The COMPASS Computing Farm Project

M. Lamanna

CERN, Geneva (CH) and INFN, Trieste (I)

Abstract

The COMPASS experiment at CERN is building a large facility for the offline computing in close collaboration with the CERN IT division: the COMPASS Computing Farm (CCF). The motivations, the experience, and the plans for deploying a 2,000 Spec INT95 computing farm are discussed. The main technical points are the use of Intel-based Linux PCs for both the I/O and the CPU intensive tasks, and the use of an object database to store all the data.

Keywords: PC Farm, Linux, Objectivity/DB

1 Introduction

The COMPASS experiment will start commissioning the detector and taking data in the year 2000. COMPASS is a fixed-target experiment with a diverse physics programme at the CERN SPS. The apparatus will perform a number of different measurements, in many different configurations, notably using both muon and hadron beams in the 100-300 GeV range at very high intensities[1].

The high data rate (over a few months per year) to be processed and the need for a flexible software environment to cope with the experiment's different measurements have pushed the COMPASS collaboration to design the offline analysis software from scratch, using new techniques and solutions. The foundation of the offline environment is given by a software framework for the data analysis and the deployment of a very large computing architecture with large input-output and storage capabilities. To provide the right solution to the computing problem, COMPASS, together with the CERN IT/PDP group, proposed the COMPASS Computing Farm project.

The CCF's main tasks are:

- Central Data Recording (CDR; 35 MB/s many months a year).
- Reconstruction of all data (10G events, 30 kB each, 300 TB/y).
- Analysis and filtering (samples below 1 TB will be exported).

COMPASS decided to use the CDR service to record all data: the online system does not write events on tape at the experiment site, but sends them over a few km of dedicated network to the computer centre, where the CCF, tape servers, and the corresponding high-speed tape drives are located. Transfer rates of the same order of magnitude as COMPASS anticipates have already been routinely sustained for the NA48 experiment.

The estimated computing power is 2,000 Spec INT95, which will be provided by some 100 PCs. The choice of network technology is Gigabit and Fast Ethernet. A disk pool of a few TB is being setup: presently most of the disk space is SCSI but EIDE is also under consideration.

Another major requirement is the use of a database layer integrated with a storage manager. The solutions for the data storage are still under evaluation, together with the development of the C++ reconstruction framework.

2 The underlying data and analysis models

COMPASS decided to adopt a computing model such that all the data are reconstructed in a single reconstruction facility in parallel with the data taking.

COMPASS decided to write the bulk of the reconstruction programs in C++. The reconstruction and analysis program CORAL (COmpass Reconstruction and AnaLysis; `http://coral.cern.ch`) has been designed using object-oriented techniques. It uses an object database called Objectivity/DB which has already been investigated at CERN for present and future experiments (RD45, NA45, LHC experiments).

The large quantity of data (300 TB/year) means that all the data may not be disk resident at the same time. The limitation in total disk size can be made transparent with the use of a Hierarchical Storage Manager (HSM). The SHIFT software is being upgraded in the IT/PDP group into a full HSM system called CASTOR, which is that first HSM candidate for COMPASS.

The data flow model should foresee different scenarios; the most relevant ones are the DAQ mode and the quasi-online reconstruction mode.

In the DAQ mode, events are written on the online farm event builder disks in raw format; a run comprises $\approx$ 10 files, reflecting the number of parallel event builders. The run files are transfered to the CCF and the events populate *in parallel* the federated database ("stage1"). The original files are deleted after "stage1", the corresponding databases are moved to the HSM, the federated database is updated accordingly and finally the databases are deleted from the CCF disks.

In the quasi-online reconstruction mode, a data base is deleted from the disk only after the extra condition of having been read from a CPU client for track reconstruction has been satisfied. The events are read as objects via the Objectivity/DB object server (AMS).

The analysis model is based on a run unit, a time interval in which one tentatively assumes that the data-taking conditions are stable. This coarse granularity allows the problem of robustness and error recovery to be assessed in the simplest way. The treatment of a data set (a file or a database file) is done as an atomic action, either successful or failed without relevant side effects (therefore it can be completed in principle by a retry). The actions which are performed *in parallel* are designed not to require process intercommunication and not to depend on the detailed time schedule of the processing of the different subparts.

2.1 CCF model

The model proposed for this computing facility is a farm using as much as possible "commodity solutions", mainly PCs and mass-market network technologies.

At the beginning of 1998 we started with the idea of deploying a number of mid-range RISC servers for I/O plus a cluster of cheap PCs. In this document, *data servers* refer to the I/O intensive core of the farm, and *CPU clients* the number-crunching PCs (Fig.1). Other more conventional solutions were discarded because of the cost. From the beginning, the farm was split into two areas with specific responsibilities and optimised hardware and software, namely Unix RISC servers for the I/O intensive part and Windows NT on dual Pentium boxes for the CPU intensive activities.

