MICE data handling on the Grid

J Martyniak, for the MICE Collaboration,
Imperial College London, Blackett Laboratory, London SW7 2BW, UK
janusz.martyniak@imperial.ac.uk

Abstract. The international Muon Ionisation Cooling Experiment (MICE) is designed to demonstrate the principle of muon ionisation cooling for the first time, for application to a future Neutrino factory or Muon Collider. The experiment is currently under construction at the ISIS synchrotron at the Rutherford Appleton Laboratory (RAL), UK. In this paper we present a system – the Raw Data Mover, which allows us to store and distribute MICE raw data – and a framework for offline reconstruction and data management. The aim of the Raw Data Mover is to upload raw data files onto a safe tape storage as soon as the data have been written out by the DAQ system and marked as ready to be uploaded. Internal integrity of the files is verified and they are uploaded to the RAL Tier-1 Castor Storage Element (SE) and placed on two tapes for redundancy. We also make another copy at a separate disk-based SE at this stage to make it easier for users to access data quickly. Both copies are checksummed and the replicas are registered with an instance of the LCG File Catalog (LFC). On success a record with basic file properties is added to the MICE Metadata DB. The reconstruction process is triggered by new raw data records filled in by the mover system described above. Off-line reconstruction jobs for new raw files are submitted to RAL Tier-1 and the output is stored on tape. Batch reprocessing is done at multiple MICE enabled Grid sites and output files are shipped to central tape or disk storage at RAL using a custom File Transfer Controller.

1. Introduction
The international Muon Ionisation Cooling Experiment (MICE) [1,2], which is being constructed at the ISIS synchrotron at the Rutherford Appleton Laboratory (RAL), UK, is designed to demonstrate the principle of muon ionisation cooling for the first time. In the MICE Cooling Channel, muons will pass through a series of absorbers (e.g. liquid Hydrogen) and RF accelerating cavities with the intent that their transverse momentum relative to the beam axis will be reduced, while their momentum along the beam axis is maintained. For a significant number of muons – i.e. a beam – this represents a reduction of the beam emittance (i.e. “cooling”), however the precision of the emittance measurement required to demonstrate the ionization cooling effect dictates that in MICE the particles are tracked individually through the Cooling Channel (see e.g. [3]). MICE is therefore equipped with a set of detectors (time-of-flight counters, and Cherenkov/KL/EMR detectors) [4] to discriminate between the
muons of interest and the background of other particles, and a pair of tracking spectrometers [5] that measure each muon’s trajectory before and after the Cooling Channel. Signals from these detectors are captured by the MICE DAQ system [6] (developed from the DATE package from the ALICE experiment) and each run is written to disk as a series of 250 MB chunks. After that run has finished, a File Compactor process collates the data chunks for that run, along with some logging and online monitoring output, into a single tarball, including a list of files and their md5 checksums to allow integrity verification further down the processing chain. These tarballs (known as the raw data) are written to a separate disk server to allow the DAQ to resume data-taking with the next run as soon as possible.

2. The Raw Data Mover
The principal role of the Raw Data Mover is to make a safe copy of the raw data, by uploading the data tarballs to the Castor Storage Element (SE) at the GridPP Tier-1 facility at RAL [7]. Castor maintains two tape copies of the data for added security.

The Raw Data Mover has two steps: an initial temporary copy is made on to intermediate storage, and then uploaded to the Grid from there. The first step is intended to avoid bottlenecks on the DAQ disk access; as the DAQ writing data to disk must take priority during a run.

At the end of each run, data for that run are stored in a tarball prepared by the DAQ and placed in a directory, which is actively watched by the Data Mover. The mover process starts the initial copy only when the tarball is ready (which is indicated by a presence of a relevant run dependent semaphore) and only if no other DAQ writing activities are taking place, which is controlled by a global lock file. After the initial copy to intermediate disk storage is made and the file integrity is verified the main part of the Data Mover is triggered and the Castor uploading process begins.

The Grid uploader performs a check-summed transfer to Castor, registers the file with the LFC and stores the basic files attributes in the MICE Metadata Catalogue. The Raw Data Mover workflow is shown in figure 1.

