. Without attempting to estimate it on more solid grounds, the overall inefficiency of the migration chain was arbitrarily assumed to be of the order of 10%. Assuming a 90% efficiency for the availability of all hardware resources allocated, the total sustained input rate on the 9 input tape drives was thus estimated in about 90 MB per second, as listed in table 3. This is of the same order of magnitude of the total conversion rate on the conversion nodes.

Table 3. Input tape drive specifications

Total # input drives (STK 9940A)	**9**
Input rate per drive	11 MB/s
Input efficiency	90%
Total input rate	90 MB/s
Input tape size	60 GB
Typical file size	1.4 GB
Time to read in a full tape	1.7 hours

The output tape drives, whose characteristics are listed in table 4, are a more recent model from the same vendor of the input tape drives, with higher data rates and larger capacity per tape. As their availability is limited, only 5 such drives have been allocated. These were estimated to generate a total output rate of 77 MB/s, enough to complete the migration in the planned time frame. To make sure that the output drives are never idle waiting for input data to be read in and processed, the total input and conversion rates are instead slightly higher.

Table 4. Output tape drive specifications

Total # output drives (STK 9940B)	**5**
Output rate per drive	17 MB/s
Output efficiency	90%
Total output rate	77 MB/s
Output tape size	200 GB
Time to write out a full tape	3.3 hours

While event data must be transferred to new tapes in a different format, also event metadata must be migrated, into a relational database. The typical size of 30 kB per event shows the need for a sustained migration rate of about 2500 events per second. The output database system should therefore be capable of a minimum insertion rate of 2500 rows of data per second. A single relational database server, running on commodity hardware, is sufficient to handle this data rate.

While the volume of event metadata represents only a few percent of that of event data, the total volume of event metadata to store in a RDBMS sums up to a few TB. In the present initial phase, only two database servers are used, mainly to allow for cold database backups without interrupting the migration. Taking into account the future access patterns to the data from COMPASS users, we foresee to distribute the data already collected by COMPASS between five servers, partitioned according to COMPASS “periods”. Another three to five servers will be added to store metadata for the new data collected by the experiment in the upcoming 2003 data taking.

4. Software framework for the migration

The preparation for the COMPASS migration occupied up to five people full time over the last 6 months of 2002. The most complicated issues during the development phase were the investigation of the COMPASS data structures, the development and validation of the data conversion application, the setup of a scalable framework to distribute jobs to many conversion nodes using a large number of input and output tape drives, and the bookkeeping of the completed workload.

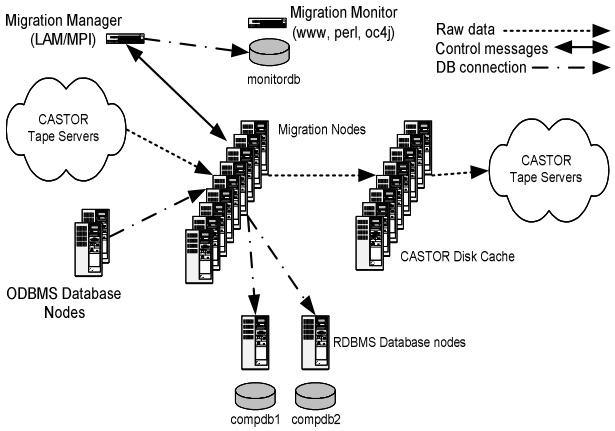

Figure 1. Compass migration framework

The topology of the software framework developed to perform the migration and conversion of the data is shown schematically in Figure 1. Many instances of the conversion application, running in parallel on the migration nodes, are monitored and controlled by a single manager application running on a separate management node. The manager and the conversion applications, both implemented in C++, communicate using the LAM [7] open source implementation of the Message Passing Interface (MPI) standard [8]. The system is designed using a push-architecture, where the master distributes the workload to its slaves, waiting for them to signal the successful completion of their task or any errors encountered. The bookkeeping of the completed workload, stored in a dedicated database on a separate monitoring node, is kept up to date by the manager application and can be visualized in real time through a Web server running on the monitoring node.

