BASTA, the IN2P3 heterogeneous simulation cluster

Y. Fouilhé, J. Furet, M. Gaillard, N. Giraud, P. Larrieu, J. O'Neall, A. C. Pardo, W. Wojcik
Centre de Calcul de l'IN2P3, F-69622, Villeurbanne, France

G. Grosdidier
Laboratoire de l'Accélérateur Linéaire, IN2P3-CNRS, F-91405, Orsay, France

To cope with the increasing computing power required for HEP experiments, the IN2P3 has set up a heterogeneous workstation cluster connected to our IBM mainframe running VM/CMS. Only "CPU–bound" jobs are run on the cluster. Controlling NQS, the locally developed Basta Queueing System (BQS) keeps the CPU close to 95% busy. Data files produced during jobs are transferred to the VM/CMS mainframe, which acts as file server for the entire IN2P3 user community. The user environment includes CERN libraries plus tools to hide the differences between the versions of UNIX. The cluster produces slightly more CPU than a 3090/60S mainframe, with good reliability.

Introduction

The IN2P3 Computing Center (CCIN2P3) in Villeurbanne offers a centralized computing facility to French nuclear and high–energy physicists. This service previously was based on an IBM 3090/60S mainframe running VM/CMS with HEPVM software. At the beginning of 1991, user requirements were foreseen to exceed the real capacity of the mainframe. An important part of this demand was for "CPU–bound" HEP simulation jobs, typically running for ten hours and writing a cartridge (3480) of almost 200 megabytes of data.

The idea of using "cheap" workstations to absorb this load was retained after discussions between IN2P3 physicists and Computing Center engineers. Parallel development of different aspects of the project was then carried out cooperatively by these same users and engineers. The main points which drove the BASTA (BAtch STAtions) working team were heterogeneity of the cluster, environment transparency, use, wherever possible, of existing or standard tools and overall system reliability. Another important consideration, born from our HEPVM experience, was to compare projects with the SHIFT group at CERN, so as to maintain as much compatibility as possible for those of our users who might also use SHIFT or other such facilities at CERN.

We will describe our choices of hardware, the batch environment, the file system organization and those user tools needed to work properly on this new facility. Afterwards we will review user experience and the current status of BASTA, before concluding with perspectives for the future.

Choice of hardware

From the inception of the BASTA project, it was considered essential to maintain independence with respect to any given hardware manufacturer. Respect of this criterion necessitated the use of UNIX operating systems, available on all stations. After making benchmarks (in the spring of '91) on different platforms using the DELPHI detector simulation, a choice was made of two architectures: the IBM RS6000/550 with AIX and the Hewlett Packard HP9000/730 with HP-UX. Also, about this time, CERN announced that the CERN libraries would be maintained on those workstations.

These two systems, each running a different flavor of the UNIX operating system, were at that time the most powerful workstations available on the market, each being approximately equivalent in power to 1 CPU of the 3090/60S, which we have chosen as the "IN2P3 unit" (or "UI", 1 UI = 8 CU, CERN units).

We chose to interconnect the processors by the simplest method, a "thin" Ethernet connected to the mainframe via a standard IBM 3172 controller unit.

The Basta batch system

The standard distributed batch product under UNIX is NQS (Network Queueing System). It manages queues distributed among physically different machines with a reliable internal protocol. Unfortunately, NQS lacks a proper load balancing mechanism, certain control options on submitted jobs and a user–friendly query procedure. The decision was taken to provide future BASTA users with a "UNIX batch à la SLAC" interface constructed on top of NQS. This layer distributes batch jobs efficiently among processors and allows full control by users of their queued and running jobs. This layer, called BQS, for BASTA Queueing System, is implemented as a set of small scripts, C programs and daemons. Care was taken to follow POSIX recommendations of the time. The following are the principal BQS user commands:

- *qsub* submits a batch job to the cluster (replaces the NQS command of the same name)
- *qjob* queries job status
- *qhold*/*qrls* holds/releases a job
- *qdel* deletes a queued or running job
- *qalter* modifies job characteristics

BQS, like SLAC batch, defines different classes of jobs in order to distinguish between test and production.

We employ the CERN version of NQS but the only CERN mods we use are the *prologue* and *epilogue*, used to establish and purge the *user batch directory*.

Operator commands are also available to configure, start or stop the different entities of BQS (and NQS). Since time–limit enforcement did not work on AIX 3 and does not exist on HP-UX, BQS implements its own time–limit checking using signals. Operator commands also exist to control the dispatching algorithm by modifying job and group priorities (target values for monthly usage).

External file service

The other key element in the BASTA batch system is file communication with the VM mainframe, which manages tape cartridges stored in our StorageTek robot with four silos (about 22,000 cartridges). For this reason, the VM machine is a file server for BASTA.

