

## What is SAM?

SAM is a data handling system organized as a set of servers which work together, communicating via CORBA, to store and retrieve files and associated metadata, including a complete record of the processing which has used the files. Specifically, it is designed for the following tasks:

- 1) Provide storage utilities to add a file to a permanent storage location (generally, by interfacing to a mass storage system implemented on a robotic tape store).
- 2) Track locations and comprehensive metadata for each file in the system.
- 3) Deliver files on request to systems that have deployed the file delivery components of the SAM system, employing for this purpose an appropriate choice of file delivery protocols and of routing protocols. The preferred file transfer protocol on WANs is GridFTP.
- 4) Cache files on local disk for the duration of the job, for a lengthier period for re-use, or semi-permanently as specified by the local SAM administration. Files in the SAM cache are then also available to other SAM stations to be copied from one SAM cache to another.
- 5) Provide methods of job submission, for local or grid-aware systems, which are coupled to the file delivery mechanisms and can utilize file location information and system information for performance optimizations.
- 6) Track processing information down to the level of per-file delivery and consumption status, permitting processing to be organized on the basis of the consumption history of a particular dataset (set of files).

## Basic SAM concepts

SAM is organized around the concepts of a dataset, a snapshot, a consumer, a project, and a station. A **project** runs on a **station** and requests delivery of a **dataset** to one or more **consumers** on that station. The consumer is a user application. The dataset is a specification of file metadata, which is resolved by executing a specific database query to a list of files (the **snapshot**). The station is a particular collection of hardware resources. The project is a SAM process which may begin file delivery before any consumer starts, and which continues until the last file of the snapshot is delivered, or until the consumer(s) request it to stop, or until it times out. Files that are requested by a project are delivered from (a subset of) the locations known to the SAM catalog, as specified by the routing protocol for that station, to the station cache, which may be a set of physical disks mounted on one machine, or a distributed cache consisting of disks on a set of nodes. Files delivered to station cache are temporarily protected from deletion until the consumer which needs them has issued a signal to release them. No files are replaced in the station cache until a new project request does not have space, and then files are deleted according to a programmable policy (currently, least recently used). The information about which files have been successfully delivered to the project is reported back to the catalog and stored. Files can be pinned in a station cache; that is, marked as unavailable for deletion until an administrator of the system issues a command to unpin them.

Communication with the SAM catalog is handled through a middle layer process called a DBServer. This is a python process which communicates with the Oracle database and with the other SAM components using CORBA. Deploying multiple instances of the DBServer and configuring SAM components to communicate with particular instances permits a scalable distribution of the communication load (up to the capacity of the Oracle database itself).

## Using SAM

A user of SAM needs to know how to define a dataset, how to run a project, how to create and run a SAM-enabled consumer application, and optionally how to store files into SAM. For these tasks, SAM distributes a command line interface, a python user api, C++ interfaces constructed for the DØ and CDF frameworks, and a web interface as an additional option for dataset creation. DØ and CDF have created experiment specific tools which wrap the running of projects and consumers. Some test sites also have access to the JIM job submission system, which will send SAM-enabled jobs to a grid of SAM stations. Web pages for browsing the SAM catalog metadata, and a growing set of utilities for monitoring the system are also supplied.

## Installing and configuring SAM

Installing SAM has a few prerequisites: a user account on the system called sam with a particular UID; prior installation of the Fermilab product distribution software (ups/upd); system configuration to call 'ups startup' during boot time for those systems which will run production SAM servers. Then, it is necessary to install the clients, the servers, and at least one file transfer protocol understood by SAM. GridFTP is the preferred protocol for WAN transfers, and requires installing in addition the Globus security infrastructure and obtaining the necessary certificates. Sites with firewalls will need system administrator assistance to open particular ports.

## Use cases of SAM

SAM is in use in production by DØ for several different use cases. The DØ online system and several offsite Monte Carlo production centers deploy SAM File Storage Servers, using these to store collider and simulation data into ENSTORE (the Fermilab mass storage system) via SAM. These data are then accessed by the Fermilab DØ systems and by remote DØ systems running SAM stations. The onsite stations are purely Linux systems (the desktop cluster CluED0), mixed Irix-Linux systems (CAB, the reconstruction farm), and the large Irix SMP (d0mino). The CluED0 station is used primarily for small-scale analysis jobs; CAB for large-scale analysis jobs; and the d0mino station for high-throughput applications (picking individual events out of large datasets, distributing large datasets to remote stations). Remote analysis stations have been established at many remote sites; about 20 such stations are active now, with varying configurations.

SAM is not yet in production at CDF, but is being tested on a central Linux system and at several remote stations, for Monte Carlo production and for analysis jobs. CDF is now writing SAM metadata for its raw files from the online system.

## SAM on Grids

SAM stations can be united in a grid with a submission system using Condor and Globus grid tools, as mentioned above. The future plans for SAM include enhancing its main components to permit even more use of Grid tools, including virtual organization tools, technology for creating run time environments on general-use clusters, and multi-layer caching strategies.

## Installing and configuring SAMGrid

Installing a client, submission, monitoring, or execution site for grid-aware SAM usage is similar to installing SAM. The client sites (where the user submits a job that is sent to a submission site) are lightweight, requiring only a JIM product. Other types of site require the Globus security infrastructure (except for monitoring sites?), an XML database installation, and the relevant SAM and JIM products. As is the case for SAM, the Fermilab product distribution software is used. Again, sites with firewalls will need to open particular ports.

## SAM Operations

An important part of a data handling system is its operations model. For SAM, the model is a three-level hierarchy of monitoring and response. An experiment using SAM supplies shifters who monitor the production SAM systems using Web and command line tools supplied by SAM and the experiment. Shifters report problems which they cannot solve to an expert-on-call from the SAM team. The expert reports problems which are bugs or design issues to the developer(s) of the affected components(s).