Grid and CDB

Janusz Martyniak, Imperial College London

MICE CM37 Analysis, Software and Reconstruction

MICE Data Handling on the Grid

- Raw Data Mover – upload raw data files onto safe tape storage as soon as the data have been written out by the DAQ,
- MAUS reconstruction:
  - offline reconstruction,
  - batch reconstruction (reprocessing).

Raw Data Mover
initial conditions

- We want to copy data files to tape as soon as they are available on the data acquisition box,
- Before data transfer can begin data for each run are compacted – placed into a tarball, including a list of files and their md5 checksums to allow integrity verification further down the processing chain – when ready a relevant run dependent semaphore is placed,
- The Data Mover will start only if no other DAQ writing activities are taking place.

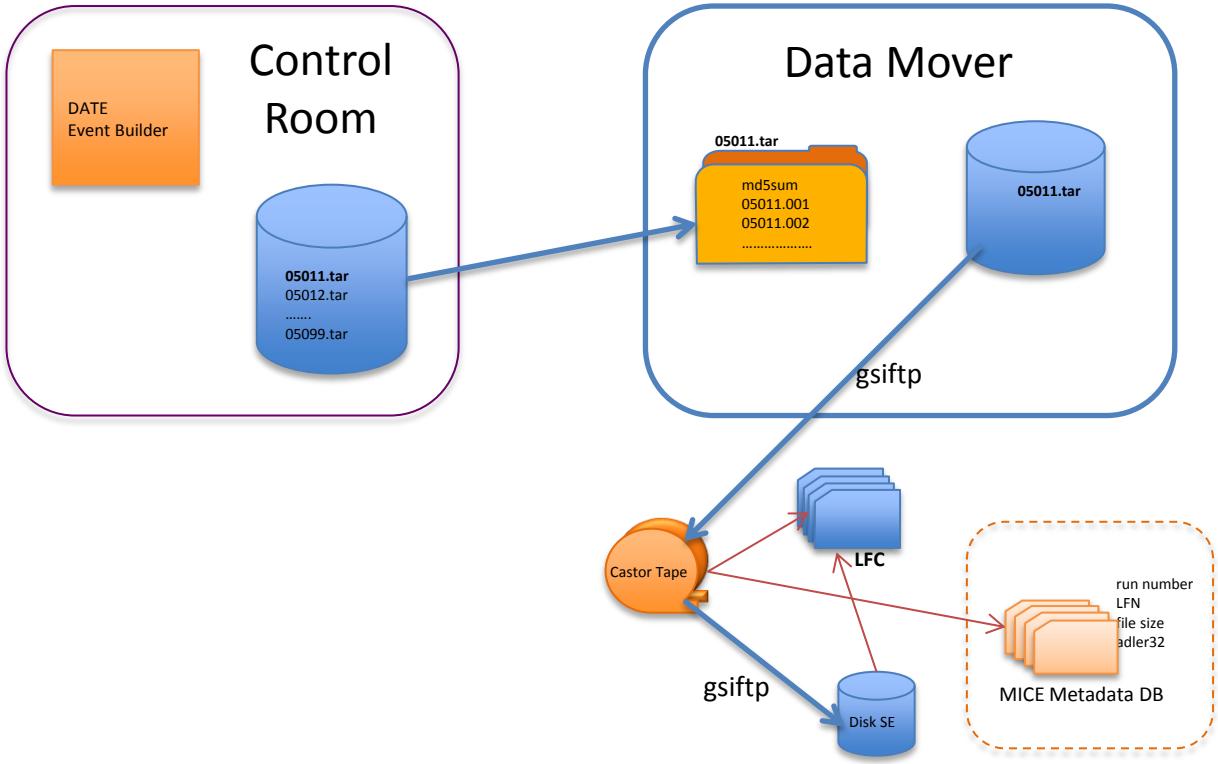
Raw Data Mover (1)

- The Mover has 2 chained parts – the initial copier/file integrity verifier and the Castor uploader,
- The directory where the Compactor places data and run dependent semaphores is actively watched by the Mover and subject to conditions outlined on the previous slide, an initial copy to an intermediate storage is made,
- After the copy internal data file integrity is checked, Btw, we had 3 cases (in ~5000files) where the integrity check failed. In 2 cases we could fix the file.

Raw Data Mover (2)

- Successful initial copy triggers the Castor upload process,
- Files are placed on 2 tapes for redundancy,
- We replicate files to the PPD SE (this step will be removed),
- Checksum (Adler32),
- Register with the LFC,
- Register with the MICE Metadata DB.

Raw Data Mover Workflow



Raw Data Distribution

- We run a FTS based data distribution program at Imperial to pull data from Castor to IC,
- Triggered by new records in the MICE Metadata DB,
- Publish on the WEB at Imperial,
- Also use it for reconstructed data,
- Could be used to download data to other sites.

Raw Data Mover

Summary

- Developed in summer 2011, run since Oct 2011,
- Successfully used to store data on Castor to date,
- Uses Python API which requires rather old gLite UI. Newer UI broke the APIs, never fixed to my knowledge,
- Have to test a new SL6 UI,
- Move the code to a new SL6 hardware,
- Use robot certificate, currently my personal,
- s/w on bzr at RAL, acq. box and IC -> launchpad.

