

**Fermi National Accelerator Laboratory**

**FERMILAB-Conf-98/382-E**

**D0**

## **The Sequential Access Model for Run II Data Management**

Lee Lueking et al.  
For the D0 Collaboration

*Fermi National Accelerator Laboratory  
P.O. Box 500, Batavia, Illinois 60510*

December 1998

Published Proceedings of *CHEP 98*,  
Chicago, Illinois, September 1-4, 1998

## **Disclaimer**

*This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.*

## **Distribution**

*Approved for public release; further dissemination unlimited.*

## **Copyright Notification**

*This manuscript has been authored by Universities Research Association, Inc. under contract No. DE-AC02-76CHO3000 with the U.S. Department of Energy. The United States Government and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government Purposes.*

# The Sequential Access Model for Run II Data Management

Lee Lueking, Frank Nagy, Heidi Schellman, Igor Terekhov,  
Julie Trumbo, Matt Vranicar, Richard Wellner, Vicky White

October 15, 1998

## **Abstract**

The challenges confronting the Run II data management and access system include storing, managing and providing access to the hundreds of terabytes of data. A system employing a Sequential Access Method of delivering files has been designed and a prototype is being built. The design and function of this system is discussed and a status report of the project is provided.

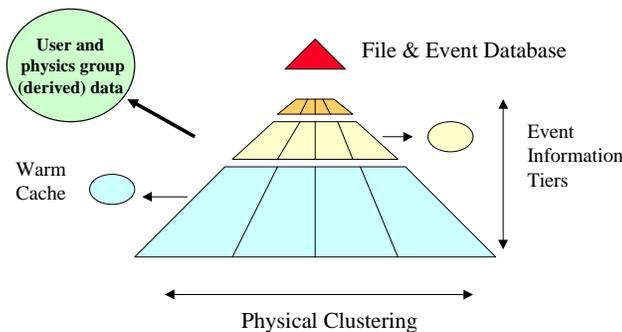


Figure 1: Data organization in the Sequential Access Model.

## 1 Introduction

The scale of the data management effort for the Run II is set both by the anticipated size of the dataset and the number and diversity of the expected analyses. The D0 data set is expected to total 560 TB. Of this, 300 TB will be RAW data from the DAQ system and the remainder needed for reconstructed and subsequent analysis stages. The CDF Run II data will be even larger with an estimated total of close to 1PB with about 500 TB in raw data.

This note describes what is called the “Sequential Access Model” (SAM). SAM shares many features with the data management approaches employed in the previous CDF and DØ data runs with a file-based system of storage and bookkeeping. The goal of this model is to optimize the use of storage resources such as tape mounts and drive usage. In order to facilitate this goal, we have four primary objectives: 1) Clustering the data onto tertiary storage in a manner corresponding to access patterns, 2) Caching frequently accessed data on disk or tape, 3) Organizing file requests to minimize tape mounts, and 4) Estimating the resources required for file requests before they are submitted and, with this information, making administrative decisions concerning which data to deliver. In addition, it is desired to unload the burden of individual file tracking from the analysis physicists, and place it onto the data management system. Reliability of the system is also considered paramount, and must be built into the design from the beginning.

Two major facets of the approach are data organization and data access modes. Details of the data organization are shown in Figure 1. The basic level of granularity in this strategy is the event. Event information is organized into tiers in a pyramid where the lower tiers contain the RAW data and large amounts of information for each event, while the upper tiers contain a refined level of event information. The peak of the data organization contains an event catalog, file meta-data and handles to the underlying file data. Significant in the data organization is physical data clustering, based on anticipated access patterns, which makes the information at each tier more accessible. A discussion of the clustering strategy being considered for DØ is given in [1]. Also, a system of caching the most frequently accessed information improves the overall system performance. A more complete description for the SAM system and its requirements and design can be found in [2], [3] and [4].

