DØ Note 3465
CD Note TN0088

Requirements for the Sequential Access Model Data Access System

Jon Bakken, Mike Diesburg, Dorota Genser, Lee Lueking,
Frank Nagy, Don Petravick, Ruth Pordes, Heidi Schellman,
Marilyn Schweitzer, Igor Terekhov, Matt Vranicar, Vicky White

June 15, 1998

Abstract

Designing a system to provide access to the data involves file and event catalogs, movement of files, read and write access routines, careful bookkeeping of physics content, and resource management. All of this requires a framework, not only of software, but of assumptions and interfaces between the many parts of an entire data handling system. Based on the requirements in this document we will proceed to the specification of the core Data Access Framework, which we call SAM (Sequential Access Model).

SAM will make use of the services of other parts of the whole data handling system, such as a Storage Management System. Other parts of the data handling system, with their own specialized requirements and functionality, such as Farm Production Systems, will make use of the services of SAM. The boundaries between these parts of the system and their services are, as yet, still only generally understood. They will become clearer during further stages of specification. These Requirements say only what the overall system must do, not which part of the system will implement the functionality. A concerted effort has been made to state requirements in an experiment-independent way where possible.

Contents

1 Scope

The overall data handling system for the experiment will consist of a collection of:

- hardware components
- software components
- resource management policies and implemented rules
- data organization and clustering definitions and rules
- data storage methods, rates, capacities, and implemented rules
- data access methods, rates, policies, and implemented rules

Together this collection must be built and configured to meet certain goals of the experiment concerning the reliable storage of its data, and the ease and frequency of access to the data. The primary objectives are to store all raw data produced by the online system and to provide access to the data for all subsequent stages of data processing so that physics results can be extracted as quickly and efficiently as possible.

In this document we will attempt to further quantify and elaborate on the above issues.

2 Base Assumptions

The following is taken as understood and much of it is documented in the Data Management Needs Assessment Group Report [2].

1. The total volume of data which will result from raw, reconstructed and various types of summary data will be between 0.5 Petabyte (D0 estimate) to 1 Petabyte (CDF estimate) per experiment [2]. Calibration and Alignment data are additional to this,as are Run Conditions, Luminosity data, Event catalog, and File catalog data.

2. All of the raw data, and almost all of the other data produced, will be stored on some serial media tape or cartridge (we will call this simply tape). We expect the size of the tape or cartridge to be about 50GB and the read/write data rates to be no less than 6MB/sec.

3. Only a fraction of all data will be able to be disk resident at any moment. The exact amount is constrained by cost. It will be on the order of 14-28 TB per experiment [2]. Disk space will be used as a cache for the currently active and most frequently used data. This disk space will therefore have to be managed.

4. In order to automate the mounting of tapes and to provide rapid access to frequently used datasets we will make use of one or more *automated tape libraries (ATL)*. We use the term robot and ATL interchangeably in the rest of the document.

5. The total capacity of the ATLs will be considerably less than the total amount of data and as a result much of the data will reside on shelf resident tapes. There will be frequent (on the order of once or twice per day) changes made to the tapes resident in the robot, as Raw data passes through reconstruction processing and new Raw data is entered.

6. There will be sufficient Farm processing power provided to keep up with reconstruction in real time, plus approximately 80,000 MIPS of analysis processing [2]. We need to manage our data delivery so that we can most effectively utilize network bandwidth, CPU and storage resources.

7. Each physics group, and each individual physicist, would ideally like to repeatedly traverse large datasets of relevance to them, quickly. In order to achieve this, for most users, most of the time, prioritized and optimized access to the data will be done. This will be achieved by 1) enforcing policies on clustering of the data, 2) providing access to the data only via the *API* and *application framework* provided by the system, and 3) by partitioning and allocating resources to particular modes of access. It is accepted (well almost!) that we can define an event data unit which satisfies over 80 percent of the analysis data access needs, and which is between 50 and 100 KB/event. For full details of the baseline assumptions on sizes and streaming see [1] [5].

8. We need to do a good job on *bookkeeping* of where all the *files* are, how they were produced, what the associated *run conditions* were, and which parts of the data were the source for physics results. The *bookkeeping* information needs to be easily accessible, with accurate reports available.

9. All of the hardware components of the system will be commercial CPUs, disks, tape drives, ATLs, network interfaces and switches. In other words, we will not be building any of this hardware ourselves, merely integrating it.

