

A data handling system for modern and future Fermilab experiments

R A Illingworth

Fermi National Accelerator Laboratory, PO Box 500, Batavia, Illinois 60510, USA

E-mail: illingwo@fnal.gov

Abstract. Current and future Fermilab experiments such as Minerva, NOvA, and MicroBoone are now using an improved version of the Fermilab SAM data handling system. SAM was originally used by the CDF and D0 experiments for Run II of the Fermilab Tevatron to provide file metadata and location cataloguing, uploading of new files to tape storage, dataset management, file transfers between global processing sites, and processing history tracking. However SAM was heavily tailored to the Run II environment and required complex and hard to deploy client software, which made it hard to adapt to new experiments. The Fermilab Computing Sector has progressively updated SAM to use modern, standardized, technologies in order to more easily deploy it for current and upcoming Fermilab experiments, and to support the data preservation efforts of the Run II experiments.

1. Introduction

Fermilab Scientific Computing Division is responsible for supporting the computing requirements of multiple experiments which are either currently running or will start to do so in the next few years. The data handling needs of these experiments range from tens of TB per year up to potentially many PB per year. Most of these experiments are significantly smaller in personnel than the large colliding beam experiments that used to run at Fermilab and so cannot provide dedicated experts to manage their data needs. In order to minimize the support load, a common solution is required for all these experiments. After evaluating various options, including writing a new data handling system, or adopting an existing one from either within High Energy Physics or outside of it, the decision was taken to adopt the existing SAM system used by the D0 and CDF experiments at Run II of the Fermilab Tevatron.

2. The SAM data management system

SAM (Sequential Access via Metadata) was originally begun as a data handling system for Run II of the Fermilab Tevatron[1, 2]. It has been used very successfully by the D0 and CDF experiments. For use with current and future experiments the architecture is now being modernized in order to build on the extensive past experience from over a decade of operations, while integrating modern technologies, and provide a generic solution capable of being used by multiple experiments. The original version required a heavyweight client implementation which made integration with experiment frameworks difficult. This has been replaced with lightweight HTTP REST (REpresentational State Transfer)[3] based interfaces that are much simpler to use.

SAM is designed as a highly automated, “hands-off”, data management system that does not require routine administrative intervention. Data staging from tape, and from storage element to storage element is automatic, with no requirement to manually select datasets to be made available at a particular storage element. The philosophy is to bring the data to the jobs, rather than expect the jobs to run where the data is already available.

3. Modernizing the SAM architecture

SAM was designed as a stand-alone system intended to handle the entire life-cycle of HEP data including tight coupling with the storage system and batch system. Since then a variety of standardized services have become available as part of the Grid infrastructure. The requirement is now for a lighter-weight system that integrates with the rest of the Grid.

For this purpose the SAM architecture is being reworked to remove little used or unnecessary features, and to provide a simpler and easier to use interface to the system. Originally CORBA[4] was used to provide a Remote Procedure Call interface between SAM components. However, CORBA is a complex distributed object-oriented system and interfacing with it required sizable client libraries which made integration with experiment frameworks difficult. The CORBA interfaces have been replaced with an HTTP interface based on REST principles. This is simpler to use, and as HTTP client implementations are ubiquitous, easier to integrate with experiment frameworks or any other component that needs to communicate with the data handling system. Additionally it allows use of standard Grid PKI authentication methods. The use of simpler interfaces implemented using a familiar protocol enables a loose coupling with Grid components such as job submission and storage and allows the removal from SAM of features – for example, disk pooling – which are provided by other services.

4. Integration with the FIFE project

SAM is now part of an integrated system – the Fermilab Fabric for Frontier Experiments (FIFE)[5] – which provides common tools and systems for scientific data processing. SAM interacts with components such as the *FIFE-Jobsub* glideinWMS based job submission system[6] and the *art* framework[7] using HTTP REST interfaces. This provides a loose coupling, so that it would be possible to interoperate with other services if required.