In addition to the data being strategically organized, delivery pipelines are defined to satisfy the many users and expected analysis approaches. There are 5 basic types of delivery needs anticipated: 1) File and event database scans, 2) Experiment shared random access disk, called “Thumbnail” 3) Coordinated large dataset processing, referred to as “Freight train”, 4) Random

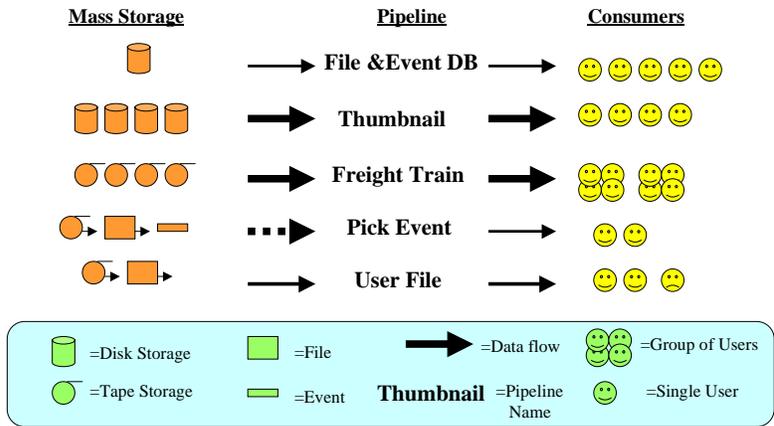


Figure 2: Access modes for the Sequential Access Model.

event picking and, 5) Individual user on-demand file access. These modes of data access each have specifically defined requirements for mass storage bandwidth, disk staging space, processing CPU and the number of concurrent users. These pipelines are briefly summarized in Figure 2 showing their differing characteristics in a symbolic fashion. In addition, there may be other access modes which emerge during the run.

## 2 Details of the System

### 2.1 SAM Design

The system consists of network distributed components making it scalable and versatile. It has been designed to function for all of the access modes described above. It employs CORBA interfaces among the various modules, which are currently written in JAVA, Python and C++. The information required to catalog each file and track all of the processing details are stored in an Oracle 8 data base.

Any data access system is, of course, closely related to many other systems, in particular the Storage Management System (SMS), CPU availability or batch processing systems, and the experiment data model. However, this design is decoupled from these systems making it less complicated than more integrated approaches. This, in turn, makes it easier to build and maintain.

### 2.2 The Main Components

The hierarchy of the SAM system being built is illustrated in Figure 3. Data on tape is managed by the Storage Management System. Hardware configurations are established for various types of data processing, such as farm reconstruction, large scale data processing, or analysis activities. Associated with each of these configurations is a process called a *station manager* which manages the resources, disk and bandwidth, locally available. *Projects* are defined which consist of one or many files to be delivered to the station. Each *project* is managed by a *project manager* and many *project managers* can operate simultaneously on a given *station*. The users of the data in the system are referred to as *consumers*. Each running *project* may have one or more *consumers* performing various activities on the files being delivered to the *station*. When all *consumers* registered with a *project* have completed processing a given file, the file is removed from the local

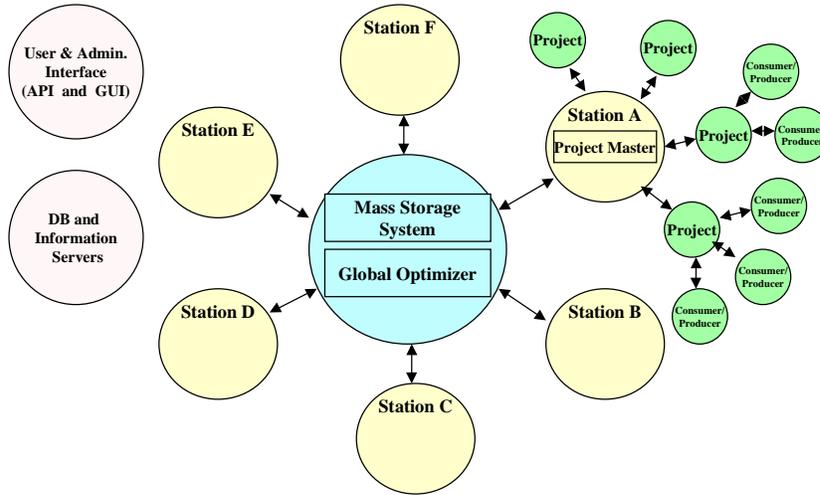


Figure 3: The hierarchy of the components in the SAM system.