10. Some of the software components will be commercial or free-ware components not developed in-house specifically for this data handling system. Examples are: operating systems, compilers, scripting language interpreters, device drivers, ATL control software, *storage management software (SMS)*, *database management systems* with their associated query and reporting tools.

11. Some of the software components will need to be developed in order to provide the necessary framework for the data handling system and to provide functionality not found in off-the-shelf commercial or free-ware components. This part of the system we are calling the *Sequential Access Model (SAM)* system, and the project to develop, integrate and test it - the SAM project.

3 The Sequential Access Model

Several somewhat different approaches to data handling and data access have been under discussion in the Data Access working groups.

The essence of the Sequential Access Model (SAM), and the origin of its name, lies in the following:

1. A *file* is the primary storage unit which the overall data handling system manages and provides access to. The consumer of data is provided with a file of *event data units (EDUs)*, of some size and type, and is expected normally to process the entire file reading event data units from it SEQUENTIALLY. The system is optimized for this type of access.

2. An *event* is a logical storage unit which the overall data handling system catalogs and can arrange access to. Rarely consumers are expected to want access to one particular *EDU* in

a file. The system, although providing this facility, is not optimized for this type of access. Unless future optimizations to a file format are adopted access to a single EDU in a file will require reading SEQUENTIALLY through the file in order to access the required EDU. The only exception to this is *event catalog* information, which is expected to be accessed randomly through a database management system interface.

3. Groups of users who wish to iterate over a large physics *dataset* will be identified and will declare their intentions to the system. They will each be presented files for them to access SEQUENTIALLY. Fetching and providing access to the files will be done in a coordinated way, in order to minimize the total number of tape mounts and the network bandwidth needed. The system is optimized for this type of usage. The data organization is also optimized to assure that the data in such physics datasets are clustered (streamed) into physically contiguous storage in files.

4. Users who wish to iterate over smaller *derived datasets* will use the system to cluster their data into files which then can be accessed SEQUENTIALLY most of the time. They will either use the data handling system to manage the migration of such files to and from tape storage, or they will place those files in private disk areas, not managed by the overall data handling system.

5. Users who wish to access data which is not clustered in a manner which matches their access pattern will expect to declare this intention, and to specify their data selection, up-front, to the system. They will also expect that their access will be subordinate to the structured method mentioned above and that they will vie for a limited set of resources allocated to such access modes.

The proposed data handling system also includes many features not specifically related to this mode of sequential traversal of files. All data is, in the end, stored in physical files whatever the nature of the clustering, streaming, and organization of data, and whatever storage facilities and modes of access are provided. Many elements of the system are not specifically related to sequential access. Their implementation details depend on a) design decisions and technologies chosen b) the amount of "automated" (as opposed to manual) resource management and status information which the system will provide and c) the particular dataset selection tools which the system will provide. These elements include:

- A *File catalog* of all files known to the system
- A *Run catalog* and a *database* of *Run Conditions* information, including Luminosity data
- An *Event catalog*
- Resource management components including optimizing the use of certain resources, such as robot arm, tape drives, disk cache, etc.
- Components which translate 'logical' requests for data in terms of physics criteria or run conditions into 'physical' locations of data in files.
- Transaction logging components - to support summary usage statistics
- Quality Assurance, Monitoring, and Error Handling components
- User Interface and Configuration components
- Reporting and statistics on the use of the system and the status of the data.

4 Primary Requirements

1. The system must store and manage data files which may reside on disk, tape in a robot, or tape on shelf. Files created in various ways, including from the online system, from production farm processing, from Monte Carlo, and from a user writing his own file of event data or n-tuples, will be imported into the system for storage. Subsequent access to those files must be provided in an efficient and optimal manner. The movement and migration of files in the system must be managed using algorithms with configurable parameters.

2. The system must support the following access modes for the users:

   (a) Large *dataset* processing, in a sequential manner. For datasets of sizes 1-100 TB one cycle of access should last no longer than a month, and all the "registered" users must be able to consume every event in the dataset. Several such large datasets should be able to be processed simultaneously, up to a total of about 20 percent of the total data expected in Run II in a one month period.

   (b) File or small dataset: the user accesses a few files. Depending on whether the files are currently deemed active by the user's experiment, the turn-around must be several minutes or 1-2 days.

   (c) *Event picking*: random access to RAW and reconstructed events, the times are the same as for file access.

   (d) Databases: intensive use of event/file/run catalogs necessitates use of a robust commercial database which must provide flexible query and reporting capabilities and respond to most queries in a few seconds maximum. We must be able to determine if a given file is disk resident or must be obtained from tape.

