Reprocessing H1 Data on a CCIN2P3 computer farm

Ursula Berthon [a], Sylvie Dagoret-Campagne [b,1], Anne Marie Gaillac [a], Evelyne Lebreton [b], Andrea Migliori [a]

[a] *LPNHE, CNRS-IN2P3, Ecole Polytechnique, France*

[b] *LPNHE, CNRS-IN2P3, Université Paris VI-VII, France*

This paper deals with the reprocessing of about $30*10^6$ events of the H1 Collaboration done on the CCIN2P3 computer farm Anastasie in Lyon (France) in spring 97. The system originally developed in DESY had to be adapted to the particulars of CCIN2PR. Especially 2 constraints necessitated a certain number of new developments:

1) working under a batch system environment,

2) working on a distributed Architecture.

The structure of the program and the performances obtained for this reprocessing are given. The scheme presented here could very easily be adapted to a different experiment, allowing to do parallel processing inside of a normal batch system.

Key words: Massive Parallel System ; Networking; Batch environment, Distributed Computing.

1 Introduction

To overcome CPU-power shortage, the H1 collaboration decided in 96 to partly export the raw-data reprocessing task to computer centres outside of DESY. A third of the total event statistic acquired in 96 , i.e. $30 * 10^6$ events, was entrusted to the IN2P3 institute. Previously the full reprocessing [1] was performed on a SGI Challenge multiprocessor machine, using the up to 20 processors of the machine in parallel continuously for about 3 months. On SGI the system consists of a master-program handling a certain number of parallel and independent reconstruction tasks ("H1REC"), and servers delivering and accepting physics events via shared memory.
At CCIN2P3, the main constraint was to use the distributed batch system instead of a dedicated multi-processor machine. The master program is running on a special dedicated service machine, whereas the reconstruction tasks enter the normal batch queues on Anastasie[2]. Its scheme is shown on the Figure 1 and an exhaustive description of its principles is given in reference [3][4].

2 Description of the production system

It consists of an interactive part(Master,Receiver and Sender programs), an input and an output server and a certain number of batch clients running in parallel. The communication of the parallel processors inside a multiprocessor machine via shared memory had to be transformed to communication via the network between batch jobs and a server machine. For this purpose, additional Input/Output servers for the distribution and the collection of the events to reconstruction tasks had to be developed. In order to assure continuous data flow, Client-server architecture was used to deliver events on request. In addition a task-server was written, collecting status information from the different batch clients, and allowing a

[1] Corresponding author.

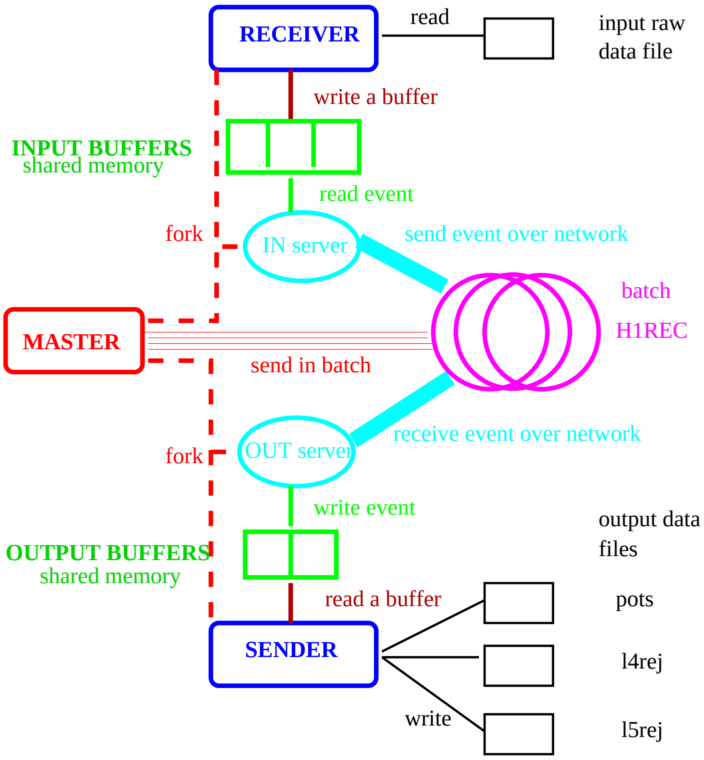

Fig. 1. Scheme of the program (see the explaination in the text).

better control of the system. Raw data are automatically staged-in from DLTs and the reconstructed data are automatically staged-out on cartridge 3490.

The machines we are using, are IBM/RISC 6000 and different sorts of HP 9000 connected by a local ATM network.

2.1 The master

At the beginning, the Master forks the input (Receiver/IN server) and output processes (Sender/OUT server) and submits a certain number of reconstruction tasks to the local batch system BQS [5].

Then in a control loop, it surveys the status of clients in order to keep the number of batch-jobs constant. The status of the Master is also kept in a control file allowing to restart after a stop. Via a steering file, the user can interact with the system to modify some parameters such as number of batch jobs)

2.2 Data Input and output and the event servers

The shared memory of the service machine - splitted into input/output buffers - is used to store events. The I/O operations are made in two steps: on input, the first step is a permanent reading of events from a disk-file into the input buffer in shared memory. This is done by the receiver. The second part, done by the 'IN server', is to deliver an event to a client upon a request. Symmetrically, the client sends events to the 'OUT server', which transfers them into output shared memory. When the output buffer is full, the Sender copies events on output disk files. Access to the buffers is controlled via semaphores.

2.3 The reconstruction task

Reconstruction tasks are normal batch jobs doing event reconstruction, except for the reading/writing of events that is done by requests to the servers running on the service machine outside of the batch system.

3 Performances

The event rate at input and output is proportional to the number of running batch jobs. For 10 reconstruction tasks running in parallel the effective input rate was 4.5 Hz, the output rate was 1.8 Hz and the data flux was of 240 Kb both at input and output. For 25 reconstruction tasks running in parallel the effective input rate was 7 Hz, the output rate was 2 Hz and the data flux was 344kB/s on input and 268kB/s for output. About 60 % of the events are filtered out in the reconstruction tasks. The mean CPU time per event at input is 5.1-7.1 ui (2.4-3.2 seconds).

4 Conclusion

The scheme we have used is a convenient way of doing parallel processing in the framework of a traditional batch system. Only small parts would need to be rewritten to use it for parallel processing for a different experiment in distributed batch farm with a system like NQS for example. Using a client-server architecture to distribute events over the network has been proven to be quite efficient.

Acknowledgements

We express our gratitude to J. Ganouna director of the CCIN2P3 and it team, G Farrache,J Furet,R. Rumler,Y. Fouilhe,W. Wojcik. for a very close collaboration to the project.

References

[1] Data logging and online reconstruction in H1
P. Fuhrmann, R. Gerhards, U. Kruner-Marquis, J-E. Olsson, Z. Szkutnik.

[2] CHEP'94 - San Francisco Apr-94
http://www.in2p3.fr/html/ccin2p3/ef_doc.html

[3] Reprocessing des données à Lyon
U.Berthon,V.Boudry,M.Jaffre,E.Lebreton,JP.Pharabod
13 juillet 95.

[4] Feasibility study for re-processing the H1 data on the IN2P3 computer farm
S. Dagoret-Campagne, E. Lebreton, G. Farrache, Y. Fouilhe, W. Wojcik,J. Furet,
H1-09/96-491,LPNHE-/96-07.

[5] Utilisation des Ressources Unix à l'IN2P3
http://infodan.in2p3.fr/CCHELP.html