

SAM Overview and Operation at the D0 Experiment

Lauri Loebel-Carpenter, Lee Lueking, Carmenita Moore, Igor Terekhov, Julie Trumbo, Sinisa Veseli,
Matthew Vranicar, Stephen P. White, Victoria White

Fermilab, Batavia, IL 60510

July 25, 2001

Abstract

SAM is a network-distributed data management system developed at Fermilab for use with Run II data. It is being employed by the D0 Experiment to store, manage, deliver, and track processing of all data. We describe the design and features of the system including resource management and data transfer mechanisms. We show the operational experience D0 has accumulated to date including data acquisition, processing, and all levels of access and delivery. We present various configurations of the system and describe their use in the collaboration.

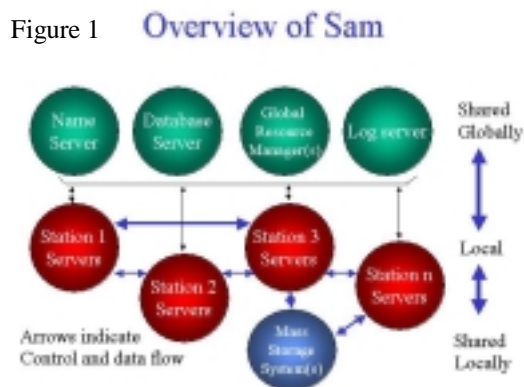
Keywords: Data Management, Data Access, GRID enabled system, Resource Management

1. Introduction

The SAM system was developed at Fermilab to accommodate the high volume data management operations needed for Run II Physics and to enable streamlined access and data mining of these large data sets. SAM stands for “Sequential Access to data via Metadata” where the “sequential” refers to sequential events stored within files which are in-turn stored sequentially on tapes within a Mass Storage System. However, SAM is largely devoted to transparently delivering and managing caches of data as it is needed on the various D0 computing platforms. SAM is designed with a distributed architecture, using CORBA as its underlying framework. This has enabled the system to scale to meet the data distribution needs of the D0 Collaboration which includes 60 institutions and over 500 Physicists located all over the globe. Metadata and configuration information for the entire system are currently maintained in a central Oracle database repository maintained at Fermilab, and as the system matures, it is planned to distribute this function in a way that will reduce latencies and make the overall operation more reliable. Data from the Dzero detector operation, and simulation data are likely to require more than a half Petabyte of permanent mass storage over the next two years. Mass storage systems located at Fermilab and collaborator sites worldwide are integrated into the system to fulfill this requirement.

2. Overview of the SAM System

SAM is based on a distributed network architecture and a detailed description is available elsewhere[1,2]. All components in the system use CORBA[3] services to communicate with each other through well defined CORBA interfaces. An overview of the system is shown in Figure 1. The way the system is illustrated here, there are three levels of server, 1) Shared Globally, 2) Local unshared, and 3) local shared. The globally shared services include the CORBA Naming Service, the database server, a global resource manager, and a log server. The name server allows all other components of the system to find each other by name. The database server has numerous methods which process transactions and retrieve information from the central database[4] as it is requested by clients. There are one or more resource managers deployed to control and more efficiently use resources like ATL tape mounts, tape drive usage, and network capacity. A log server is used to gather logging information from the entire system for monitoring and debugging purposes.



A component called the *station* is deployed on local processing platforms, and is unshared outside of the context of its set of CPU and disk cache resources. Stations, however, can communicate among themselves and data within one station's cache can be replicated to other stations on demand. Local groupings of stations, at a physical site eg. Fermilab, can share a locally available Mass Storage System. In the case of Fermilab, and some other Dzero collaborating sites the MSS is Enstore [5] and this sharing is controlled by mountpoints served by the Enstore system to the desired station platforms. Other MMS types, including HPSS, are used within the D0 collaboration and interfaces to these have also been provided.