5. Metadata catalogue and datasets

SAM contains a metadata catalogue which is used to store information about experiment files. The metadata fields mostly consist of key-value pairs. Experiments are not restricted to a predefined set of metadata fields, but can freely define their own fields. The metadata also stores provenance information for a file, specifically the parent files from which it was derived and the application and version that was used to create it. Table 1 shows an example of the metadata from the NOvA experiment.

There is a query language for retrieving sets of files by their metadata values which has been enhanced in the new version of SAM to take advantage of experience gained with the original implementation. This allows simple queries like "run_number 13501 and file_format raw", which returns all raw data files from run 13501, and also more complex queries such as "run_number 13501 and file_format raw and not isparentof:(application reconstruction and version S12.02.14)", which returns all raw files from run 13501 which do not have a derived file reconstructed with the specified version of the software.

Flexible datasets can be created as stored queries. These datasets are evaluated dynamically; if more files that match the query criteria have been added since the dataset was created they will be included when it is used. End users can define their own datasets and are not restricted to a predefined set created by an administrator.

Table 1. Example metadata from NOVA experiment. This file is a reconstructed root file which was derived from a single input raw file.

File Name	reco_r00013501_s00_t00_numi_S12.02.14.root
File Id	4079577
File Format	root
File Size	18931848
Crc	4220712658 (adler 32 crc type)
Content Status	good
Data Tier	reconstructed
Data Stream	0
NOVA.DetectorID	ndos
NOVA.HornConfig	LE010z185
NOVA.HornPolarity	FHC
NOVA.Label	preMeta
NOVA.Special	none
NOVA.SubVersion	1
Online.RunEndTime	1329248111
Online.RunNumber	13501
Online.RunStartTime	1329244506
Online.SubRunEndTime	1329248111
Online.SubRunStartTime	1329244506
Online.Subrun	0
Reconstructed.base.release	S12.02.14
Runs	13501 (commissioning)
Parents	ndos_r00013501_s00_t00.raw

At job submission time the user only needs to provide the dataset name; they do not need to know anything about the file locations. The SAM portion of a processing task is referred to as a *project*; a project is responsible for delivering all the files in the input dataset. A single SAM project can be associated with more than one batch job. In the case where there are multiple jobs in a project the available files are distributed dynamically across all the jobs, so that each file is given to one job only.

6. Storage integration

SAM has a location catalogue which stores information about where the data files can be found. The system is designed to work with a variety of storage systems including SRM or dCache. The file locations stored in the location catalogue are mapped into concrete access URLs which are handed out to worker jobs. Individual jobs do not specify any explicit list of input files to process and have no direct control over the order that the files are delivered in. Instead they request a new file and are handed a URL with which to access it.

The job is responsible for the “last-mile” access[8] of the data files from the storage system to the worker node via the provided URL; consequently SAM itself is entirely neutral as to the access method used.

Because SAM works with datasets rather than individual files it can command prestaging of files from tape or transfers from storage system to storage system before the end user job needs to access the file, enabling more efficient file access than a purely access driven system while not requiring any manual staging of datasets.