![Figure 1. Raw Data Mover workflow.](image-url)
Both parts of the mover program maintain their persistence state by means of a series of semaphores, which if present indicate the point in a workflow the system is in. When restarted, the programs try to continue their tasks from the point they were in before. In some cases manual intervention is required, in particular when a file already exists on tape on restart and is not registered on the LFC. In this case the mover does not overwrite the file for safety reasons.

3. MAUS Reconstruction

MAUS – MICE Analysis and User Software [8] – is the reconstruction and simulation framework for MICE. The software is designed as a self-contained toolkit with virtually no external dependencies so it is highly portable. This has made it easy to make it available on the Grid – only small changes related to configuration options had to be added.

We have two slightly different use cases for MAUS on the Grid, which involve reconstructed data distribution. First is the off-line reconstruction of the data immediately after it has been taken. This process is triggered by raw data file upload to Castor and its registration in the MICE Metadata Catalogue. It is essential that this step be performed without unnecessary delay so we use a dedicated, high priority Grid queue at the RAL Tier-1 computing facility. For data reprocessing however, we would need more resources so we decided to use multiple MICE-enabled Grid sites in this case. While running at RAL we have an opportunity to store files there by simply copying a file from a Worker Node (WN) since that Castor instance is a close SE for Tier-1. When running remotely we have to find a reliable way to transfer the output file from a site’s local SE to the Tier-1 Castor so that it can be stored on tape. We opted for a solution which allows us to decouple file transfers from the actual reconstruction job tasks. This led to the design and implementation of the File Transfer Controller.

3.1 File Transfer Controller

Technically speaking the File Transfer Controller is a secure file transfer client. It is meant to be called by a Grid job on completion. The job just registers a file transfer request and finishes – the rest is done asynchronously by the Controller. Not only does it eliminate a possible point of failure when transferring a file but it also allows us to use a dedicated robot certificate to store files on Castor.

3.1.1 Implementation. The system is implemented as a secure Axis Web Service deployed on Tomcat. File transfer requests are internally stored in a MySQL database. A simple cron job takes care of periodic file transfer submissions, checking transfer status and possible resubmissions. On success a file is registered with the LFC and its properties are added to the MICE Metadata Catalogue. The back-end uses gLite File Transfer Service (FTS). The reconstruction workflow is shown in figure 2.

![Figure 2. MAUS reconstruction workflow.](image-url)
3.1.2 **Performance and testing.** We have successfully used the Controller during summer and autumn 2013 data taking. After initial testing with an older version of MAUS we successfully completed reprocessing of the entire MICE data sample to date (about 5000 files/jobs) using the latest version of the reconstruction software. From the number of jobs quoted above it is clear that the Controller will not have to cope with a high load so a Web Service implementation is suitable for this task.

4. **Conclusions**

We have implemented a Data Mover system that transports MICE data reliably to Grid storage whilst minimizing clashes with the DAQ system, and a framework that performs distributed reconstruction and copies the data back to a central tape store.

**References**

[1] M. Bogomilov *et al.*, “The MICE Muon Beam on ISIS and the Beam-line Instrumentation of the Muon Ionization Cooling Experiment” *Journal of Instrumentation* 7 P05009 (2012)
[2] G. Gregoire *et al.*, “MICE and International Muon Ionization Cooling Experiment Technical Reference Document” [http://www.mice.iit.edu/trd/MICE_Tech_ref.html](http://www.mice.iit.edu/trd/MICE_Tech_ref.html) (October 2005)
[3] P. Hanlet: “State Machine Operation of the MICE Cooling Channel” #447 CHEP2013
[4] R. Bertoni *et al.*, *Nucl. Instrum. Methods A* 615(1), 14 (2010)
[5] M. Ellis *et al.*, *Nucl. Instrum. Methods A* 659(1), 136 (2011)
[6] Y. Karadzhov: “MICE Experiment Data Acquisition System” #307 CHEP2013
[7] D. Britton *et al.* “GridPP: the UK Grid for particle physics” *Phil. Trans. R. Soc. A* June 28, 2009 367 pp 2447-2457
[8] C. D. Tunnell, C. T. Rogers, MAUS: MICE Analysis User Software, IPAC (2011)