storage. As local storage is freed the next file , or set of files is eligible to be delivered from the central storage.

The delivery of data from central storage is performed in a coordinated fashion controlled in two parts of the system. First, each *project manager* consolidates requests for each *project* being managed under its control, and organizes file needs according to location in tertiary storage. Secondly, the *global optimizer* receives requests for files from the many *stations*, and allows them to be executed in a way which provides the desired tape, robot and network resource allocation to each station or type of data access.

### 2.3 The Storage Management System

Although the Storage Management System is outside the scope of SAM, there are several important features required from the SMS.

1. Access to data must be through file-level semantics.
2. All tape activity within the tape libraries, and to and from storage shelf must be managed by the SMS.
3. The system must allow the data to be physically clustered to tape in the manner defined by the experiment. Files should not span multiple tape volumes.
4. A mechanism is needed for sending priorities in conjunction with file requests to allow more control over resource allocation.
5. The system should have a queuing method which helps optimize the use of resources, such as arm time and tape mounts.
6. Retry and fail-over features must be available for failed tape read/write activities.
7. Open tape format to allow removal and exchange of tapes with other sites.

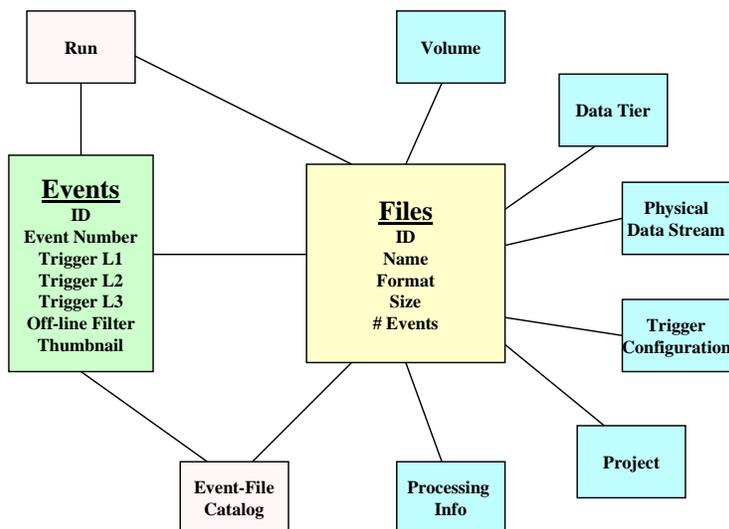


Figure 4: Simplified schema for the event and file database.

#### 8. Reliable and unattended operation.

These features, and others, have been carefully considered and are incorporated in the SMS design which has been prototyped at Fermilab, currently called Enstore. Additional information about this system are discussed in another CHEP98 talk [5]. A more complete discussion of the DØ MSM requirements can be found in [6].

### 2.4 The File and Event Database

The event and file database is a core component of SAM, and tracks all processing activities and file locations within the system. It also provides event level trigger and stream information which enables the system to retrieve individual events from the overall data store. The schema for the database is shown in Figure 4 in a much simplified form. This picture shows that files are a central component of the database, with location, data type, physical stream, trigger configurations, and processing information around them. The event catalog will contain information for each of the  $10^9$  events. For many of the files in the system, specific catalogs will also be maintained which will precisely map the Files table to the Events table. Oracle 8 is used for the database and its size is expected to approach half a Terabyte when all of the Run II data and processing are recorded. A mechanism is provided for importing and exporting data into and out of the system.

