COMPASS Computing Farm Project

Index

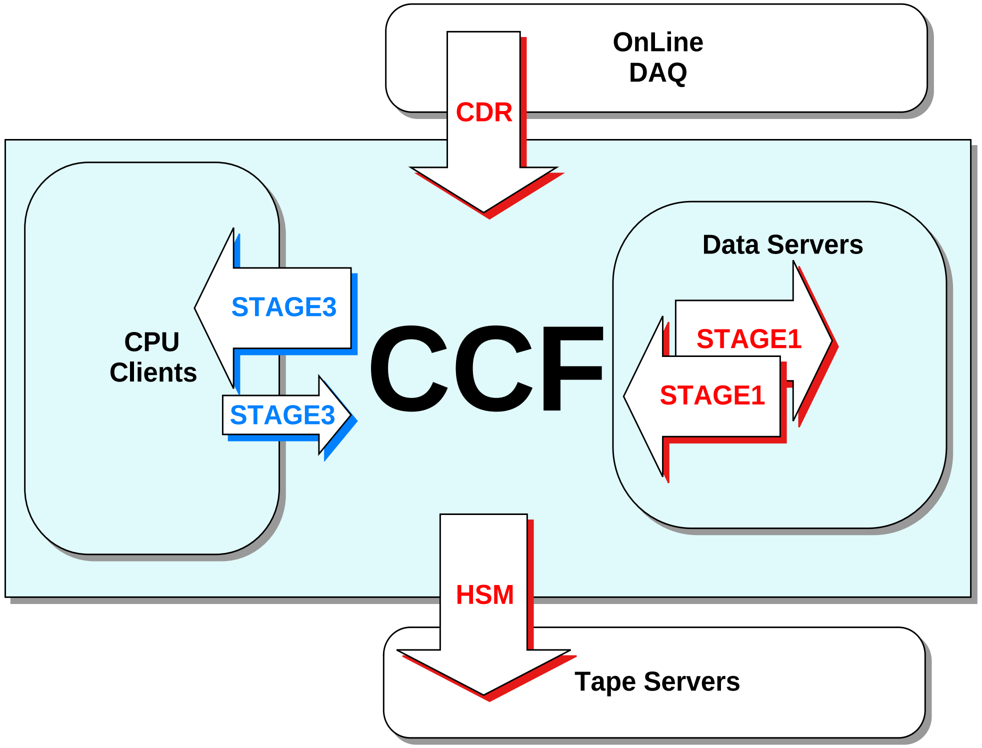
- The CCF project
- The CCF model
- The CCF software
- DAQ mode at 35 MB/s tests
- Quasi-online data processing mode tests
- Conclusions and outlook

CCF project

- 35 MB/s of data out of the experiment DAQ: use of the Central Data Recording approach (NA48, NA45, Test Beam)
- 20,000 CU (~ 100 PCs equivalent computing power) needed to reconstruct the events at the same speed of the DAQ: quasi-online processing model
- PC hardware has to be deployed to provide the bulk of the computing power
- C++ reconstruction program (Coral framework; see A. Martin presentation at this conference)
- C++ objects stored as such in a hierarchical way (Objectivity/DB). New technology to be understood
- Managing a large distributed system (PC + few TB disk space) in connection with a tape service, with a Hierarchical Storage Manager (HSM) layer (HPSS, Castor)

CCF model

- Data servers and disk pool
  - Network and disk traffic
  - Unix (DEC and SUN, now Linux PC)
  - few TB SCSI disk pool (EIDE in future?)
- CPU Clients
  - Number-crunching PCs (WNT, now Linux)
- Network technology
  - Gigabit and Fast Ethernet
- Data flow model
  - Data set driven
  - Run -> 10-20 streams of parallel transfers -> parallel population of DBs and reading (for reconstruction



DAQ mode (only red arrows)

Quasi-online processing (red and blue arrows)

Other modes (DAQ + playback from tape)

CCF software

The construction, the tests, the deployment, the operations, the maintenance, the upgrades of a system with more than 100 major components, require automatic procedure and a deep understanding of the dependencies among all the building blocks (hardware and software).