MAUS Reconstruction on the Grid

- GRID MAUS installation via CVMFS at RAL,
- Propagated to Brunel and Imperial,
- Was SL5, now SL6, MAUS_v0.6.0,
- MAUS is self contained, thus relatively easy to install on the Grid,
- Only small modifications to the configure script were necessary.

MAUS Grid Reconstruction Framework

Requirements:

- Triggered by successful raw data upload to Castor,
- Processed at RAL T1, but also at remote sites, mainly for reprocessing,
- Reconstructed data shipped back to RAL and placed on Castor tape.

MAUS Grid Reconstruction Framework

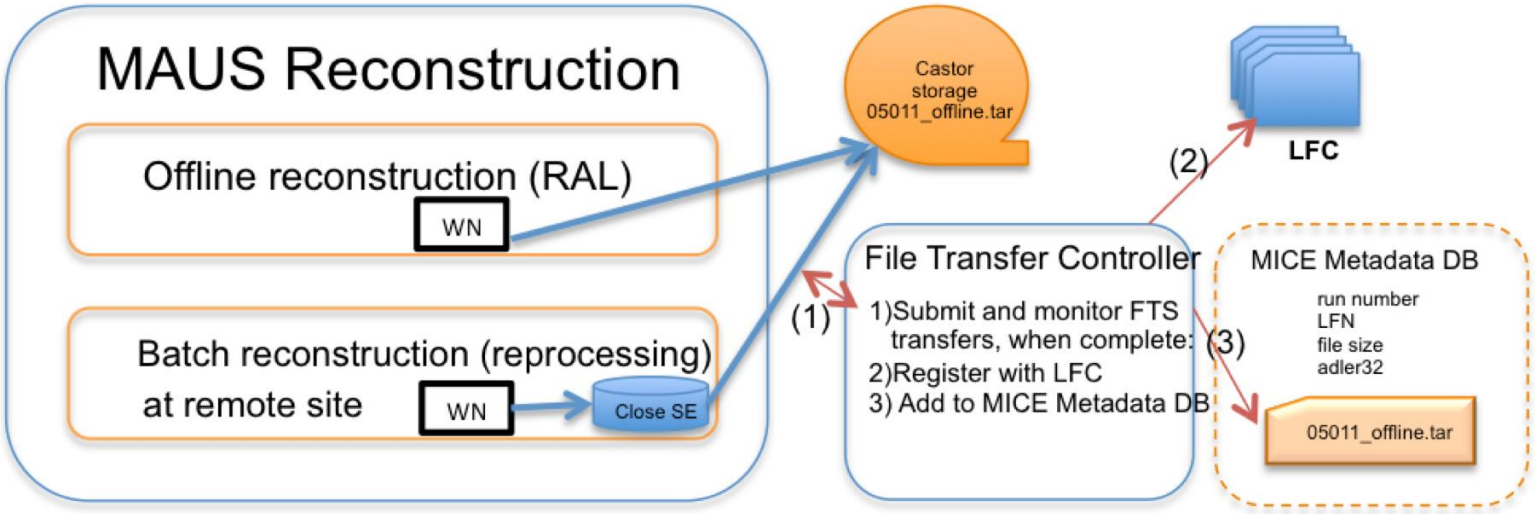
Implementation

- Job submission system periodically looks for new raw data records in the MICE Metadata database,
- Long term Grid proxy stored at MyProxy server,
- Offline reco jobs submitted to a fast, dedicated T1 queue at RAL,
- Reprocessing done at remote sites,
- Keeps track of submitted jobs, completed jobs,
- Failed jobs may be resubmitted,
- Output files from successful jobs are scheduled to be shipped back to RAL by a dedicated File Transfer Controller.

File Transfer Controller

- For RAL jobs it serves just as a MICE Metadata catalogue WS interface, there is no file transfer involved – RAL Castor is the *close* SE,
- For reprocessing the Controller registers a file transfer requests from Grid jobs,
- File transfers back to RAL are made asynchronously by the Controller (backend is implemented as a FTS client),
- We eliminate a potential point of failure – the Controller can resubmit file transfers easily and monitor them. It also registers the Castor replica with the LFC and MICE Metadata database.

Reconstruction Workflow



MAUS Reconstruction, contd.

- Successfully reconstructed data from the last run, done practically w/o delay,
- Reprocessed all available raw data with the, latest version of MAUS at Imperial and Brunel (2 weeks of running).

Reconstruction, todo

- Add more sites for reprocessing,
- Add software tags per CE – for now Brunel and Imperial well tested (for MAUS version and raw data presence),
- Modify MAUS to allow user defined tmpdir *outside* the MAUS installation area,
- This would allow to run MAUS tests at sites – to allow efficient software tagging,

Configuration Database

Project inherited from Antony Wilson

The database is the central place where vital MICE configuration data are stored:

http://micewww.pp.rl.ac.uk/projects/configdb/wiki#The-Configuration-DB-Wiki

- The database (postgres) is hidden from the users by the Web Service,
- Python client API is provided,
- There are 2 DB instances, master (Control Room) and slave (PPD), any updates are automatically pushed to the slave,
- Only Control Room has write access (to master),
- There is a read only replica (PPD).

Configuration Database

My duties include:

- Server maintenance and development (like schema changes etc.),
- WS interface development,
- Python API maintenance and development
- Resolving CDB related issues.

a lot of work ….