Migrating event data simply consists in reading it from one medium, making a number of checks and conversions, and writing it to another medium. However, when migrating thousands of files on tape, which is a sequential medium, the order in which the system accesses the files has a great impact on the speed of the migration. Before the migration chain was started, all COMPASS files were thus grouped according to the thousands of different tapes they reside on, to make sure that all files on the same tape are read in together at the same time. This reduces the otherwise enormous overhead caused by mounting and winding tapes to each requested file individually. Since the data sets collected during the various periods may differ both in their physics content and in the format used for their persistent representation, the migration actually proceeds period by period. The total number of input tape mounts will therefore be higher than the total number of distinct input tapes available. This overhead is however negligible on the time scale for the full migration.

On each conversion node, all files contained on a same tape can thus be migrated sequentially. A migration application, spawned on the conversion node by the monitoring node, processes one input tape at a time, sending to CASTOR the list of files in that tape that should be staged to local files. CASTOR itself determines the optimal order in which files should be read in from the given tape. File sizes range from some hundreds of megabyte to 2 gigabytes, with an average around 1.4 GB. A variable number of files are staged in at each bulk load, whose typical size is about 60 GB. When all files on the given tape have been staged in successfully, the monitoring node starts the actual conversion on the migration node.

For a given tape, the processing cycle described above is represented in the migration application as a finite state machine, implemented in C++ using the SMC open source State Machine Compiler [9]. In the absence of errors, a single migration application rotates through four successive states, Idle (I), Staging (S), Loaded (L), Migrating (M), then back to Idle. All I→S and L→M transition are only performed on the manager's request, while the S→L and M→I transitions simply indicate the successful completion of staging in and migrating a data file. The I→S transition request is accompanied by the name of a new tape to stage in and process.

The migration framework is designed to keep each input tape drive constantly busy staging in data to disk. To make optimal use of the dual-CPU migration nodes, two instances of the conversion application run in parallel on each node. The coordination of the two applications on each node is ensured by the manager, which models their status according to the finite state machine shown in Figure 2. Most of the time, nodes are in the SM or MS state, where one application stages in data on two of the five available local disks, while the latter migrates the data previously staged in on two different disks. The MM state, with both applications migrating data at the same time, is forbidden to avoid competition for CPU or for network access to the remote output CASTOR pool. The SS state,

where both applications stage in data at the same time, is forbidden to reduce the queuing time for input tape drives on all other nodes.

The conversion applications are connected both to the input object database and to the output relational database. The input files, where event data and metadata are both stored in ODBMS format, are "attached" (i.e., brought online) to the object database as soon as they have been staged in to the local disks of the given conversion node. The migration programs read the data from one file at a time, convert it to a different format, and write it to its output destinations.

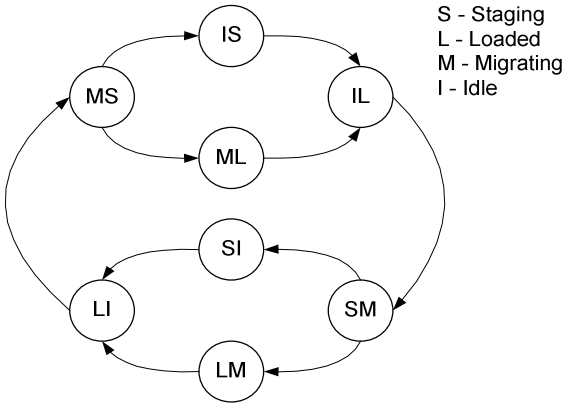

Figure 2. Two-process Finite State Machine

Event data is stored in flat files in CASTOR using the DATE format, while event metadata is written out both to the RDBMS, in a format specific to the database. The event metadata is also written locally to disk on the migration node, in an intermediate file format that may be used for recovery purposes. While all files staged in on a conversion node are being converted, the output DATE files are staged out to tape using CASTOR. Staging out is not as complicated as staging in, as the access pattern of the data does not require any reorganization of files on the output tapes. Staging out is done by the migration program, using the CASTOR API, writing directly over the network to the CASTOR disk cache in a similar fashion as writing directly to files on local disk.

5. Interim status report and updated plans

The present status of the ongoing work at the time of writing (on January 17, 2003) is shortly presented in this section. The migration started off on November 29, 2002, using an initial setup with less than half the required hardware resources. In particular, only one input drive and no dedicated output drives were available at that time. A large portion, but not yet 100%, of the required resources has in the meantime been received and put in production. Additional tape drives and migration nodes will be available in the coming weeks. While tape drives can be added dynamically at runtime, the addition of more migration nodes will require a restart of the system.