This model is also well suited for the collaborating institutes, in that it is not an expensive monolithic mainframe, and may incorporate existing hardware. COMPASS institutes from Germany and Italy have shown interest and some prototypes already exist. These analysis farms will be fed with tape-exported samples, to perform the final analysis of specific reactions (a total export in not possible due to the full data sample size). On the other hand, we plan to access some of the data not present in these remote farms (RAW data samples, calibrations) via wide-area network.

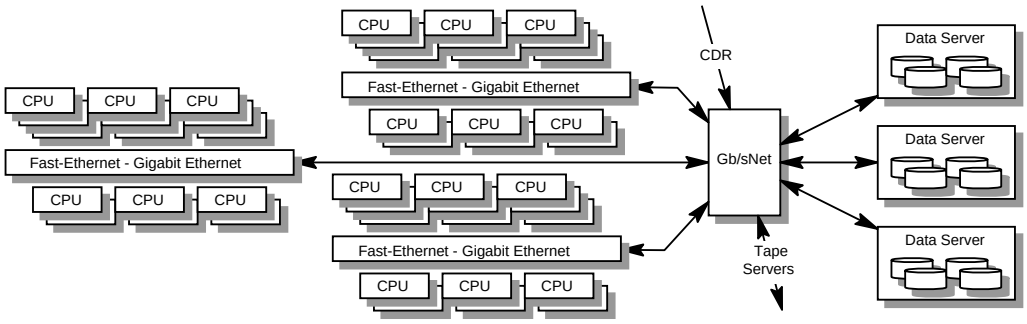

Figure 1: A schematic of the CCF. The CPU clients are on the left, on the right the data servers (holding the farm disk pool). The lines represent the main network connections and data flows: the CDR line represents the data flux from the experiment to the farm; the Tape Server line represents the data exchange between the farm and the tape infrastructure.

3 The components of the CCF

The decision to shape the CCF in a modular environment was taken very early, on the basis of both technical and economical considerations. The deployment of "independent" units in all CCF areas (instead of single high-end devices) best addresses the problem of the performance to price ratio for each area, gives the possibility to increase the configuration in small steps, to share hardware and resources within similar projects[1], and, most importantly , to remove single points of failure.

The concept of modularity overflows quite naturally into the idea of deploying independent components, which means that one accepts heterogeneity on the first place even within the same area. As matter of fact we have run having heterogeneous data servers (DEC and Sun) or different operating systems for the CPU clients (Linux and Windows NT) *at the same time*.

The idea to split the problem of the farm design and operation into well-defined components can be obviously quite popular, but very often the drawbacks are overlooked. The decomposition only make sense if each component has a well-defined interface *without hidden dependencies*: this is presently the target we are aiming for, but in general it cannot be fully realised. Any component should be considered only if its choice does not affect the freedom in choosing other components (with the net result of no flexibility at all and overall more modest performances).

The outcome is that all solutions should either be tested against others (e.g. Windows NT vs Linux) or at least allow for possible future comparisons (i.e. use generic solutions or exploit only generic functionalities from a software component). Obviously, the tests should be performed if possible in a realistic environment to assess compatibility and scalability issues.

3.1 The hardware components

The CPU clients are PCs. For the data server part, we tested DEC 1200 (1 CPU; Gigabit Ethernet interface with/without a second HiPPI interface), SUN 450 (2 CPUs; Gigabit Ethernet interface with/without a second HiPPI interface), and PCs (2 Pentium II CPUs; either with Gigabit or Fast Ethernet interface). The PC (with Fast Ethernet interface) is also the present solution for data servers. The SUN proved to be a reliable solution and it is considered as a fall-back (I remind the coexistence of different data servers has been already tested). The performance of the DEC 1200 was not adequate in our environment (high sustained I/O rate with many concurrent streams).

[1]Presently (January 2000) at CERN many dedicated facilities are being deployed, using the similar hardware and the same system administration scheme, which add up to more than 400 PCs.

3.2 The software components

The control software has been written from scratch, using the experience of the original CERN CDR. The Perl language has been selected: among all advantages, it can be used with Windows NT clients almost transparently and is accepted by LSF (the batch system) as a scripting language. All other scripting languages have been removed. Most of the network activities are delegated to the CERN SHIFT software (RFIO) and to Objectivity/DB (AMS). The distributed batch queueing system is LSF (Load Sharing Facility). It has been selected because of availability and existing expertise; the main drawbacks are its cost and complexity.

For monitoring and performance analysis, the data are either broadcast via UDP or written into local log files. The monitoring will play a central role in the commissioning of the final prototype, providing immediate feedback about problems. For example, good prototypes of monitor programs for the Objectivity/DB AMS (CPU consumption for each thread) and Lock Server (CPU consumption, transaction statistics) exist. Specialised scripts to test the installation of the CCF software and of the underlying hardware and software environment also exist. A set of Perl scripts feeds tables to PAW or ROOT to analyse the results using histograms, graphs, and results tables. Perl/Tk is used to provide a simple GUI interface.

