PRODUCTION FARM REPORT

Jaroslav Antos, Antonio Wong Chan, Paoti Chang, Yen-Chu Chen,

Stephen Wolbers, G. P. Yeh, Ping Yeh

INTRODUCTION

- The prototype farm is made up of 14 worker nodes (fnpc100-113) and 4 I/O nodes (fnpca, fnpcb, fnpcc, fnpcd), each containing dual Pentium-II 400 MHz CPU's).

- It has been available since October 98. Art Kreymer installed CDF Run II software shortly thereafter. CDF has exclusive usage of 4 worker nodes (fnpc100-103) and 1 I/O node (fnpcc).

- The Production Farm group has been running CDF Run-II Production Executables on the prototype farm since November 98.

- The Production Farm group concluded throughput and reliability tests of the prototype farms in February 99, using all 14 worker nodes.

- The group is designing a conceptual model of the Production Farms software system. Discussions and implementation of the software system is ongoing.

GOALS OF THE 14-NODE TEST

- The goal of the 14-node tests was to study throughput and reliability of the prototype farm, using CDF Run II software.

- The 14 worker nodes were divided into the following queues:

```
fnpc100-103  --> Worker_CDF_1
fnpc104-107  --> Worker_CDF_2
fnpc108-111  --> Worker_CDF_3
fnpc112-113  --> Worker_CDF_4
```

- We performed various throughput tests by copying large files ($\sim$ 320 MB) back and forth between the worker nodes and the I/O node via “rcp” and “scp” in a variety of ways.

- The reliability test was done by running CDF Production Executables on the prototype farms for extended periods of time (2-8 hours). Longer tests (1-7 days) may be scheduled in the future.

14-NODE TESTS

- Test 1 - this consisted of copying 1 to 24 ($\sim$ 320 MB) files from I/O node to Worker_CDF_1 and then back from Worker_CDF_1 to I/O node simultaneously. All copying was done via rcp.

- Test 2 - the same as test 1, but this was done on Worker_CDF_2 and Worker_CDF_3.

- Test 3 - the same as test 1, but this was done on all 4 queues (Worker_CDF_1, Worker_CDF_2, Worker_CDF_3 and Worker_CDF_4).

- Test 4 - the same as test 3, but this was done with "scp", instead of "rcp".

- Test 5 - the same as test 1, but the copying between I/O node and Worker_CDF_1 was divided between 3 disks on the I/O node.

- Test 6 - run 23 Production Executable processes at the same time on all 14 worker nodes, which includes reading the input file from I/O node to worker nodes, event reconstruction and writing the output from worker nodes to I/O node. All I/O is done via rcp.

- Test 7 - same as test 6, but with 46 Production Executables processes on all 14 worker nodes. The first 23 are submitted, and the remaining 23 sit in the queues until the first 23 finish. All I/O is done via rcp.

TESTS RESULTS

Each process copies the input file for the CDF Production Executable ($\sim$ $320MB$) from /cdf/data1/cdfprod0 on the I/O node to the scratch area ($PC_BATCH_SCRATCH) on the worker nodes. After the initial copying is finished, the process copies back the file from the scratch area to /cdf/data2/cdfprod0 on the I/O node.

The table above shows the average rate per process, so the total rate should be $(rate) \times (\#\ of\ proc)$. The highest rate obtained in this case is 13.3 MB/s for 4 processes.

Table 1: Summary of test 1

# of proc.	Job ID	Rate/proc.	%CPU (avg.)
1	4310	7.40 MB/s	46.4%
2	4312	5.61 MB/s	39.1%
4	4314	3.32 MB/s	31.3%
8	4302	1.24 MB/s	16.5%
16	4304	0.54 MB/s	10.0%
24	4308	0.33 MB/s	6.5%

The queues Worker_CDF_2 and Worker_CDF_3 are somewhat slower than Worker_CDF_1. I don't know how significant this is without running more tests and ruling out statistical fluctuations due to usage of the system at the same time by other people.

