Housing metadata for the common physicist using a relational database

K. Merritt, R. St. Denis, R. Kennedy, A. Kreymer, L. Lueking, F. Ratnikov, S. White, A. Sill, M. Leslie, S. Stonjek, et al.

HOUSING METADATA FOR THE COMMON PHYSICIST USING A RELATIONAL DATABASE

R. St.Denis, M. Burgon-Lyon, A.Doyle, S. Hanlon, P.Millar *Glasgow University,* A. Baranovski, G. Garzoglio, R. Herber, R. Illingworth, R. Kennedy, A. Kreymer, A. Kumar, L. Lueking, W. Merritt, L. Loebel-Carpenter, A. Lyon, I. Terekhov, J. Trumbo, S. Veseli, S. White, P. Vokac, M. Zimmler, *FNAL,Batavia IL USA* S. Albrand, J. Fulachier *LPSC, Grenoble,* T. Barrass *Bristol*, S. Burke, O. Synge *RAL,* S. Belforte, *INFN, Trieste*, U. Kerzel, *Karlsruhe University,* M. Leslie, V. Bartsch, S. Stonjek, C. Cioffi *Oxford University*, A. Forti Mancheter, F. Ratnikov, *Rutgers University,* A.Sill, *Texas Tech University* P. Kunszt, G. McCance, K. Nienartowicz, R. Rocha *CERN*

Abstract

SAM was developed as a data handling system for Run II at Fermilab. SAM is a collection of services, each described by metadata. The metadata are modelled on relational database, and implemented in ORACLE. SAM, originally deployed in production for the D0 Run II experiment, has now been also deployed at CDF and is being commissioned at MINOS. This illustrates that the metadata decomposition of its services has a broader applicability than just one experiment. A joint working group on metadata with representatives from ATLAS, BaBar, CDF, CMS, D0, and LHCB in cooperation with EGEE has examined this metadata decomposition in the light of general HEP user requirements. Greater understanding of the required services of a performant data handling system has emerged from Run II experience. This experience is being merged with the understanding being developed in the course of LHC experience with data challenges and user case discussions. We describe the SAM schema and the commonalities of function and service support between this schema and proposals for the LHC experiments. We describe the support structure required for SAM schema updates, the use of development, integration, and production instances. We are also looking at the LHC proposals for the evolution of schema using keyword-value pairs that are then transformed into a normalized, performant database schema.

INTRODUCTION

There are a number of goals we wish to achieve in this paper. First, we would like to describe the many facets of metadata. We will then examine the metadata use cases for HEP, and turn to describing SAM[1], the data handling system for CDF, D0 and MINOS, and the SAM schema. Commonality of this implementation with LHC requirements will be established through demonstration that SAM satisfies the HEPCAL use cases. The benefits of exposing the metadata services to a number of different experiments will be described. This will include outlining the standards required to allow for the database that houses the metadata to evolve in a controlled fashion while permitting 24x7 operation.

USE CASES

We have made a survey of the use cases that have been accumulated for HEPCAL [2], CDF[3], BABAR[4] and ATLAS[5]. We conclude that there are three main categories: Analysis, Job Handling and Dataset Handling. For analysis, at the highest level, a physicist sitting at a terminal wishes to take some data or generate it from a Monte Carlo, pass it through a program and produce some output. The use cases fall into two flavours: user analysis and production analysis. The former is a chaotic, uncontrolled community and the latter is an organized, planned and structured group operating on behalf of the larger (less organized) community. Both wish to run physics simulation or select a subset of data, and run some algorithm over the input dataset.

The next class of use cases, job handling, first requires the ability to estimate system resource cost. Next, there is a desire to submit a job to the Grid, perhaps with predefined metadata used to drive the analysis program. When the program is running, there is a need to monitor it as the user, either of the chaotic or production persuasion, wishes to know that the job is executing. Once the job completes, it must be possible to retrieve the data. Upon analysis of the results, there is then the desire to repeat the job on the same set of events, perhaps with different predefined metadata, or to recover from errors in either the infrastructure of data and job handling or the user analysis.

The final category, dataset handling, requires that it be possible to read metadata for a dataset, resolve the physical data and be able to use this to specify a new dataset. It must be possible to predefine metadata for the output dataset and update or add new metadata for datasets. The dataset will need to be downloaded to a local disk in order that an analysis program is able to access the dataset and there must be a capacity to add experiment specific metadata.

In order to interrogate the meaning of a dataset, it must be possible to read all the visible metadata. Upon

verification and acceptance of a dataset as being valid, it must be possible to publish the dataset and the metadata for public consumption. This allows analysis groups to work with the same sets of events and focus on physics differences in the analysis. Datasets may be produced in ways that are not suitable for physical storage, so it must be possible to transform datasets by merging, filtering, splitting or some other similar function. Finally, it should be possible to search for the datasets whose metadata match the needs of a user.