The CCF control software (written in Perl) should cope with these tasks, notably:

- Operate and monitor the CCF (from the selection of the hardware to error recovery)
- Steer the reconstruction programs
- Interact with the data storage technologies: Objectivity/DB and the HSM

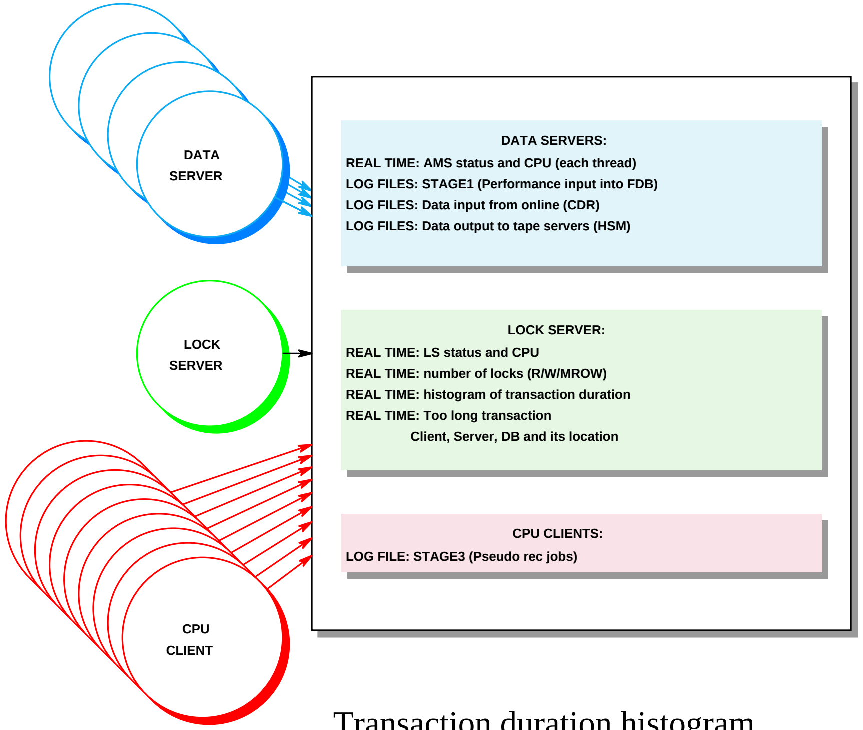
Software tools

- Installation
  - Responsibility of CERN IT/PDP
  - Standard procedures to install consistently a large number of machines have been set-up
  - CERN SUE and ASIS mechanisms (ASIS repository under CCF control); in production, all software will be installed on each node (no AFS)
  - Standard mechanisms to monitor daemons and operations
- Quality assurance
  - IT/PDP and CCF team
  - Run test suites on all nodes (for Objectivity/DB on all pairs of nodes to test network access)
- Benchmark tools
  - Pragmatic approach (e.g. suites to measure the disk speed on a server with different combinations of read and write streams; measure the network connectivity among different hosts)

Software tools

- Operations are “run based”
  - job-based output (log files)
- “Real-time” tools complement this picture
  - Run on Linux and Solaris
  - simple UDP communication system
  - Use Objy tools (e.g. `oolockmon`), kernel informations (e.g. CPU usage) and allow correlation (via time)
  - Example:

```
ools lock 949086192 1 0 0
ooams ccf005 949086196 10 2 5 0 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0
ooams ccf009 949086210 6 2 3 0 0 0 2 1 2 1 2 1 2
locks lock 949086210 20 4 0
ALARM lock 949086210 17 min : Host: ccf020 User: 15197 PID: 20079 TID: 359333887
          Mode: read DB ID: 15162 cdr12072000
          ccf010::/shift/ccf010/data01/objsrvy/na58/cdr12072000.na58fd.DB
```