After around six weeks of running, the volume of COMPASS data migrated is more than 93 TB, i.e. 35% of the total. As seen in Table 3, the corresponding volume of migrated raw data and metadata amounts to approximately 74 TB, showing that the data format conversion from ODBMS to DATE saves over 20% of storage space. All data volumes and rates quoted in the following tables and plots refer to the newer, more compact, data format.

Table 3. Migrated data

Period	Raw data [GB]	Metadata [GB]	# Events [x1000]
2002 P1C	10972.56	19.25	332.23
2002 P2A	25572.87	40.69	764.87
2002 P2C	19023.60	28.30	528.31
2002 P2D	18689.93	24.36	455.49
Total:	74258.96	112.60	2080.90

Figure 3, produced using JAS [10], shows the time evolution of the migration data rates from the start of the migration until this moment. The same software tool is used to monitor the migration during runtime. Although the data rates were quite low in the first two weeks of the migration (when we had to correct a few unforeseen errors encountered), peak data rates above 4.5 TB/day were achieved in the following weeks, corresponding to a sustained throughput of almost 60 MB/s. After a stop of a few days around New Year's Day, when nobody from our group was available to restart the system after an error, the migration is now proceeding relatively well.

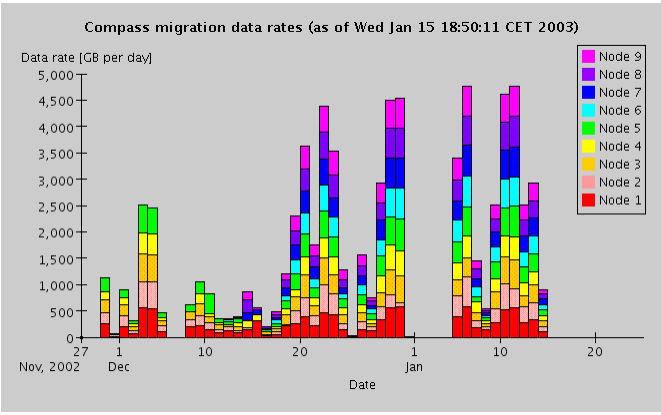

Figure 3. Data throughput [GB per day]

Figure 4, in particular, zooms on the time evolution of the migration data rates during the last three days. It shows that a sustained throughput around 60 MB/s is

being achieved when the system is up and running. Taking into account the 20% difference in data volume between the old and new format, which had not been considered in our initial analysis, this corresponds to 75 MB/s, higher than the minimum rate of 70 MB/s estimated in Table 2 to migrate 300 TB in 50 days.

The load distribution between different migration nodes is fairly even. The two stops in the figure correspond to external problems - the unavailability of output tapes and a technical stop of the CERN Computer Center. The latter was foreseen and impossible to avoid. CASTOR was partially affected by the stop and suffered initially when turned back on. Lack of tapes is instead a logistic problem, which could have been avoided.

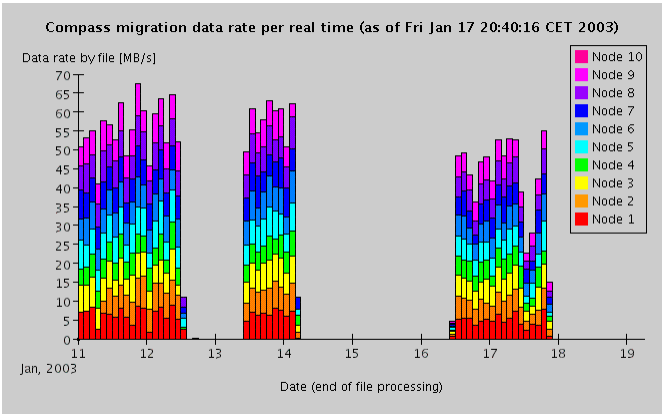

Figure 4. Data throughput [MB per second]

Figure 5, although very similar to Figure 3, shows the time evolution of the migration data rates measured using a different unit, in number of migrated events per second. Evens are migrated at a peak rate of 1500/s, comfortably lower than the worst-case scenario of 2500 database row inserts per second considered in our initial analysis. That estimation had been intentionally conservative, using an event size of 30kB, lower than we now observe (40kB in the new format, or 50kB in the old format).