4 The tests

An online farm prototype and the corresponding software to generate mock data with the appropriate characteristics (data rate, events size) have been developed to test the CCF. This farm has been set-up at the experiment site and uses the full network infrastructure which will be used in the experiment. The control software to control network transfer and tape transfer was tested during summer 1999 to do the CDR for the COMPASS test beam data.

4.1 CPU client test (Windows NT vs Linux)

Before having a Linux version of Objectivity/DB, only Windows NT could be used in the CCF. With the first version for Linux, a clean comparison using both operating systems was possible. This was done writing objects from a PC (memory to network) to a fast RISC server equipped with Gigabit Ethernet. Single and multiple stream behaviour has been studied. The comparison shows typically a 30% boost in database access when Linux is the operating system. After these tests the development on Windows NT was stopped and all PCs moved to Linux.

4.2 PC data server test

The use of Linux PCs as data servers is very appealing. Even disregarding the exceptional price performance ratio, it is clear that having the same basic unit (a PC) for both data servers and CPU clients gives maximum flexibility (for example to tune the ratio of CPU clients to the data servers).

The first test and deployment of data servers at CERN (PCs with SCSI disk pool) has been pioneered for NA48 and for the CERN Computer Centre tape servers. The CCF was the first benchmark of the use of Objectivity/DB (AMS; versions up to 5.2). The behaviour is comparable with the measurements on DEC and Solaris machines. A PC with Fast Ethernet can write to its local disks with its AMS objects at a speed close to the network limit.

4.3 The DAQ Mode Tests

The CCF has been tested in the DAQ mode, up to a set-up with 11 data servers, at increasingly input rate. The main results are that the behaviour scales with the number of data servers and that the 11 data servers can sustain about 35 MB/s for many hours (“stage1” run on data servers: fig.2).

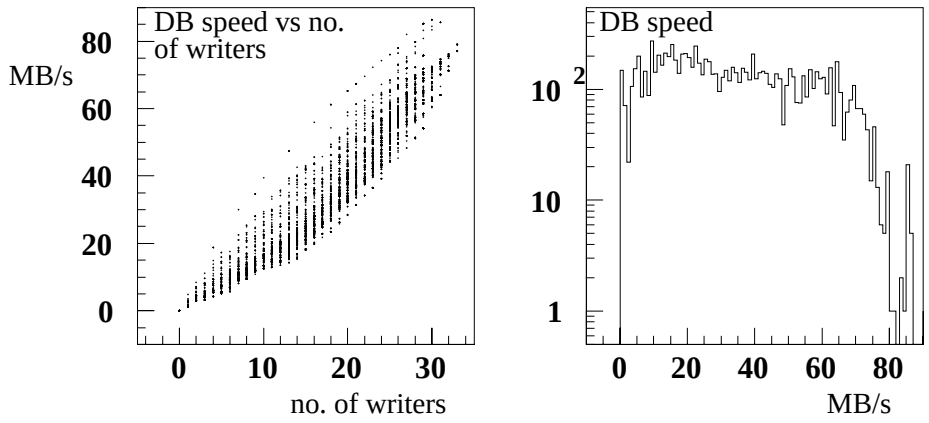

Figure 2: The "stage1" subsystem in the 35 MB DAQ test (Events conversion into objects in an Objectivity/DB federated data base). The rate of data in MB/s is shown as a function of the number of active parallel writers and as an histogram.

The sustained 35 MB/s figure means that at the same time the following activities coexist, all of them at the 35 MB/s: the raw data are input in the CCF and written to the data servers disk; the same flux is input into the databases; databases are read from disk and copied to a remote system to mimic the HSM transfer. All quantities are measured over a test period of ≈ 8 hours.

4.4 The Quasi-Online Reconstruction Mode Tests

The Quasi-Online mode is presently under test; configurations with 5 data servers and about 10 MB/s are run to test the behaviour of the multi-threaded version of the AMS (v.5.2). The system is stable but the AMS behaviour is not yet satisfactory because of persisting time-out problems, which could be recovered at the expense of resubmitting of some reconstruction jobs.

5 Conclusions

The first full CCF prototype will be commissioned in May 2000. The project is evolving towards a Linux PC farm. Data servers will be most probably Linux PCs.

A long period of testing gave us the possibility to select the appropriate hardware for every option, to shape the architecture and the control software in a modular way, and to test many prototypes in many different realistic conditions.

The experience with the Objectivity/DB package is rather positive, but open problems remain and need to be fixed. The usage of Objectivity/DB within the CORAL framework is well-understood and sufficiently isolated to consider it a separate interchangeable component.

In the commissioning of the CCF, the most critical part will be the ability to study in detail and in the real environment the behaviour of the critical components of the CCF (like the AMS), which explains our investment in monitoring tools.

I would like to acknowledge the support and the help of the IT/PDP group, the COMPASS Collaboration, and the COMPASS Offline Group.

References

1 COMPASS, COMPASS Proposal, CERN/SPSLC/96-14, SPSLC/P297, March 1, 1996; COMPASS, Addendum 1, CERN/SPSLC/96-30, SPSLC/P297 Add. 1, May 20, 1996.