Table 2: Summary of test 2

# of proc.	Job ID	Rate/proc.	%CPU used (avg.)	Queue
4	4419	2.50 MB/s	18.5%	Worker_CDF_2
4	4321	2.54 MB/s	20.1%	Worker_CDF_3

The * indicates where we had “rcp time out problems”, which means the “rcp” copying process aborted before the file copying process was over. There seems to be a limit on how many processes can be running on the I/O node at the same time (between 28-56). I don't know if the “rcp” problem is due to too many processes or due to an intrinsic “rcp” limitation. Some of the “rcp” processes don't run at the exactly same time, so one cannot infer from this table the total rate by simply multiplying the rate/proc. by the # of proc.

Table 3: Summary of test 3

# of proc.	Job ID	Rate/proc.	%CPU used (avg.)
1-1-1-1	4335	5.66 MB/s	32.5%
4-4-4-2	4371	1.10 MB/s	9.1%
6-6-6-6	4341	0.75 MB/s	7.9%
8-8-8-4	4377	0.40 MB/s	9.6% *
12-12-12-6	4383	0.25 MB/s	6.6% *
12-12-12-12	4365	0.20 MB/s	5.9% *

The usual “scp” version (not the faster “blowfish” version) was used for this test. Again, I saw that “scp” is 3 times slower than “rcp”. However, there were NO time-out problems with “scp”.

Table 4: Summary of test 4

# of proc.	Job ID	Rate/proc.	%CPU used (avg.)
8-8-8-4	4423	0.16 MB/s	0.49%

This test was done by dividing the input files to be copied among 3 disks on the I/O node (/cdf/data1/cdfprod0, /cdf/data2/cdfprod0 and /home/cdfprod0). Both the copying to the scratch areas of the worker nodes and the copying back to the I/O node was divided among the 3 disks. When the load is divided (a), both "rcp" processes are equally fast, but if the load is put on a single disk (b), then the latter "rcp" process is several times slower. This may be due to all "rcp" processes trying to access the same disk at the same time, reaching the limit of the disk (or SCSI bus) I/O very quickly. When the load is divided, the average transfer rate is $\sim$ 20.4 MB/s.

Table 5: Summary of test 5 (a)

# of proc.	Job ID	Rate(in)/proc.	Rate(out)/proc.	%CPU (in)	%CPU (out)
8 (a)	4874	2.38 MB/s	2.73 MB/s	16.2%	15.4%
8 (b)	4880	3.95 MB/s	0.70 MB/s	48.5%	3.2%

This checks the times when the various processes in Job ID 4874 started and ended. Although the jobs started sequentially, the overlap is considerable, so our throughput rates are a good approximation of the true performance of the system.

Table 6: Summary of test 5 (b)

# of proc.	Job ID	Section #	Time started	Time ended
2	4874	test1	17:12:24	17:15:47
2	4874	test2	17:13:37	17:18:16
2	4874	test3	17:14:53	17:19:50
2	4874	test4	17:16:01	17:21:52

The total CPU time for each process is $\sim 9,930$ sec on average, while each process takes $\sim 10,502$ sec of clock time on average. The clock time includes job scheduling, job start-up and end by FBS. This gives an average CPU utilization rate of 94.6%.

The first Production Executable I ran on the prototype farm in November 98 took about 7.0 CPU sec/event. In January 99, it became 7.8 CPU sec/event, and now it takes about 9.8 CPU sec/event. In parallel, the November 98 Production Executable used about 18.6 MB of memory. In February 99, it became 25 MB. The one used in the 14-node test used 46 MB. It's growing very fast, and this may have an impact on the specifications of the worker nodes we buy for the Run II farms. We need to keep an eye on this.