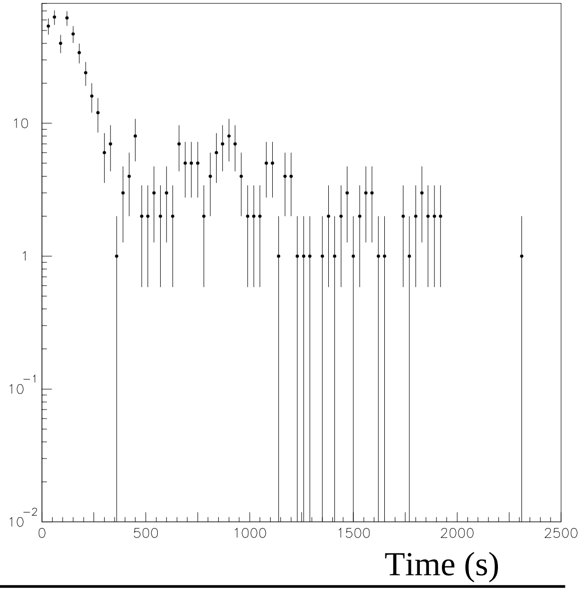


DATA SERVERS:
REAL TIME: AMS status and CPU (each thread)
LOG FILES: STAGE1 (Performance input into FDB)
LOG FILES: Data input from online (CDR)
LOG FILES: Data output to tape servers (HSM)

LOCK SERVER:
REAL TIME: LS status and CPU
REAL TIME: number of locks (R/W/MROW)
REAL TIME: histogram of transaction duration
REAL TIME: Too long transaction
Client, Server, DB and its location

CPU CLIENTS:
LOG FILE: STAGE3 (Pseudo rec jobs)

Transaction duration histogram



CCF prototype at CERN

25% of the final CCF prototype

installed in the CERN computer centre

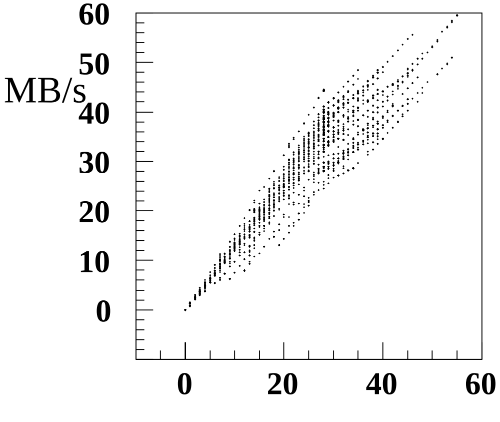
35 MB/s test in DAQ mode

- Data server stage: 11 PCs with ~ 1.5 TB disk space
- About 35MB/s mock data sent from COF to the CCF data servers (CDR stage) in ~30 concurrent parallel streams using the RFIO software
- CCF data servers convert the data in Objectivity/DB data base (Stage1). This stage is performed locally (using the Data servers CPU: no AMS involved). This is a choice to leave the AMS free to serve the CPU clients
- Data bases sent to some machines simulating the tape servers (HSM); the original data deleted when the corresponding events finished HSM stage
- The data bases are erased from disk (and the corresponding federation database updates are performed)

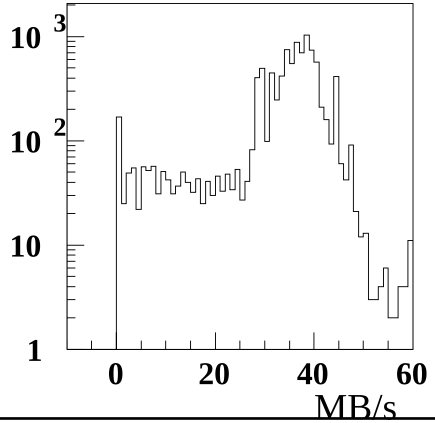
35 MB/s results

- The behaviour of the CDR system during 6 hours is shown. The CDR system rate has a negligible idle time (the transfer of the data of a run is finished just in time for the next run transfer)
- The stage1 system: a mean number of 17 Objectivity/DB clients write concurrently into the same federation. The typical rate of a writer (local I/O) is 2.4 MB/s (up to 4 concurrent stage1 per PC). The idle time is < 2%