3. In addition to the above "read" access modes, there will be provision for several modes of writing data. The RIP project will enter data into the ATL and some process will then make it known and available for access, via this data handling system, within 15 minutes of being written to tape. Farm processing subsystems will write files into the system. A coordinated group of users processing a large dataset (a project) will write files into the system. Individual users may write derived data,in the form of user-generated files, back to the system for subsequent retrieval in the same way as other project generated files are retrieved. The "import" of user data into the system will be done in a controlled and overall consistent manner. The system will catalog all known files recording the format of the contents, and, as appropriate, associated Run, Luminosity, Data Stream, and input files, will be stored in the catalog.

4. All access modes must be coordinated by the system so as to balance hardware usage, bandwidth for data, and user latency. SAM system and User processes will be assigned different priorities and clearance levels, based on the process' mission, amount of information requested, and politics. The configuration of these parameters will be flexible and programmable and will be controlled and viewed by a friendly and easy-to-use user interface to the system. Process priorities/privileges will aim at guaranteeing:

   (a) No (user) process interferes with *DA* writing the RAW data.

   (b) Writing of reconstructed data and/or user generated files is not severely compromised by user reading other data at the same time.

(c) No unreasonable user request will compromise overall system performance, nor lock out other users from use of the system.

(d) Resources will be partitionable, so that efficient serving of a large dataset to a large number of people would not be compromised by random file/event accesses.

(e) In critical situations, authorized users will be granted prompt access to any requested data thus bypassing reasonable prioritization.

(f) Mechanisms will be provided to ensure that, when necessary, writing of tape cannot be disrupted or delayed by read requests for files on the same tape volume.

5. To facilitate the most efficient access to the data, it is anticipated that data will be clustered onto physical tape volumes in a fashion believed to coincide closely with subsequent usage patterns. This means that like types of physics triggers, and data formats, will be stored together on tapes. The system will provide ways to specify the data clustering and to write data accordingly.

6. The SAM system will provide access to data, and specification of *projects*, based on SAM user and *analysis group*.(How this corresponds to UNIX system user and group will be determined later). Users will belong to one or more *analysis groups*. Serving of large datasets will be done in the context of an analysis group or *project*. Resources of the system will be allocated and controlled on a group and project basis. Disk included in the system will be managed by the system and files will be located, tracked, and migrated between storage levels. Use of tape, disk, robot, network, and cpu resources will be optimized based on

   (a) overall system considerations, as listed in 2) above

   (b) group

   (c) access mode

   (d) project - i.e. the particular activity in progress

7. The system must be tolerant to hardware failures (tape losses, tape drive failures, disk failures, network disconnections etc):

   (a) No data will be lost beyond the reasonable and inevitable (less than 1 percent) tape loss of RAW data. Absolutely no data will be lost due to system software errors.

   (b) A small data subset will be designated by the physicists as crucial. The crucial data will never be lost even with any normal tape loss rate; the appropriate replication will be provided

8. There will be extensive and easy-to-use *Monitoring* and *Auditing* facilities provided as part of the system. These will allow users and administrators to not only monitor the state of the system and track their use of the system, but also to guage the effects of failures and data losses. It is important that the system provide accurate and rapid feedback to users about the status of their requests.

9. Above all, the system must be kept simple, partitionable, distributable, and configurable.

   (a) It must be able to handle multiple distributed processors, all working on the same dataset.

(b) A subset of the data handling system must be usable on remote institutions, without needing a commercial database.

(c) In the systems's simplest form a *consumer* must be able to access event data by simply reading a file, using a minimal version of the system *API*.

(d) A partitioned piece of the system must be able to handle as much of the resource management, file inventory, and consumer access as possible locally, before consulting with, or accessing files using global system resources, such as database and ATL.

5 Non-Requirements

Either outside the scope of this system or undesirable are the following features:

1. The disk space allocated to an individual and used to store private datasets is outside of the scope of the system and will not be managed, backed up, or otherwise automatically touched. The system will simply deliver files to user disk locations, and store files from user disk locations, when requested to do so. The data handling and access system is distinct from an automatic archiving and backup system.

2. SAM is not a batch system. Although users may be informed about which station, or machine they should run their analysis or other processing program on, their job will not be run on that machine - it will be their responsibility to initiate the job on the appropriate machine.