In the operational system there are, in fact, many variations to this picture. The DB Servers are easily cloned to run in parallel to distribute the load and make the system more reliable. In the future a more sophisticated server might be built that includes multi-threading and request queuing, but this has not been needed so far. Resource manager servers are also deployed throughout the system as needed. There are resource management issues that are global in nature, like the movement of data throughout the interconnected web of stations, but most of the contentions are among local station groupings, like those that share a robotic tape library.

The most complex of all the SAM deployed pieces is the *station*. Overall, the station's responsibilities include 1) Storing and retrieving data files to and from available mass storage systems and other stations, 2) Managing data stored on cache disk over which it has exclusive control, 3) Launching *project managers* which oversee the processing of data requested by users (*consumers*) in the form of well defined projects. These functions are provided by the servers within the station, as shown in Figure 2. The *station manager* oversees the removal of files from the cache disk, and instructs *file stagers* to add new files. All processing projects are started through the station server, which starts *project managers*. Files are added to the system through the *File Storage Server* (FSS), which uses the Stagers to initiate transfers to the available MSS. More detailed accounts of the station's operation and resource management can be found elsewhere [6,7].

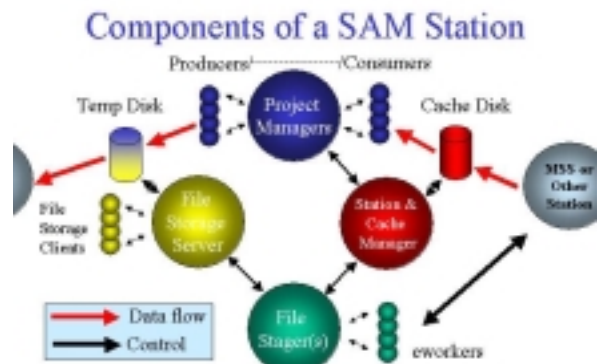


Figure 2

3. User and Administrative Interfaces

In order to use the system, user and administrative interfaces are provided to add data, access data, set configuration parameters, and monitor the system. Interfaces to the system are in the form of UNIX command line, Web GUI's, and a Python API. There is also a C++ interface provided for accessing data through a standard D0 framework package. Most of these interfaces are connected to the system through CORBA with the exception being a database browsing tool which offers a convenient method to build web-based queries.

Adding data to the system requires a way of describing the data, and a map which tells the system where the data should be stored. The description of each data file is required in order for the file to be accepted into the system. This description is a small python file that contains information like physics attributes, parentage, number of events, event range, processing application and version, date, and many other attributes describing the files origin. Data from the detector includes an event catalogue specifying event number, trigger information, and luminosity tags for each event in the file. Monte Carlo file descriptions include attributes such as physics processes, decay channels, detector noise, and number of minimum bias events. As data is processed from input file to output file, information is inherited by the children, and only new attributes are needed in the description files. Actual data storage into the system is performed using a simple "sam store" command with the single argument being the name of the description file. From the information contained therein, the system consults a map stored in the SAM database known as the auto-destination map. This map directs the data file to be stored in the correct MSS system in a predetermined location.

Once data is in the system it can be accessed at any station using tools that allow users to define datasets, or groups of files, and process them in projects. Datasets can be created by entering a set of constraints, based on the attributes entered with each data file, to create what is known as a *dataset definition*. This definition is then used to query the

database and determine a set of files that match the criteria. The result is called a *dataset*. Because the specific files in the dataset can change, as new data is entered into the system or lost due to attrition, the datasets are versioned to insure reproducibility. Once a dataset is established, it can be submitted to a station with a well defined *application* and all the files processed in what is known as an *application project*. There is a simple command line interface, “sam submit”, that is used to place the job under the control of the batch system on the particular machine(s) where the station is operating.

User and administrative interfaces to the system allow convenient methods to find data and manage and configure the system. User interfaces are provided to query for data and create datasets based on physics and on-line data-taking attributes. A simple web-based query tool provides general information about the number of files, size, and number of events for the user’s choice of constraints. A command line interface as well as a sophisticated dataset editor web tool are provided to create dataset definitions and datasets. Administrative interfaces include facilities for adding users and assigning them to groups, adding and configuring stations, creating valid data locations, setting volume or file status, adding valid application and version information and others.