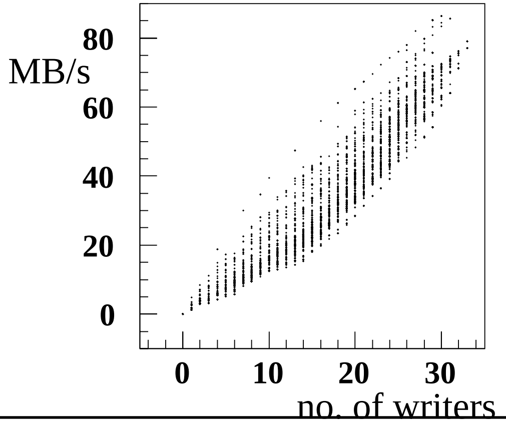
Transfer rate vs no. of streams



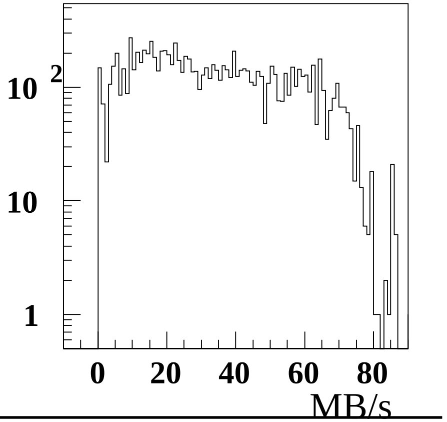
Transfer rate



DB speed vs no.
of processed (Objy writers)



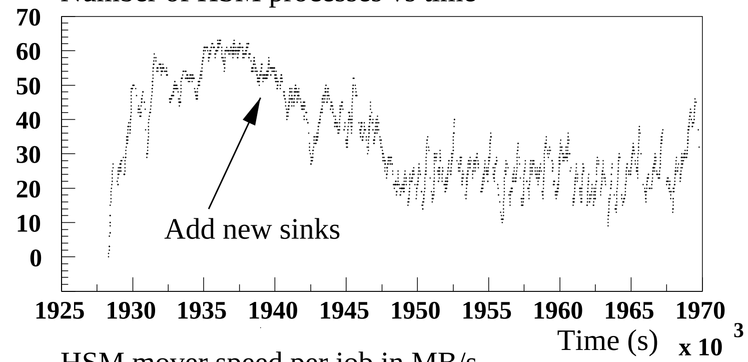
DB speed (Objy writers)



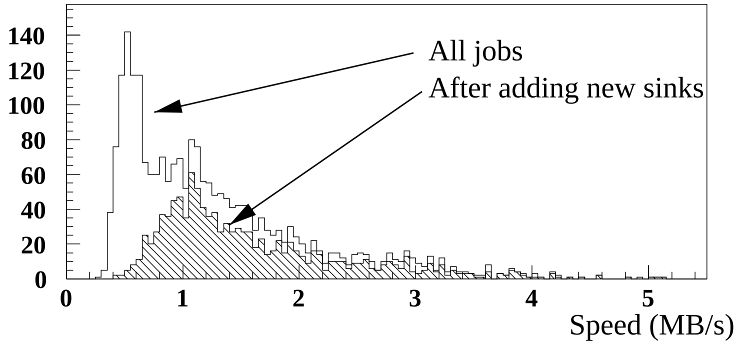
35 MB/s results

- The CCF prototype here is capable to sustain the Objectivity/ DB rate concurrently with the CDR (input) and the HSM (output).