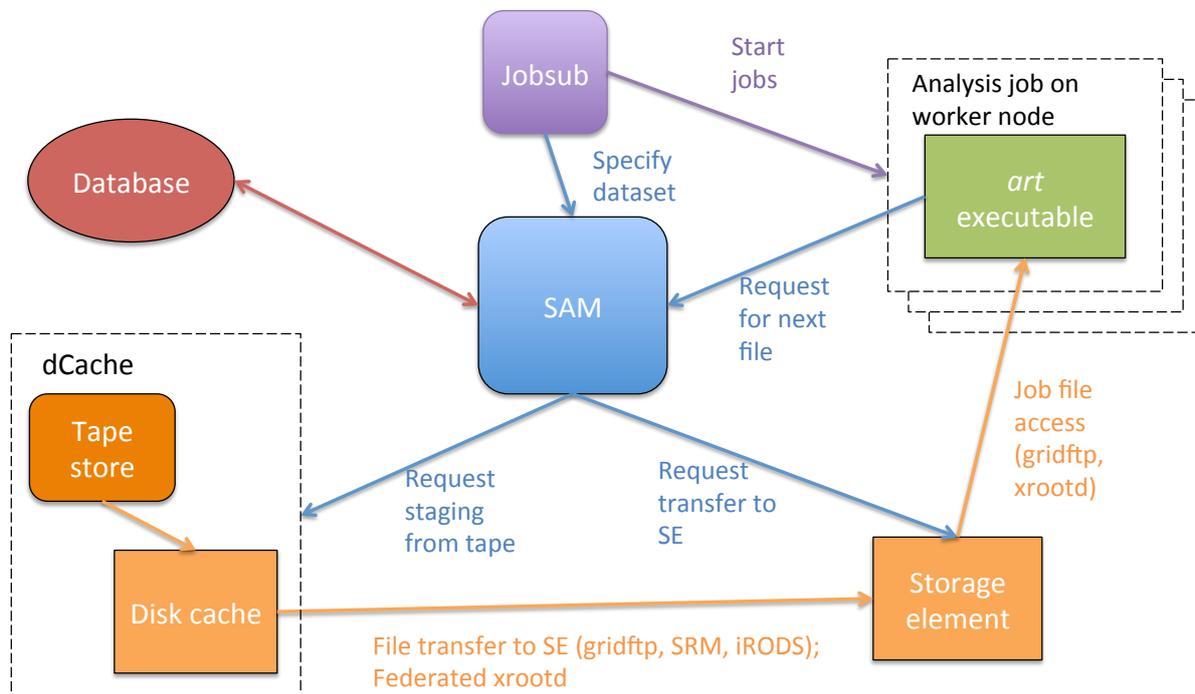


Figure 1. Running a job using SAM for data input

7. Running jobs

The data flow when running a data processing task is shown in figure 1. The task starts with the job submission system notifying SAM that a particular dataset is required by starting a project for it. If staging of the files is required it can begin at this point. The jobs are submitted to a computing element and once each job has begun execution it attaches itself to the project and begins requesting files. When they have finished with one file they notify the system and are given another to process, and this continues until all files have been provided.

7.1. Monitoring and tracking

All SAM file activity is logged in a database. This keeps a complete record of which files were given to which jobs, and which files and jobs were reported as having completed successfully. An example of the monitoring available is shown in figure 2. This particular SAM project shows the effects of the dynamic data distribution; although the individual jobs started at different times they all finished at about the same time as files were distributed across all the jobs until no more remained. Additionally, the bottom job in the graph was running on a faster worker node, and so processed each file more quickly. Rather than simply finishing earlier it was assigned more work.

7.2. Recovery of processing failures

The tracking of all file activity allows for simple recovery for files that were not correctly processed. This can happen for a variety of reasons; the storage system may not have been able to provide some of the dataset; the job may not have completed processing due to preemption or crashes; or output may have been lost due to some failure of the copy back procedure. SAM is capable of automatically creating a recovery dataset for any project that consists of all the

