Padova

Padova, Thuersday 22 April 2004

First experience with ORCA Analysis on Grid

a user point of view

Stefano Lacaprara

Stefano.Lacaprara@pd.infn.it

INFN and Padova University

Job

- ORCA 800,
- Access to Digis formerly produced at LNL (tt2mu),
- Access to DST (tt2mu) transferred from CERN via CNAF
- Simple jobs: printout plus histograms,
- Private library and executable,
- Submission from PD UI,
- No data discovery, jobs forced to go to LNL,

Job Preparation

- Code development on local machine (my own),
- Test of code running on locally produced data (SingleMuon, available in PD via RFIO),
- Copy of library, executable and `.orcarc` on UI (`gridit003`)
- Job preparation script reusing private code (`perl + bash`) written long ago,
- Changes to produce `jdl` : trivial (with Federica's help!),
- Got GRID certificate (not so easy, even if rather documented)
- Get `proxy`, and submit to RB: CNAF or CERN when CNAF down: some magic (Federica!)

Job Preparation (ii)

- What the job does:
  - Source script to set up environment (Marco)
  - create ORCA 800 area (`scram project`) on WN, using local ORCA installation (M)
  - copy (via input sandbox) tarball with lib(s) and exe
  - move libs and executable to proper places (some ORCA/scram expertise needed),
  - get `.orcarc` fully set up via sandbox
  - Execute job
  - put root file in output sandbox

Job submission

- Single job directly via `edg-job-submit`
- Get job id from terminal (mouse cut and paste!)
- Get job status via `edg-job-status` using “mouse–recorded” id
- Get job output sandbox when status done, always via mouse
- For multiple submission (up to 100 jobs in parallel) used a `perl` script (written long ago for LSF, adapted)
- Save id’s on a file
- Wrote a (rather complex) `perl` script to retrieve multiple job status and sandbox if all ok

Data

The Real Mess!!!

- Digis available at LNL since long time (PCP)
- Missing: MetaData with Digi (and SimHits) attached
- Missing: PoolCatalog with PFN of all files location
- Stole (I mean **really** stolen!) full MetaData from CERN
- Produce Catalog from stolen one updated for LNL EVD and MetaData: partially via Pool commands (too slow and complex) large via editor (and large use of RegEx)
- Put Catalog(s) on defined place
- Set `InputFileCatalogURL` by hand to proper catalog(s)

Data (ii)

THE REAL MESS!!!

- DST available at LNL: pushed from CNAF
- Missing: MetaData with anything attached
- Missing: PoolCatalog with PFN of all files location
- Full MetaData not available nowhere
- Deep Winter Mode Access: no run attached!
- Run `FixColls` (COBRA tool) directly on collection EVD run per run (Marco)
- Get **oid** and put it (them) in `.orcarc`
- Done for a couple of runs (~ 5000 events), resulting in a multi–line, very complex and error prone entry in `.orcarc`
- Catalog built by hand (M) and set by hand in `.orcarc`

Results

positive

- The machinery, however complex, can be forced to work
- Job submitted via grid to LNL
- Job execution (after some job debugging iteration)
- Job submission and execution overhead not dramatic (but no data discovery)
- Can get back the results

neutral

- No real Grid job!
- Job forced to run at LNL
- Data prepared by hand(s) (DC04 problem, not grid)

Results (negative)

- Develop on a machine, move all tested code to a UI and then submit job from it
- A generic user machine must be allowed to submit to grid, ie to be a UI (in principle possible, via a set of rpm's + script, not tested)
- Interface to Grid service not friendly
- output `edg-whatever` designed to be human readable, not script readable (eg multi line...)
- What if I submit a job and lose the id? Grid-leak?
- Sometimes job submission failed, need expert to see why (error message meaningless)
- Problem with RB unavailability: need expert to switch to other one (must be automatic!!!)
- Must source by hand script to get CMS environment (VO==CMS)

Results (negative) (ii)

- No comment on DATA availability and information flows: not much to do with grid, a lot with CMS computing model (?)
- A amazing lot of people, expertise, magic, stealing etc to have something usable, and only to real expert
- Absolutely not for end–user/analyst
- Need work to deal with jobs id's, jobs status querying and sandbox recovery
- Developed ad–hoc script to handle multiple jobs
- job return status mostly meaningless: crashed jobs ok, good jobs reported as bad

Results (negative) (iii)

- Getting the output sandbox is a nightmare!!!!
- Must ask one by one when the job is declared to be over
- Only partial control on where get back the results (default is `tmp`, can easily crash the UI, no scalable at all!!)
- I want the job to push back the output when finished
- I guarantee the availability of UI
- I'm ready to lose all output if UI off-line, much better that have to retrieve all outputs one by one, move it to a decent place and eventually change the name (all by hand)

Future

- Most depends on DC04 data availability in a decent way
- Deep Winter Mode is not for user
- Can think to attach run at Tn if T0 will not do it
- Want to have a local catalog available and up to date with local PFN
- Data discovery cannot be done on a file basis
- No matter what will be the performances of RLS, my “typical” job will require $\mathcal{O}(10^5)$ files, not thinkable to search for all of them each time!!!!
- Current RLS implementation is similar to a filesystem w/o directly
- All files (can be $\mathcal{O}(10^6)$) on /
- Idea of directories to sort files out since early $'70$

Future (ii)

- Get DST (a full dataset) in a Tn
- Get all Full MetaData as well
- Produce (by Tn) a catalog with all PFN of MetaData and EVD: only once, (eg from RLS)
- Publish the local catalog (Tn dependent) on RLS
- Generic user ask for DataSet/Owner
- Query the RLS for catalogs for catalog containing that D/O (may be in RLS MetaData) just one file (or fews)!!!
- Put the result of the query in `.orcarc`
- Use the result of the query to decide where to run
- Run the executable

Future (iii)

- What if (part of) a Dataset in different location?
- Can have RLS MetaData stating which event are available from a catalog, and also which type (AOD, DST, Digis, MC)
- In case of full dataset access, split jobs according to RLS metadata of catalogs for user required dataset/owner
- LNL catalog has event $1 \rightarrow 1000$, PIC $1001 \rightarrow 2000$, CNAF $1 \rightarrow 2000$
- Implement sort of directory structure in RLS

Future (iv)

- Short time scale (before Aachen Muon week?) test should be possible
- Basic tools already tested and more or less usable
- In case, can force running on given Tn
- Pros
  - Allow user access to data via grid,
  - use grid data discovery,
  - should have reasonable performance (just fews files to be found),
  - should even scale
  - can even cope with job splitting
- DATA MUST BE REALLY AVAILABLE!