SAM

Having described the various use cases that must be solved by a metadata implementation we wish to consider SAM[1]. The SAM paradigm is that a **project** runs on a **station** and requires the delivery of a **dataset** to one or more **consumer process** associated with the station. The consumers perform a transformation on the dataset and output files to be stored, together with metadata for these files. File delivery is stateful and a permanent record of data handling is kept for the project. For SAM, a dataset is a group of files containing events. These events may be from a single run or a variety of runs where a run is a period of stable conditions for the experiment. A dataset is associated with all files containing events that have satisfied some trigger conditions. A station is a collection of services that interact with the consumer process, satisfying the demand of the consumer process to obtain data.

SAM is implemented using a relational database with more than 100 tables. D0, CDF and MINOS use this single schema. Being relational, it matches metadata from a variety of services described below. It is monolithic and this takes advantage of the efficiency of database constraints. A single set of SAM services can easily deliver more than 360 filenames/minute to a consumer with minor impact on modern servers. The schema is flexible and can be updated in a controlled fashion.

In order to provide retrieval of data, SAM manages file storage (replica catalogues). Data files stored on tape systems at Fermilab or indeed on disk as well, and at any location around the world are tracked by SAM and SAM services determine the most efficient way to route files to the user.

To allow the user to be able to read metadata and resolve the physical data SAM manages file metadata cataloguing. As illustrated in Figure 1, data files are at the heart of SAM and are delivered by SAM to the user's local disk. The user can also store output on SAM and hence retrieve it later for further analysis. There are fixed metadata, such as file name, size, CRC, the group for whom the file was produced, the data tier (raw, reconstructed, secondary etc.), the application that produced the file, the locations of the file, the detector (CDF, D0, MINOS), the runs contained within the file, the event information (D0 only) the projects that have been run on the file and analysed it, the luminosity and the trigger and data stream from which the file was produced.

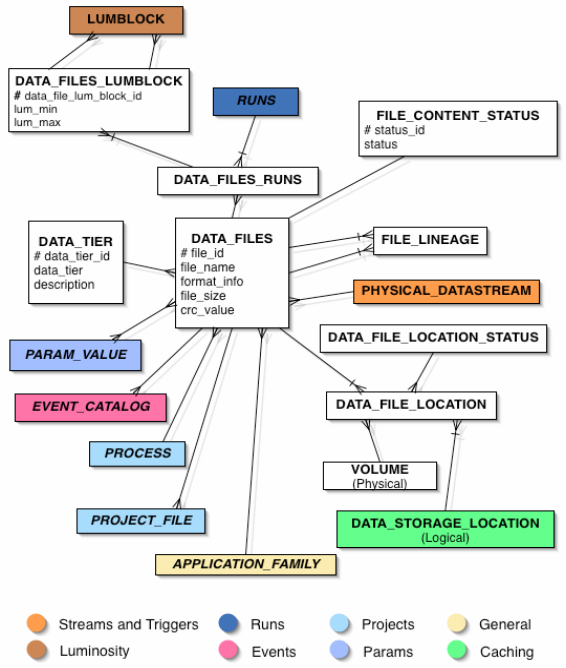

Figure 1: The Heart of SAM: the File metadata. It is surrounded by blocks representing the various services whose metadata are also kept in SAM.

An essential component of the file metadata is a connection to a set of "free" metadata. This feature is a common feature found in ATLAS and LHCB as well. While fixed metadata allow easy and performant querying, free metadata are useful for application specific items and satisfy the requirement for experiment-specific metadata. As illustrated in Figure 2, the keyword-values pairs are grouped into categories with types within the categories being the keywords. Values are then stored separately and may be shared amongst parameters for efficient usage of space. For example, if one wishes to store parameters for the Pythia Monte Carlo program, one can put the parameter types into this one category. The queries can be more difficult and one has to be careful of the access to theses parameters. The parameters also have a connection to a "request detail" portion of the database that describes requests for processing. This can be used then to define the parameters, and hence file content metadata, ahead of time, thus satisfying another use case.

Up to this point, the description of the metadata for SAM involves two flavours: values associated with describing the physical characteristics of files and values associated with describing the physics in the files. To satisfy the use case that it be possible to read all the metadata associated with a dataset, SAM manages definitions of datasets based on these physical and physics metadata. Definitions of datasets are stored in the SAM Database by group and user. These definitions are resolved to a list of files whose metadata satisfy the

requirements of the definition when a user chooses to either run a project. An example of the kind of definition used is “data_type physics and run_number 78904”. The utility for this satisfies the use case involving matching a dataset to the user requirements as well as allowing the user to examine the contents of the dataset.