Number of HSM processes vs time

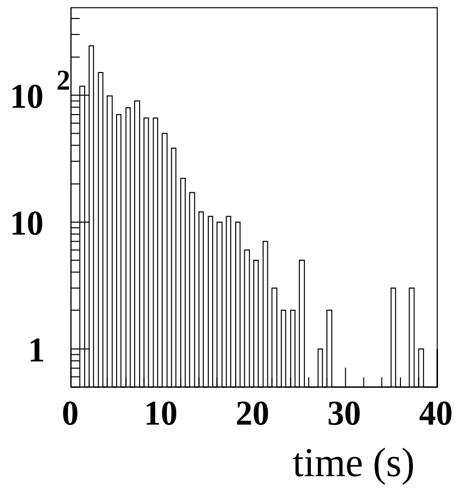


HSM mover speed per job in MB/s

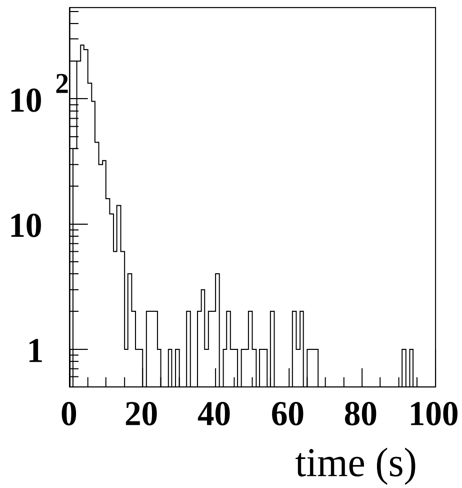


- Extensive monitor of DB quantities in realistic conditions

Data base file creation



Containers creation



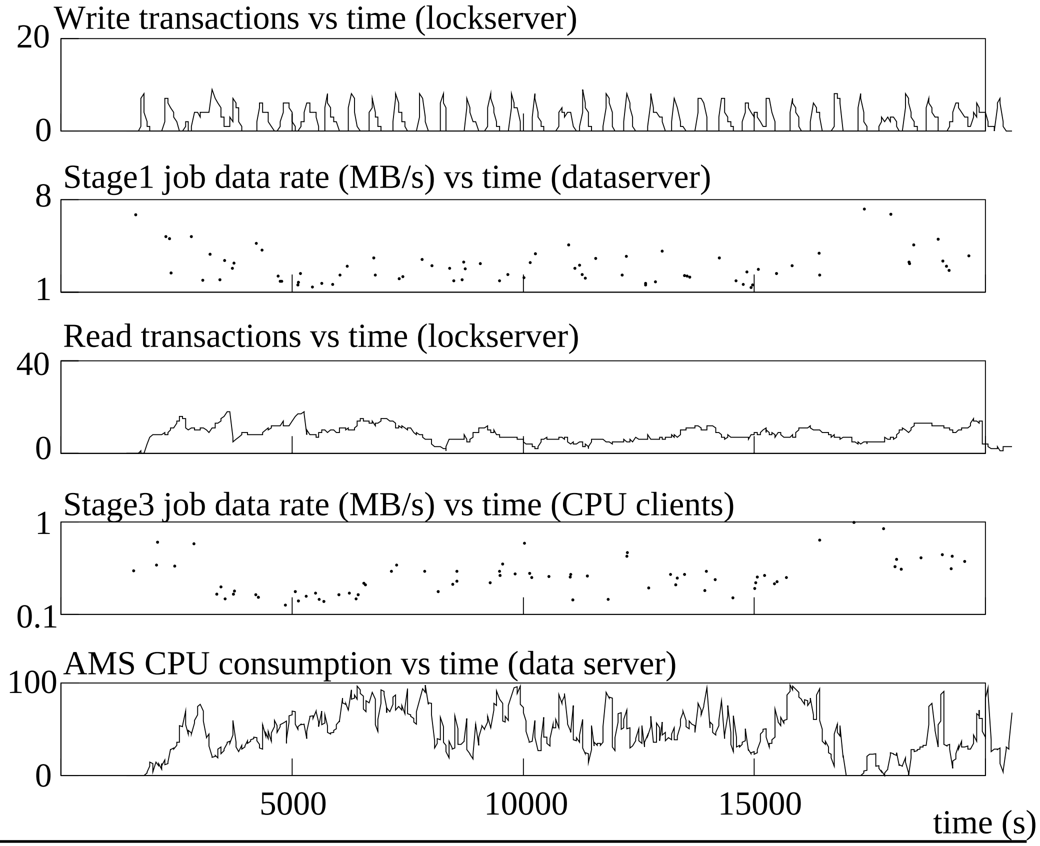
- Minor lock problems which can be recovered; possibly understood.

CCF quasi-online processing

- The data flow is similar, but the CPU clients have to scan *all* data bases via AMS before stage out from the data server’s disks.

- Objectivity/DB 5.2 (first release with multi threaded AMS)
  - Data servers: 4 PCs with ~ 0.8 TB disk space; 25 CPU clients; LSF selects the CPU clients
  - ~25% of the hardware available -> ~25% of the rate (8-9 MB/s): up to now bound ~ 7.5 MB/s due to high CPU consumption in the AMS (known bug)

- Operational problems with the Objectivity/DB 5.2 (bug fix expected)



Conclusions

- The Compass Computing Farm is (at the 99% level) a Linux PC farm
  - Fast evolving PC technology
  - Linux!!!
- Monitor tools
  - Nice experience; we think they are crucial to commission the farm
- Very nice results in DAQ mode (35 MB/s test)
  - Basic mode of operation achieved
- Encouraging results in Quasi-online mode so far (25% of the farm)
  - AMS much improved; maybe multi threading support not yet fully mature
- COMPASS technical run starts in May 2000!
  - CCF commissioning and test before
  - Operation at 100% during part of the run