Initially, only binary data files of up to 200 MB, created by the simulation job, had to be copied to tape. Analysis of this need drove us towards a "staging" solution: Instead of keeping a cartridge drive busy for ten hours, the file is stored on a disk local to the station; then at job end, a process writes the file synchronously to the cartridge via mainframe. After studying the relative merits of various ways of transferring these data, we decided to develop our own facility, which we call *ztage*. In response to user demands, *ztage* later was enhanced to provide an interface to VMARCHIVE and to send files to any BITNET node. *Ztage* uses the client/server model over the stream sockets API and meets the desired goals of efficiency and reliability. For instance, if VM is down when *ztage* is called, *ztage* waits for the VM system to come back up again. When necessary, *ztage* also handles the conversion of FORTRAN–generated binary files to the CMS format.

File system organization

One important goal of BASTA's organization was to provide users with a uniform name space for files and to hide underlying operating system differences. User files are physically stored on disks connected to an RS6000, which manages the user file system (/u), mounted via NFS (Network File System) at the same mount point on all the other processors.

Every processor has its own complete operating system and staging area on a local disk. For each architecture there is one copy of the CERNLIB binary files, also mounted via NFS. The CERNLIB sources (.car, .cra and .cmz) are mounted from the mainframe, acting as a file server.

All temporary files created by a batch job (including the 200 MB output file) are written to a user batch directory on a local staging disk, thus facilitating purging of these files at job end. As already mentioned, this directory is created and purged by NQS during execution of the job and therefore exists only during the job.

The *sfget* command used to allocate file space, the method of creating a user batch directory, and the format of the *ztage* command were all done in such a way as to maintain compatibility with the same (or similar) commands on SHIFT at CERN.

User environment

The user environment consists of the C–shell and a number of "scripts" designed to facilitate different operations (compilation, file allocation, FORTRAN file association, etc.) and to hide differences between the UNIX systems. Users log on to a BASTA station to submit jobs; the only supplied interface to VM is the *ztage* command.

User program migration and experience

The Computing Center encourages users to develop their programs in such a way as to be executable on both types of stations. The production job script must guarantee that the proper binary files will be executed on each machine. Environment variables set in profiles called from the C–shell (*.login* profile) are used for this purpose. This is our main reason for using the C–shell as batch shell. The user still has to manage and maintain two different sets of program files (user libraries, images, input data) and tool files (code management directives, compile and load scripts).

The principal difficulties encountered by physicists in migrating their simulation jobs to the chosen UNIX systems were due to the different FORTRAN compilers: problems related to I/O (especially OPEN), automatic loading of BLOCK DATA, problems due to over–optimization. It was found that care must be exercised regarding the preservation of local variables across subroutine calls, the default option being different for HP-UX and AIX. Use of non–optimal FORTRAN options can heavily impact performance but is sometimes required in order to achieve correct compilation of a routine or correct behavior of the program at run time.

Users also perceived system run–time error messages as being unclear. The asynchronous print of standard and error log files is also a problem. However, both interactive debuggers were considered superior to that available under VM/CMS.

Also on the positive side, the BASTA environment allows very easy access to a batch job during execution. It is possible, for example, to consult any file while a job is writing it and also to link interactively, via the symbolic debugger, to the batch process itself in order to examine it, and to stop or release it afterwards.

Status and Performance

Production began in mid–July 1991 with two IBM stations. Since then we have gone thru several hardware upgrades to the actual configuration of nine "batch processors" (6 HP9000/730 and 3 RS6000/550). During the first year of production, over 35,000 UI hours (equivalent to IBM 3090/60S hours) were carried out. Current production is over 6,000 such hours a month. This was more than could be done on the 3090/60S (since upgraded to an ES/9021–820).

The weakest element in BASTA has proven to be TCP/IP on the VM mainframe.

Concluding remarks

The use of powerful but relatively inexpensive UNIX–based workstations for simulation work has been proven viable. In terms of cost of equivalent mainframe CPU, such a station is amortized in less than a month.

For the future, we are considering increasing the capacity of the cluster. A typical simulation job (12 hours of real station time at 90% CPU followed by *ztage* of its 200 MB in 10 minutes) is equivalent to a mean network usage of about 5 KB/sec per station. This is well below saturation and we estimate that from 30 to 60 equivalent stations could be put on the Ethernet, with perhaps some adjustments such as sequencing of the *ztage* transfers. We also plan to implement a production cluster with a mean large–file transfer rate of 40 KB/sec or more per station. For this purpose, we would supply input "ztaging".

As an already heterogeneous cluster, we are of course open to future machine offerings from the same or other constructors.

Currently, five large physics experiments are producing great quantities of data on BASTA and others are migrating their programs. We would like to cite once more the excellent cooperation between physicists and Computing Center personnel which has led to the success of the BASTA project.