3. There will be no attempt to track centrally the processing of each event by each consumer will be made. Transactions and status recorded in a database should be at no finer granularity than a whole file. Connections to a database during a user job should be kept to a minimum, sufficient only to provide access to the necessary calibration, alignment and luminosity information.

6 Detailed Requirements

6.1 Data Organization

Data resulting from a single event (collision in the detector) will be organized and physically grouped with data from other events along two different axes, which are described in the next two sub-sections.

6.1.1 Data Tiers

The system must support the concept of *data tiers*, or branches, of event data [5]. There are many different clusters of data, which we call *event data units (EDU)*, associated with a particular event. Each different type of EDU has a different set of contents in terms of its member banks, classes, structures, etc. Each EDU is stored as a unit - i.e. in close physical proximity. Each different type of EDU is a different tier of data. The system must support the separate cataloging of different data tiers.

1. At minimum, the system will provide for the following tiers of event data to be cataloged and accessed, by tier:

   - *Raw event*

- Fully reconstructed event, including possibly some parts of the raw data (EDU250)
- Summary *Reconstructed event* (EDUnnn where the size nnn is between 50 and 150 KB)
- Highly compact summary physics data (EDU5 or *Thumbnail* event)
- *Event Catalog* - i.e.. a few up to 100+ bytes of information about each event
- User-defined or Unknown

2. Collections of EDUs of the same data tier will be stored together physically in files. Each file will therefore be cataloged as belonging to a particular system-defined data tier.

3. Heterogeneous collections of EDUs of different data tiers may be mixed in a single file. However, that file will be cataloged by the system as belonging to a User-defined or Unknown data tier.

4. Additional data tiers to those listed above may be defined.

5. For the Raw data tier each file will contain EDUs from only one *Physics Run*.

6. Data from the Thumbnail tier will be highly dynamic and will be replaced or evolved frequently, as understanding of the physics evolves.

7. For every event (collision in the detector) there will be data in the system for the Raw data tier, and for the Event Catalog tier. Data for other tiers may be present in the system.

8. Data of a particular data tier, for a particular event, may be present in the system in more than one file. Extremely careful bookkeeping is required.

9. We must be able to find in which file, or files, in the system a particular EDU resides. This system must be able to create, either permanently or temporarily, a complete directory of the contents and layout of each data file.

6.1.2 Data Streams

Apart from physically clustering together data of a certain tier, the system will support a classification of data based on physics criteria:-

1. The system will support the concept of a *physical data stream*. Events will be classified as belonging to a particular physical data stream based on physics criteria, such as trigger and filter bits.

2. The system will support the concept of a *logical data stream*. Events will be classified as belonging to a logical data stream if the *meta-data* describing them, such as their trigger, their physical data stream, the range of Run numbers, etc. satisfies certain criteria, recorded as part of the logical data stream definition.

3. Files will be cataloged as belonging to a particular physical data stream when all the EDUs within the file belong to that physical data stream. Files which contain heterogeneous data which cannot be classified as belonging to any known physical data stream will be cataloged as belonging to data stream Unknown or All.

4. The system will support and maintain associations between logical and physical data streams. It will provide for data streams to be hierarchically composed of other data streams. It will provide access to data based on specification of data stream (either logical or physical) at any level of the hierarchy. The access to data will be efficient only if it is based on physical data stream, or set of physical data streams.

5. The *storage management system* software (SMS) which controls the writing of files onto tape must provide control over which files, belonging to which physical data streams, are placed on a tape, and in which order.

6. Since we cannot guarantee, at this point, that the SMS software will write a file to one single tape, the file catalog of the system must support the possibility that a file is made up of an ordered pair of file fragments, each residing on a physical tape.(We will not allow for the possibility that the SMS software writes a single file to more than two tapes!) It is highly desirable that a single file is written to a single tape. Imposing such a requirement on the SMS software would simplify the *File Catalog*.

7. The ability to catalog whole tapes as belonging to a particular physical or logical data streams desirable.

6.2 Data Storage and Data Migration

In practice, when considering storage of data, we are concerned with five levels of storage hierarchy. Files, in any particular system or mode of access, need not pass through all five of these levels.

- The user's own disk - not managed by the SAM system
- Disk managed by the SAM system and used as a buffer or cache for files
- Disk managed by either the SAM system or the SMS and used as either a buffer
- Tapes in a robot-controlled tape library
- Tapes on a shelf - i.e. a manually controlled tape vault

1. The Raw data must all be stored on readily accessible tape media. If the tape is not in a robot controlled tape library then it must be in near-line shelf storage from where it can be imported into the robot in a controlled manner.