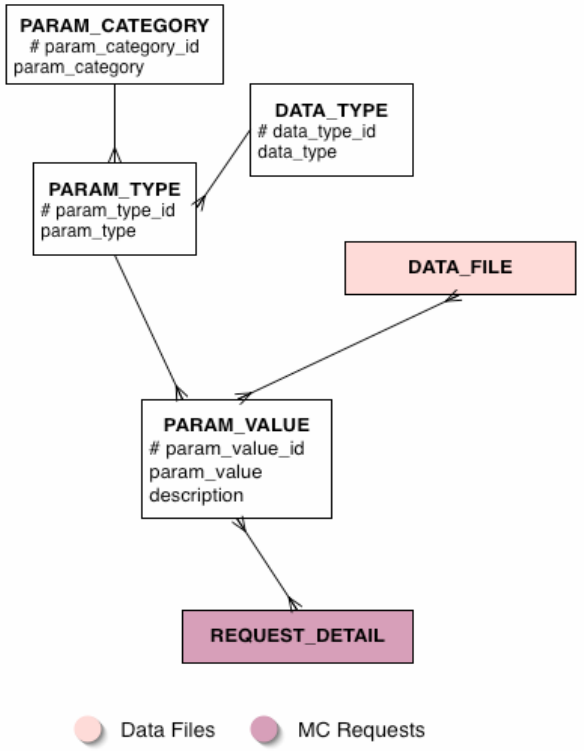

Figure 2: Flexible storage in the parameters of SAM.

When a user has defined a dataset and decides to run a project on this dataset, SAM manages the analysis bookkeeping for that project. This portion of the schema is shown in Figure 3. SAM remembers the files that were used, the ones that were processed successfully, which application was run, when it was run and on what machine it was run. Hence this allows for the use case of repeating the analysis and recovering from errors. This information is stored in the project metadata. The heart of this is the analysis project that is connected to a process, which in turn points to an application and its version. These metadata are harvested automatically for users running the experimental framework and stored with the files generated. The instantiation of a dataset definition for a project is kept in the project snapshot. SAM tracks all snapshots made from a definition. A project can have multiple parallel consumer processes, which will as an ensemble receive all files in the dataset.

The use cases state that the user must have files delivered to the local disk for consumption. SAM manages file delivery by dataset. Users at FNAL and remote sites retrieve files out of file storage. SAM handles caching or can interface to other cache systems [6]. The user need not care about where the files are located. File delivery is handled by the SAM Station. The station contains a **project master**, a service which coordinates the delivery of files from a **storage element**. The station uses CORBA to communicate with the consumer process demanding the files, with the SAM database via a database server [7] to determine the locations of the files, and with services that optimise the delivery of files according to the rules established when the station is configured. The station keeps track of the files that it has requested and may either manage a cache for files to be given to the consumer process or it may interface to an SRM and dispatch the URL’s required to obtain the files from the SRM. The station keeps a record of the cache or the files most recently requested by consumer processes that have used its services.

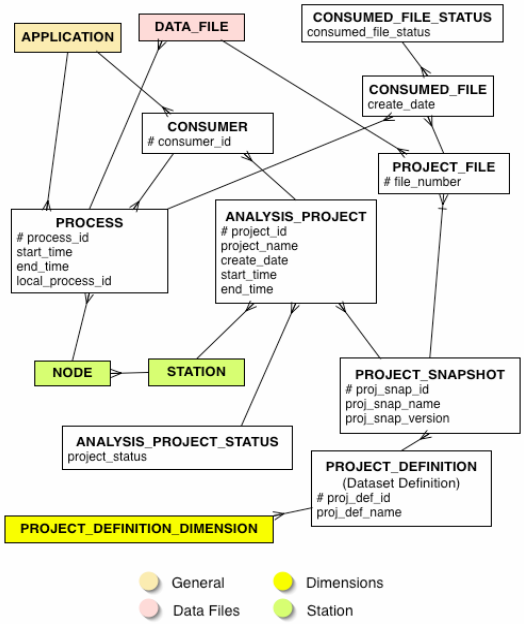

Figure 3:The project metadata.

USE OF SAM IN CDF AND MINOS

SAM has been in use by the D0 experiment for several years with extensive use in the last two to three years as the luminosity of the Tevatron has increased. D0 has over 40 active SAM sites. The performance of SAM in Run II is described elsewhere in these proceedings [8] as are the methods for monitoring SAM [9]. The CDF experiment has been preparing to use SAM for several years and extensively used it for physics analysis at sites outside of Fermilab for over a year. CDF has over 25 active sites. The deployment [10], testing [11] and global use of SAM to exploit dedicated CDF resources[12] are described elsewhere in these proceedings. It is on track to use SAM for Fermilab services as well. The MINOS experiment is committed to the usage of SAM with two sites available for first testing. Finally, CMS is evaluating SAM.