4. Operational Experience in D0

The SAM system has been used for all of the Dzero data management beginning with Monte Carlo data in 1999, and extending through the current on-line data acquisition that began in March of 2001. As of July 2001 there were 330 registered users and 20 production stations. A station named “central-analysis” at Fermilab was the first Dzero production installation of SAM, and is the most heavily exercised user facility to date. It utilizes the resources of the 174 processor SGI Origin 2000 machine, called D0mino, with almost 30 TB of attached disk. The chart of Figure 3 shows the cache turnover for this station with incoming data roughly mirroring data being removed. The chart of Figure 4 shows the amount of data added to the system over the last year. Nearly all of the Monte Carlo data is produced at sites other than Fermilab and the files and metadata are sent to FNAL over the network. SAM stations have been deployed at all of these “remote” processing centers and files are routed through cache areas managed by the central-analysis station, to the D0robot for permanent storage. Included among these sites are NIKHEF at Amsterdam, IN2P3 at their computing center in Lyon, Charles University in Prague, Lancaster University in Lancaster U.K., and The University of Texas Arlington. Stations have been deployed for data access and testing at Michigan State University, Columbia University, and Imperial College in London.

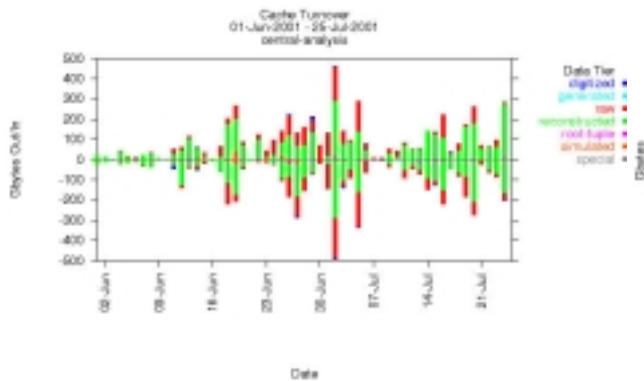


Figure 3. Cache turnover (GB) for the D0 central analysis station for June 1 to July 25, 2001.

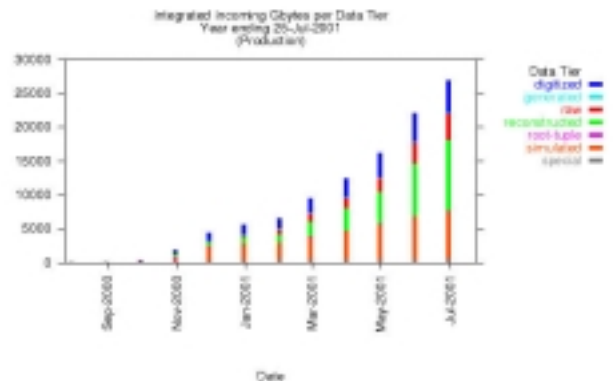


Figure 4. Integrated data added to the SAM system for the year preceding July 25, 2001.

A SAM FSS is deployed on the on-line system for storing detector data as it arrives from the Level 3 filter system. The trigger events are sorted into files and stored, with metadata and event information, into SAM. As of July over 4 TB of RAW data, more than 50M event triggers, have been stored to the system. Multiple copies of files are allowed to be stored to tape, and most of the online data has been stored in duplicate as a way of testing higher data rates and assuring no data is lost during the initial stages of the run. All of these stations employ straightforward configurations with central disk cach, however the requirements of the reconstruction farm demand a slightly more complex arrangement.