2. An operator should be able to visit a robot not more than a few times (ideally once) per day to perform the exchange of tapes between robot and shelf. The system must batch up requests for such tape exchanges, not to exceed the tape library's holding/exchange area, for once, or twice, per day exchanges. The number of tape volumes to be exchanged in a day is largely determined by the rate of movement of Raw data through the Farm systems. At full rate each experiment will be writing approximately 50 50GB tapes in a 24 hour period. The robot system must be sized to handle at least 100 tape exchanges in a 24 hour period, per experiment. The SAM software framework must create the list of tapes for exchange in a suitable format, for processing by SMS software/operators.

3. Raw data will be written into the ATL, in a manner which keeps up with the demands of the experiment data acquisition and online system. It is the responsibility of the *RIP*

project to ensure that the Raw data is stored and then within 15 minutes of closing the tape which it is written on, made accessible by the SAM data handling system, which must 'synchronize' with the SMS system. Detailed requirements for the rates, reliability, manner of presentation and storage of data, buffering requirements and error procedures can be found in the RIP requirements document [3].

4. Not all of the processed data will necessarily be retained in any one of the levels of the storage hierarchy. Summary data only may be retained, at some levels.
5. Some of the tape resident data will be also cached on disk. The size and nature of each disk cache will be configurable, under control of SAM administrators.
6. Some of the frequently used data, or most recently accessed data (such as random events *picked*) will be cached on disk.
7. The thumbnail data and the event meta-data will be stored on disk.
8. Users will be provided with an API, through which private datasets may be stored in the system, and migrated from disk to tape and back, with file catalog information, provided the user has sufficient privilege and quota (of disk, or tape, or robot slots) to perform the operation. This may be provided entirely, or partially, by the SMS.
9. The system will record, with each file of data stored, pertinent information about the creating process. This will, for example, specify the exact version of *Reconstruction* Code used in creating the file, as well as information about other processes or data, including input files, on which the file depended. Private datasets will require a subset of this catalog information to be presented along with the file. It is highly desirable that file formats allow for inclusion of the relevant file catalog information within the file itself.
10. Users must be able to mark files as busy, or unavailable for flushing from the disk cache and also to mark files as no longer in use, and therefore available for flushing from the disk cache should disk space be needed. This feature must be protected with appropriate privileges and access rights.
11. Requests for migration of files from robot resident tape to disk must be queued and ordered, based on a) the type of access requested, b) the priority of the project, group or user. Global optimizations of the use of robot arm, tape drive, disk buffer and network resources will be part of the system, based on parameters which can be readily controlled by an administrative user.

6.3 Read Access to Data

1. Five kinds of read access must be supported :-
   - *Production* – which is farms reading the data as a coordinated group of consumer processes together stepping through a dataset.
   - *Freight Train* – which is a coordinated group of users all stepping through the same dataset together.
   - *Files on demand* – which is access to particular files, non apropos to freight train access
   - *Pick event* – which provides access to, and builds, a disk-resident cache of interesting single event data. Many picked events may be bundled into a large file for convenience.

- *Thumbnail* – which is access to a subset of the disk-resident summary event data

In addition users must be able to browse and query on the Event Catalog and other Event and Run meta-data.

2. The read access loads on the system will vary greatly at different times in the experiment lifetime. From a period of great chaos and frequent re-reconstruction to a final period of steady state processing and heavy analysis. To assist in estimating data access loads we have divided the stages of the experiment into four time periods or "Seasons". For further details please refer to Season's of Sam [1], which contains detailed modelling of estimated system loads. These will guide the design of the system and constrain the hardware purchases needed to provide these levels of performance.

3. Each consumer of data, or coordinated group of consumers, accessing data in one of the supported access modes will be allocated a set of resources. In some cases projects may share certain in-demand resources - such as tape drives, robot arm, network bandwidth. The allocation of these resources must be very easy and intuitive, with an easy-to-use interface and a simple explanatory summary and diagram showing how all the resources are allocated and how they are actually being used.

4. There will be an API so that each consumer of data can specify the data they wish to access. This API will permit data to be specified in terms of a database query on any of the quantities stored in the file catalog, event catalog, tape volume catalog, run conditions or luminosity databases. A GUI will also be provided to specify this information. It must be easy to view and make reports on, all of the different projects in the system, and their status.