## 3 How the System Works

As described in section 2.2 SAM manages the delivery of files for *projects*. Projects are simply lists of files which are created by users or physics groups based on certain criteria which define data sets to be processed. An example of a GUI project workpage used to help users construct SQL queries is shown in Figure 5. This interface is written in JAVA and interfaced through CORBA to the database server which communicates to the database. With this tool, the user is able to explore combinations of conditions to see the number of files and tape mounts required for proposed projects. A test project can be converted into an active project if the user making the request has proper authorization. Authorization levels are based on the number of tape mounts needed for the project and the number of people using the data. In other words, a physics group

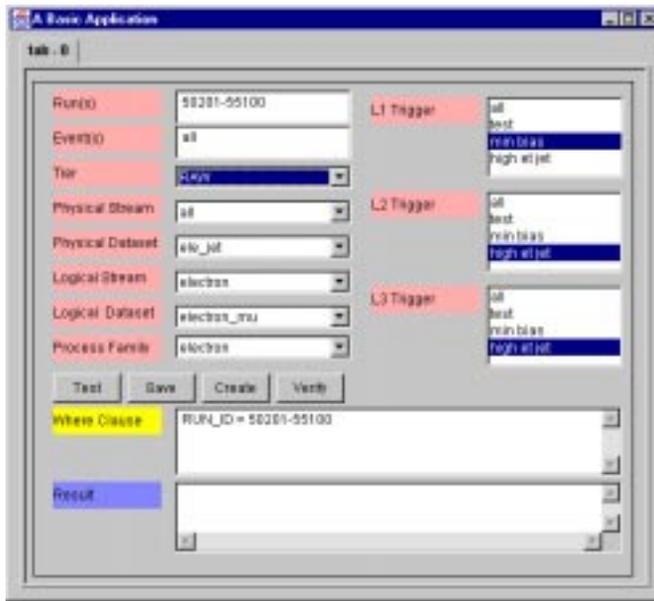


Figure 5: User GUI for creating a project definition.

would be authorized to create larger projects than individual users. A user API is also provided which would enable user programs or scripts to make on the fly requests for small numbers of files without using the GUI.

Once a project is established, it is managed by a *project manager* on a particular *station*. The event trace shown in Figure 6 shows how the system delivers files to a *consumer*. The consumer registers with the *project manager* to work on a particular *project*. The *project manager* sends requests for groups of files which it wants delivered to its *input buffer* to the *optimizer*. The *optimizer* receives requests for delivery from all the stations in the system, and applies policies to decide in which order requests are serviced. The message to start delivery for a group of files is sent to the buffer manager and it, working in conjunction with the storage manager, initiates the file delivery. The job of the system is thus to keep the input buffer filled.

The *consumer* process which is registered with the project manager, simply presents its unique token to the project master and asks for the next file in the project. The project master responds by giving the consumer the name of the next file and recording that that consumer is in the process of using that file. When the consumer closes the file, the project master is informed and marks that consumer as finished with the file. The consumer continues asking for the next file until the project is completed. There may be many consumer processes running in parallel using the same registration token to go through the data more quickly. There may also be many other consumers registered for other tasks processing the list of files for the project. Again, each type of consumer activity has a unique token and is not confused with other consumers by the project manager.

## 4 Project Status

The overall strategy of SAM has been reviewed and agreed upon by DØ as the method for data management for Run II. We have been building a prototype of the system for the last two months and many elements are coming together; the prototype system should be operating by early November 1998. Some data has been entered into the database, testing the import mechanisms. The station and consumer code has been prototyped and is working for a single project manager. We have built a log server which consolidates messages from the many parts of the system and enables us to monitor the performance and debug the entire system.