The use of a finite state machine to model the application workflow, shown in Figure 2, together with the bookkeeping of the state transitions in the migration monitor database, provide very useful tools to analyze the performance of the migration application itself. When the system is up and running, the migration nodes spend only 50% of the time in states MS and SM, and most of the remaining 50% in states IS or SI. Since it takes approximately the same time to stage in all files on a tape or to migrate them, this is a clear indication that the migration processes queue up for stage-in due to the lack of dedicated input tape drives.

In summary, although it is now clear that the migration will take longer than 50 days to be completed, the observed peak performance of the system is up to the expectations formulated before the start of the migration. The early start in November, although with limited resources, allowed us to test and improve the system we developed and to have almost a third of the data migrated by mid-January. For the rest of the migration, we expect the performance of the system to further improve, thanks to the availability of dedicated input and output tape drives. The limited number of dedicated tape drives was, in fact, the bottleneck during the initial phase of the project. We are thus confident that the project will be completed in the time allocated, by the end of March.

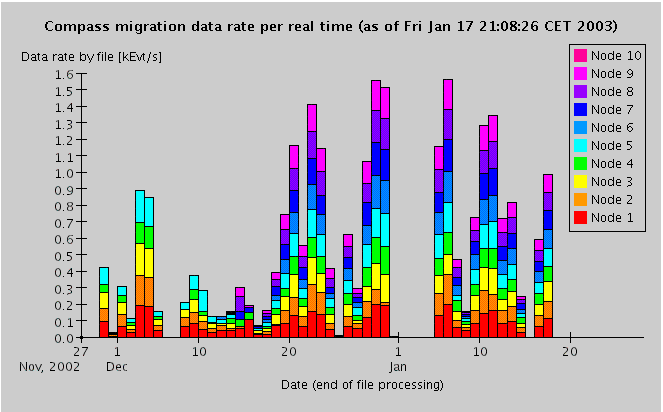

Figure 5. Migrated events [kEvts per second]

While this paper focuses on the migration of the data collected by the COMPASS experiment, the concepts and software framework developed in this context can be applied to other large-scale data migrations. As soon as the COMPASS migration is completed, indeed, the software framework developed for COMPASS will be adapted and reused for the migration of the 30 TB of data collected by another large HEP experiment, HARP [11].

6. Conclusion

Because of the rapid evolution of hardware and software technologies for data storage, the need to perform large-scale data migrations is common to all projects that critically depend on the long-term availability of their data. High-energy physics experiments, with their huge amounts of data and high manpower and material costs, are only one such example. In this paper, the migration of the 300 TB of data collected by the COMPASS experiment at CERN has been presented. The hardware and software components most relevant to this project have been described in detail, showing how a scalable and parallel framework for the migration is being built using CERN's CASTOR tape manager.

Acknowledgements

It is a pleasure to thank our colleagues from the DB, DS and ADC groups of the IT Division at CERN for contributing to the design and setup of the COMPASS migration infrastructure. We would also like to thank the COMPASS computing group for their continuous help and feedback. We finally wish to thank Tony Johnson for his kind help in using JAS.

References

[1] G. Baum et al. (COMPASS Collaboration), "COMPASS: Proposal for a Common Muon and Proton Apparatus for Structure and Spectroscopy", CERN-SPSLC-96-14, March 1996.

[2] L. Schmitt, "The COMPASS Experiment", ICHEP98, Vancouver, July 1998.

[3] A. Martin, "The Compass off-line computing system", CHEP2000, Padova, February 2000.

[4] CERN ALICE DAQ group, "ALICE DATE User's Guide", ALICE Internal Note/DAQ ALICE-INT-2000-31 v.2, January 2001.

[5] J. P. Baud et al., "CASTOR architecture", July 2002, available at http://castor.web.cern.ch.

[6] J.P. Baud et al., "CASTOR Project status", CHEP2000, Padova, February 2000.

[7] LAM/MPI (Local Area Multicomputer) Parallel Computing, http://www.lam-mpi.org

[8] The Message Passing Interface (MPI) Forum, http://www.mpi-forum.org.

[9] Charles W. Rapp, The State Machine Compiler (SMC), http://smc.sourceforge.net.

[10] T. Johnson et al., Java Analysis Studio (JAS), http://www-sldnt.slac.stanford.edu/jas.

[11] M. G. Catanesi et al., "Proposal to study hadron production for the neutrino factory and for the atmospheric neutrino flux", CERN-SPSC/99-35, November 1999.