Table 7: Summary of test 6

proc. #	Job ID	Rate/proc.	%CPU	CPU time/event	Clock time/event
1	4389	4.24 MB/s	12.56%	9.87 sec.	10.12 sec
2	4389	3.28 MB/s	12.86%	9.81 sec.	10.07 sec
3	4389	4.55 MB/s	13.10%	9.78 sec.	10.06 sec
4	4389	4.72 MB/s	14.48%	9.84 sec.	10.16 sec
5	4389	4.32 MB/s	24.94%	9.88 sec.	10.22 sec
6	4389	4.33 MB/s	24.90%	9.84 sec.	10.18 sec
7	4389	5.04 MB/s	25.46%	9.85 sec.	10.23 sec
8	4389	3.59 MB/s	24.78%	9.90 sec.	10.30 sec
9	4389	3.32 MB/s	12.66%	9.81 sec.	10.20 sec
10	4389	3.12 MB/s	13.62%	9.81 sec.	10.24 sec
11	4389	3.45 MB/s	13.61%	9.80 sec.	10.24 sec
12	4389	3.70 MB/s	14.30%	9.77 sec.	10.22 sec
13	4389	5.25 MB/s	25.54%	9.80 sec.	10.33 sec
14	4389	3.97 MB/s	19.74%	9.81 sec.	10.34 sec
15	4389	4.67 MB/s	15.74%	9.84 sec.	10.42 sec
16	4389	4.04 MB/s	16.06%	9.71 sec.	10.23 sec
17	4389	3.97 MB/s	13.52%	9.86 sec.	10.40 sec
18	4389	3.37 MB/s	12.28%	9.79 sec.	10.37 sec
19	4389	4.05 MB/s	13.32%	9.85 sec.	10.53 sec
20	4389	3.74 MB/s	13.20%	9.74 sec.	10.29 sec
21	4389	4.60 MB/s	24.56%	9.79 sec.	10.45 sec
22	4389	4.61 MB/s	24.84%	9.79 sec.	10.51 sec
23	4389	4.24 MB/s	19.78%	9.77 sec.	10.47 sec

The rates shown are the average over each of the 23 Production Executable processes and are comparable to the individual rates shown in table 6. These are two consecutive 23-processes jobs on the 14 worker nodes, taking a total of 8 hours. The system was able to sustain the continuous processing with an average CPU utilization rate of $\sim 95.1\%$, which includes the job scheduling, job start-up and end by FBS. The addition of CDF catalog, calibration, trigger database accesses, etc is likely to add time to the process. The impact of these tasks on the CPU utilization rate should be minimized by carefully designing the access mechanisms.

Table 8: Summary of test 7

proc. #	Job ID	Rate/proc.	%CPU	CPU time/event	Clock time/event
23	4441	4.35 MB/s	20.59%	9.86 sec.	10.12 sec
23	4559	3.94 MB/s	18.69%	9.87 sec.	10.26 sec

CONCLUSIONS AND FUTURE PLANS

- We have learned a lot with our 14-node tests, specially on the throughput tests.

- The data transfer rate between our I/O node and the worker nodes in a realistic scenario is ~ 20 MB/s. Note that the current prototype farm is not our final choice of hardware standards for the I/O node or switches. We are encouraged by the results we obtained, and we believe that a different system using newer technology will allow us to achieve higher rates comfortably.

- Our average CPU utilization rate when running the CDF Production Executable on the prototype farm is $\sim 95\%$, including the time needed by FBS to schedule, start and end a process.

- The Production Farm project is ramping up. The development of the Production Farm software will require frequent interactions with the rest of the CDF Offline (CDF Catalog, Calibration files, Tape I/O, Simulation, etc) and Computing Division.

QUESTIONS TO OFFLINE GROUP

1) There will be another code release in May. Will this release include some of the hooks necessary for the rest of Offline (i.e., Farms) to access Calibration, Trigger Database, CDF Catalog, etc? Of course, we (the Production Farms group) will be involved in shaping some of these issues (i.e., CDF Catalog) in some fashion, but my question is "Is there a plan to include some of these hooks for the May release?"

2) Another aspect of our testing of the prototype farm will involve tape I/O. Our current estimate is that we'll start our "Tape I/O integration" in May/June. Is this in line with the Data Handling group's estimate?

3) Our specification of the worker nodes will have to be made soon, and we expect them to be delivered by October 1999. One aspect of the specification is the memory size for each processor. Our current specifications call for 256 MB/processor, and the ability to upgrade it by a limited factor. Based on memory usage in Run I and best guesses for Run II, will this be adequate for the Run II Production Executable?

4) The prototype farm has enough CPU to serve as a small production facility for Monte Carlo. This could help the Production Farm group to test production-like conditions, to establish possible techniques for simulating events on a large scale and to test production control software. Is there any interest or need for this on the collaboration's part?

5) At some point we will need to test our production control code for splitting/merging of output data. This is dependent on first defining the output streams on Run II code and defining a mechanism by which the production control software can access this information. When can we expect this to occur?