Event Trace for Reading Production Files from the Mass Store

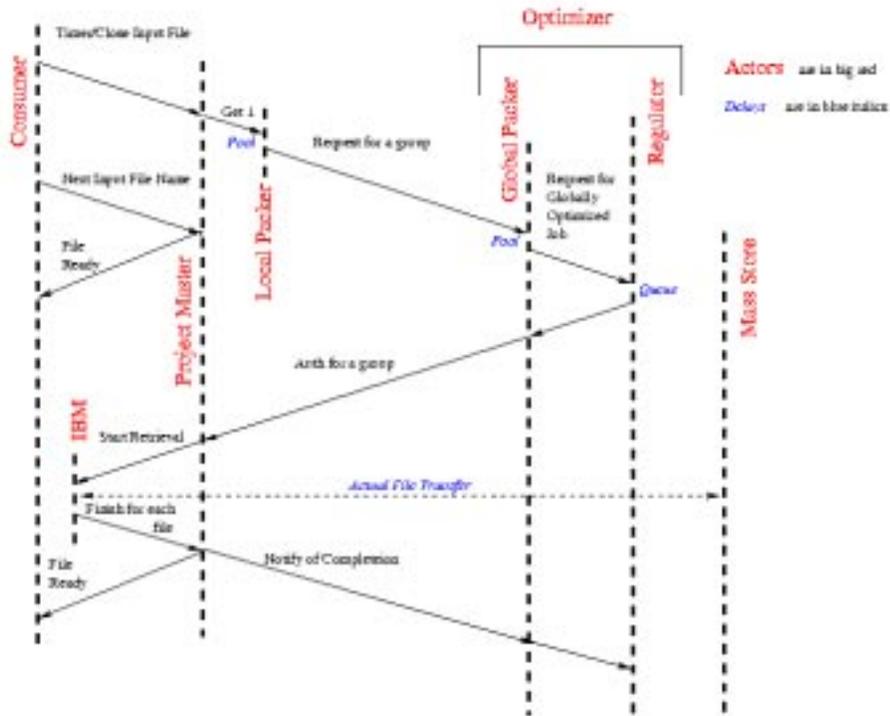


Figure 6: Event trace diagram illustrating how the system manages file delivery to consumer processes.

We have a working version of the database currently in place and have tested many example queries to confirm that the system will be performant enough for our needs. A database server has been built which provides a CORBA interface to all modules in the SAM system needing DB access. We are designing a mechanism for importing and exporting data into and out of the system which enables data movement among the various remote sites participating in the collaboration.

A hardware testbed consisting of a few PC's is being assembled and should be functional in a few weeks. This testbed is networked to the SAM database server and test systems being built for the Enstore project (SMS), and data processing farms. With this arrangement we will test the performance for various parts of the system as well as explore the reliability and need for additional features.

## 5 Summary

A data access system is being built to satisfy the needs of Fermilab Run II. The system employs sequential access to data in files which are stored on tape, under the control of a Storage Management System, or on disk. Users, or groups of users, define data sets called projects for which SAM manages delivery of the data. The system is a network distributed and, though dependent on, is not tightly coupled to the Storage Management System, event data model, or CPU/batch management. The Sequential Access Model has been approved by the DØ experiment for use in Run II and a prototype is being built and should be functional by early November. Additional details as well as links to most of the references can be found at the address given in [7].

## References

- [1] Heidi Schellman, "Assurance of Data Integrity in Petabyte Data Samples", Presented at CHEP 98, September, 1998.
- [2] "Sequential File Access Model Evaluation and Design for Run II", October, 1997.
- [3] "Requirements for Sequential Access Model", June 1998
- [4] "Architecture for Sequential Access Model", June 1998
- [5] Don Petravick, "ENSTORE - An Alternate Data Storage System", Presented at CHEP 98, September 1998.
- [6] DØ Run II Computing and Planning Board, "MSS Software for Run II - DØ Evaluation", June, 1998.
- [7] <http://runIIcomputing.fnal.gov/sam>