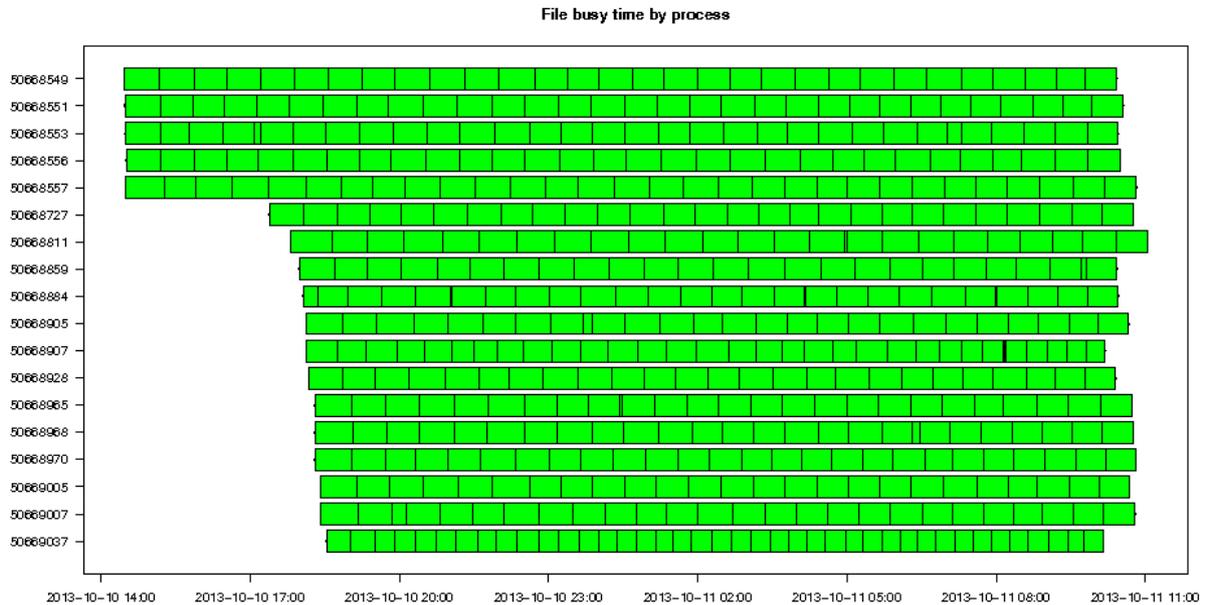


Figure 2. SAM project monitoring. The large bars show the time an individual job spent processing a file; the gaps between were time spent waiting for file deliveries.

files that were not successfully processed in the first pass.

8. Deployment

The SAM system is in use by the Minos, Minerva, and NO ν A experiments (figure 3). All of these began with the original Tevatron Run II version and have since migrated to use the updated version. It is currently in the process of being deployed for the μ Boone, DarkSide, and LBNE experiments (the latter will not take data for some years, but needs to catalogue and access Monte-Carlo files), and is planned to be used at the Muon g-2 experiment.

9. Long term data preservation for the CDF and D0 experiments

As previously described, SAM is the existing data handling solution for the Tevatron experiments CDF and D0. Both experiments intend to migrate away from the legacy SAM system and to the same updated version as is being used by the current and future Fermilab experiments. This approach allows the experiments to retain their existing metadata databases while not requiring long term support of the legacy SAM codebase.

10. Conclusion

The modern and future Fermilab experiments are making use of an updated version of the SAM data handling system as previously implemented for Run II of the Fermilab Tevatron. The updated implementation builds on experience from the Tevatron, as well as taking advantage of modern computing technologies. The design is flexible, experiment agnostic, and highly automated, making it suitable for a wide range of high energy physics experiments.