In order to support robust, long-term production use and to facilitate migration to other experiments, both of which require significant schema changes, it was necessary to have strict standards for deployment. These standards ensure database integrity by supporting three

instances of the SAM schema. A **development** instance is used to test the newest schema with artificial or special data. Once these tests are ready to be deployed, an **integration** instance is prepared by exporting the contents of production. A dress rehearsal is performed, running the scripts that are to be used for SAM and then running a test harness [13] to exercise the SAM services. Finally the new schema changes are placed into the **production** database.

CDF participation in SAM has provided an opportunity for revisiting of the original D0 design enriched by inclusion of the D0 experience derived from use in different phases in the experimental life cycle, such as Monte Carlo production prior to Run II, commissioning of the Run II detector and the period of stable data collection. An entirely new user community provided the trigger for a second-generation design, the need for which was recognized by the original users. In the course of this design work, the services became better defined and the boundaries explicitly drawn.

There were a number of important features of the change in the schema from the first generation of SAM. Amongst these we will give three illustrative examples. The first is the removal of the requirement of a process id for every file. While the SAM paradigm assumes all files except for raw data come from SAM projects, there are some files that will be generated outside the SAM framework. These include files from theorists with interesting processes to be shared amongst experiments and data that may have been analysed outside of the strict SAM framework but are of interested and have validity. Thus the paradigm had to have a modification. A second example is found in the luminosity bookkeeping. This is different for CDF and D0. D0 keeps their luminosity in contiguous blocks and file inheritance is used to reconstruct luminosity from derived files. CDF makes a direct link of its derived files to the luminosity data. Providing a list of blocks with pointers that satisfies both needs, which, when resolved, take on meaning within the context of the individual experiment. Finally, a change of the location of the business rules from the database to the API allowed for constraints on a variety of metadata fields to rely on the file type. For example, a file type of physics Generic must have a Data Tier of "unofficial reconstruction" if the experiment is D0.

CONCLUSIONS

In this paper we have shown that there are a variety of facets to metadata: workflow processing, physical file, file content, physics content, storage and processing. We have also demonstrated that SAM is a system of data handling and workflow services described by metadata modelled on a relational database. SAM implements the HEPCAL, CDF, BABAR and ATLAS requirements as summarized here. Migration of metadata schema with running experiments is inevitable but can be accomplished with the use of strict standards and testing. Greater understanding of the metadata architecture has been the benefit of having multiple experiments uses the same schema. In addition, natural demarcation of services, sharpening of the boundaries between services and understanding of requirements arising during different phases of the experimental life cycle have all been beneficial. Finally we have demonstrated that detailed schema and API implementations for data handling and job processing services can be shared amongst HEP experiments.

ACKNOWLEDGEMENTS

We would like to thank Fermilab Computing Division for its ongoing support of the SAMGrid project, and especially the CCF, CEPA, and Run II Departments. We would also like to thank everyone at D0 and CDF who has contributed to this project. This project is sponsored by DOE contract No. DE-AC02-76CH03000 and by GridPP.

REFERENCES

[1] http://projects.fnal.gov/samgrid.
[2] "Common Use Cases for a HEP Common Application Layer",HEPCAL. F. Carminati ,et.al, LHC-SC2-20-2002, https://edms.cern.ch/document/375586.
[3] B.T. Huffmann,et.al "The CDF/D0 GridPP Project" http://www.gridpp.ac.uk/datamanagement/metadata/SubGroups/UseCases/docs/cdf5858.ps.gz.
[4] GridPP BaBar group, "Analysis Grid Application – Use Case and Requirements Document" http://www.gridpp.ac.uk/eb/tender/BaBarJobs1.ps.
[5] "Catalog Services for ATLAS", D. Adams http://www.gridpp.ac.uk/datamanagement/metadata/SubGroups/UseCases/docs/ada_catalogs.pdf.
[6] R. Kennedy et al.,"SAMGrid Integration of SRM's", this conference proceedings.
[7] L. Loebel-Carpenter et al., "The SAMGrid Database Server Component: Its Upgraded Infrastructure and the Future Development Plan", this conference proceedings .
[8] A. Boehnlein, et al. "Run II Computing", this conference proceedings.
[9] S. Veseli, et al., "SAMGrid Monitoring and Information Service and its Integration with Mona Lisa", this conference proceedings.
[10] S. Stonjek, et al., "Deployment of SAM for the CDF Experiment", this conference proceedings.
[11] V.Bartsch, et al., "Testing the CDF Distributed Computing Framework", this conference proceedings.
[12] A. Sill, et al., "Globally Distributed User Analysis Computing at CDF", this conference proceedings.
[13] M. Leslie, et al., "Application of the SAMGrid Test Harness for Performance Evaluation and Tuning of a Distributed Cluster Implementation of Data Handling Services", this conference proceedings.