The reconstruction farm is a network-distributed system of processing nodes, each with its own cache. Each cache on each node is equipped with its own stager, and as files are processed the cache is refreshed with new data delivered to the node's disk. The staggers on each node have direct access to the D0 robot through an Enstore mount-point. A single station manager is responsible for managing this distributed cache, and analysis projects are submitted through it. Large processing projects are distributed among the many worker nodes and the staggers are instructed which files, of the many in the project, to stage. It is possible to create a situation in which a particular file has been staged to one node in the cluster, and is requested to be processed at another node. This occurs on occasions when nodes are disabled, the entire farm is stopped for maintenance, and other situations. To alleviate this scenario a feature is included in SAM that initiates an intra-station transfer and copies the file from the original node on which it was staged to the node on which it is needed.

Another variation of SAM station with a network-distributed set of processing nodes is ClueD0. This station comprises one linux file server and over 90 satellite worker nodes. The file server has a Gigabit Ethernet interface to the network, 640 GByte of physically attached disk, and it is given access to ENSTORE and other stations at Fermilab. The worker nodes are linux boxes used for desktops, located in reasonable proximity to the server and connected to the network with 100 Mbit Ethernet, but not given access to ENSTORE or other stations. These satellite nodes have various CPU and disk resources, but their systems are all managed in a consistent manner. The total cache disk deployed throughout the station exceeds 5TB. The station is configured so that the cache disks are physically distributed among all of the satellite nodes, but managed by a single station master. Each worker node has its own SAM stager process started when the node is booted. Analysis projects requesting files on any one of the worker nodes will be satisfied by one of the following actions: 1) if the file is already resident on the node it will be used, 2) if the file exists in the cache on another node in the cluster it will be copied to the local node, or 3) if the file is not in the cache of the ClueD0 station it will be brought in, through the file server node, from another station or through an ENSTORE transfer from the MSS. Jobs are submitted to the system through SAM using the PBS batch adapter.

5. Future Plans and Conclusion

Although the system has been functioning successfully for more than two years, it is still undergoing improvements and many changes are planned. More flexible station disk arrangements will be implemented soon that will allow multi-node stations to have more complex configurations. Additional cache routing features are planned, and the File Storage Server will be merged with the Station and Cache Server. A more de-centralized database model is being discussed that will make the system even more scalable and reliable. Additional job control and resource management features are planned. The SAM system is currently used for all Dzero data management and access for Run II data and experience to date indicates the system will support the needs of data acquisition, processing, and analysis. With a few planned improvements the system will scale to meet the delivery needs for the entire collaboration and we anticipate installing stations at each of our collaborating institutions worldwide.

We would like to thank the Fermilab Computing Division for its ongoing support of SAM, especially the ODS, D0CA, and ISD Departments. We would like to thank everyone at D0 who has contributed to this project, and the many important discussions we have had there. This work is sponsored by DOE contract No. DE-AC02-76CH03000.

References

- [1] The SAM team, L.Lueking(co-leader), V.White(co-leader),L.Loebel-Carpenter, C.Moore, H. Schellman,I.Terekhov, J.Trumbo, S.Veseli,M.Vranicar, S.White, C.Jozwiak. <http://d0db.fnal.gov/sam>
- [2] L.Lueking et.al., "The Data Access Layer for D0 Run II: Design and Features of SAM", CHEP 2000, March 2000, Padua Italy.
- [3] The OMG home page, <http://www.omg.com>, For C++ components, "ORBacus for C++ and Java" <http://www.ooc.com>. For Python components "Fnorb, a python ORB" <http://www.fnorb.org>.
- [4] Op. Cite., Lueking et. al., Chep 2000.
- [5] The ENSTORE home page <http://www-isd.fnal.gov/enstore>
- [6] Igor Terekhov, "Distributed Processing and Analysis of Physics Data in the D0 SAM System at Fermilab", Super Computing 2001, Denver Colorado, November 2001, to be published in the proceedings.
- [7] L. Loebel-Carpenter et.al., "Resource Management in SAM and the D0 Particle Physics Data Grid", CHEP 2001, September 2001, Beijing China.