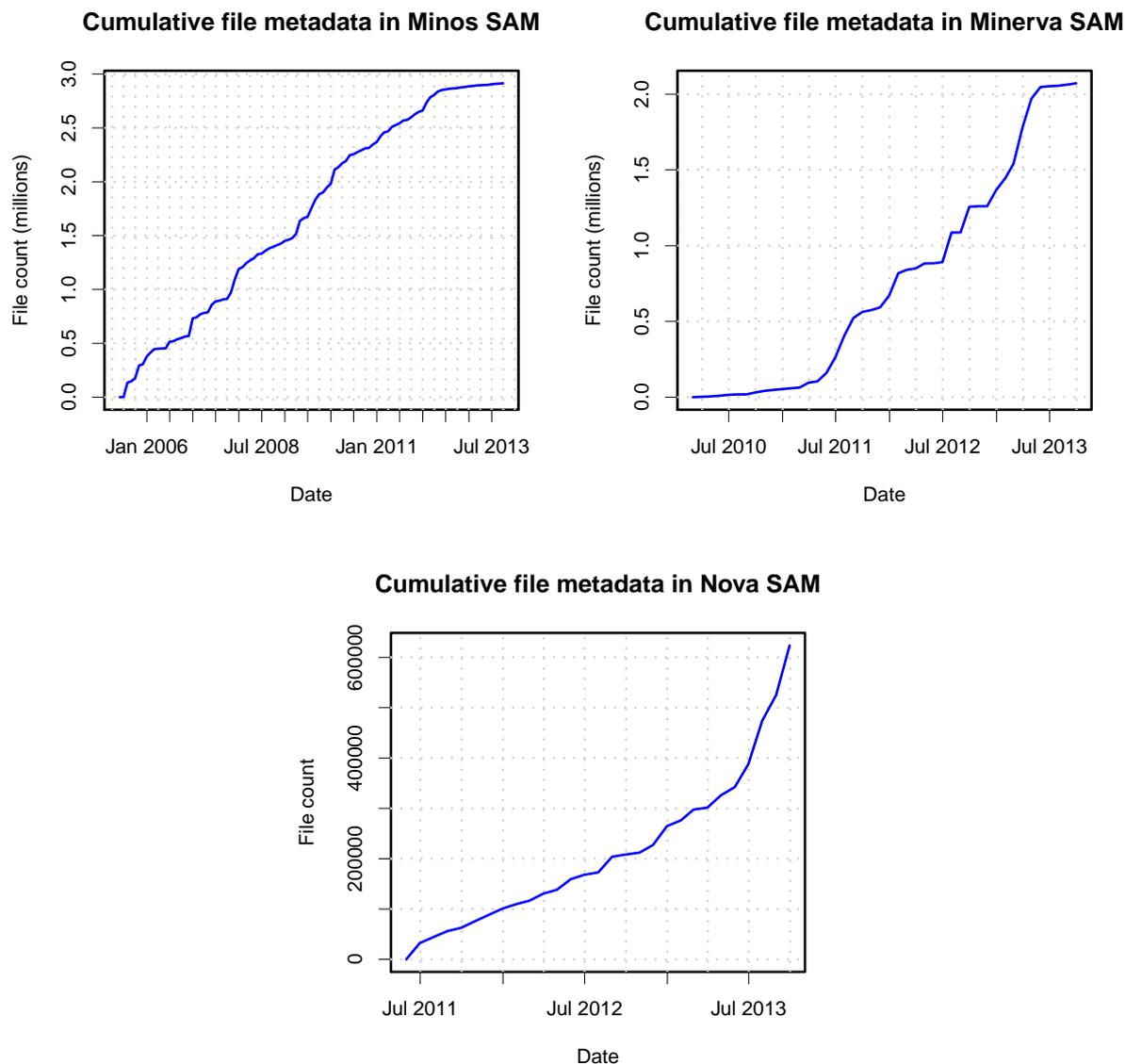


Figure 3. Number of files catalogued in the SAM databases for the Minos, Minerva and NOvA experiments

Acknowledgments

The Fermi National Accelerator Laboratory is operated by Fermi Research Alliance, LLC under Contract No. De-AC02-07CH11359 with the United States Department of Energy. This contribution is FERMILAB-CONF-13-482-CD.

References

- [1] Terekhov I 2003 *Nucl. Instr. Meth. Phys. Res. A* **502** 402–406
- [2] Stonjek S, Baranovski A, Kreymer A, Lueking L, Ratnikov F *et al.* 2005 Deployment of SAM for the CDF Experiment *Proceedings of Computing in High Energy Physics 2004* pp 1052–1054 CERN-2005-002
- [3] Fielding R T and Taylor R N 2002 *ACM Trans. Internet Technol.* **2** 115–150 ISSN 1533-5399 URL <http://doi.acm.org/10.1145/514183.514185>

- [4] Common Object Request Broker Architecture (CORBA) URL <http://www.omg.org/spec/CORBA/>
- [5] Kirby M 2013 The Fabric for Frontier Experiments project at Fermilab *Proc. of the 20th Int. Conf. on Computing in High Energy and Nuclear Physics (CHEP2013)*
- [6] Box D 2013 FIFE-Jobsub: A grid submission system for intensity frontier experiments at Fermilab *Proc. of the 20th Int. Conf. on Computing in High Energy and Nuclear Physics (CHEP2013)*
- [7] Green C, Kowalkowski J, Paterno M, Fischler M, Garren L and Lu Q 2012 *Journal of Physics: Conference Series* **396** 022020 URL <http://stacks.iop.org/1742-6596/396/i=2/a=022020>
- [8] Lyon A and Mengel M 2012 The “last mile” of data handling - Fermilab’s IFDH tools *Proc. of the 20th Int. Conf. on Computing in High Energy and Nuclear Physics (CHEP2013)*