5. Most files are expected to be about 1GB in size. However, because of concerns about data loss leading to periodic closing of files, some files may be considerably smaller. The system must be able to handle files of any size, fromzero bytes to tens of GB. It should be optimized for 1GB file sizes.

6. The system will not normally respond immediately to a user's request to access a shelf-resident tape. Instead, it will batch up that request and return a status to the user indicating that there will be a delay before access to the data can proceed. Requests for tape to be exchanged between shelf and robot will be made using SMS and ATL system facilities. The parameters controlling the batching of requests will be configurable.

7. There will be a general provision in the system to schedule work to be done in fetching, writing or managing files and disk space at suitable times of the day, based on priorities and rules encoded in configuration parameters of the system.

6.3.1 Freight Train Access

1. A coordinated group of users who wish to sequentially read through a large dataset together define the dataset (as a project). A user with appropriate privileges must be able to specify such a dataset in terms of Data Tier, Physical or Logical Data Stream, and other Run Conditions. Users, with appropriate privileges, must be able to register to participate in a particular Freight Train. This registration must be available through a program API. Starting up a Freight train creates a data delivery pipeline of a particular dataset, to many users, each of whom will eventually consume all of the data.

2. Many Freight Train data pipelines must be able to run simultaneously - resources permitting. If more than one Freight Train is running simultaneously it will be because different groups of people want to look at distinct physics streams. Therefore it is unlikely that any two Freight Trains would be simultaneously looking at the same files. It follows that there would be no significant benefit in caching the Freight Train data for use elsewhere in the system.

3. Each Freight Train shall have a disk cache, which serves also as an input disk buffer. The disk cache must be well matched to the rate at which data can be delivered (limited by the aggregate speed of the tape drives allocated to the Freight Train).

4. The disk input buffer must be sufficiently large to allow all users being serviced by the Freight Train to process their files. The users consuming the data will not be in event-by-event lockstep; there will be some spread.

5. The application framework must provide facilities for

   - terminating a consumer who is unduly holding up a freight train for whatever reason (code error, excessive processing time, etc.)
   - allocating additional CPU resources to a consumer whose processing requirements are greater and who would otherwise hold up the Freight Train. This is a desirable feature, but may not be mandatory, especially in the initial implementation.
   - re-starting a user who resumes processing of a partially analyzed dataset

6. In general, a Freight Train data pipeline needs to present statistically unbiased data across the entire dataset, or data stream, to which it is providing access.

7. Every user application must be able to obtain an accurate and complete account of exactly which events they have processed in the course of a particular circuit of a Freight Train. Also the system must ensure that the overall status of the Freight Train and all applications involved can be easily monitored, say with a web interface, charts, etc.

8. Each Freight Train shall have one or more output disk buffers.

6.3.2 Farm Production

1. Farm Users and Managers must be able to make requests for data to be processed, using a particular Reconstruction program version. They must be able to specify the data to be processed in logical terms, which must be resolved into physical data streams and/or files by the system.

2. Farm Production data access rate is constrained by the DA and by the Farm's need to keep up with the DA, plus an allowance for a certain amount of re-reconstruction. The repeat reconstruction rate is estimated to vary over the different Seasons. See Seasons of Sam [1].

3. Input buffer disk belonging to the "Farms" should be well matched to tape speed, so that the Storage System Software can deliver data directly to Farm input buffer disk. The system will not send a file to the Farm unless there is room for it on the Farm disk. The Farm processing software will arrange for Farm worker nodes to access the files either directly, or by moving them to disk local to a Farm worker node.

4. The data cached in Farm disk (as input buffer to the production processing) need not be coordinated or shared with other potential users of the same data in the system. Most other data cached on disk will be available for sharing across all data delivery pipelines in the system.

5. The system will provide foolproof and robust ways for files which are the output of the Farm to be written to tape and cataloged in the system.

6.3.3 Files on Demand

1. The system will provide access to single files, or collections of files. Users will be able to specify the files to be accessed by specifying a logical dataset selected by trigger, stream, run and run conditions criteria to be satisfied.

2. Entire files will be presented to the consuming process - the API in the consuming process will be responsible for skipping over unwanted events which do not match the dataset criteria specified.

3. The system will attempt to provide statistically unbiased data across the set if files being delivered.

4. This mode of access will be subject to restricted total resources

5. Files fetched from the ATL, or tape shelf, and cached on disk for one user will be available for access by other users as long as they reside in the disk cache. They will remain in the disk cache as long as they are

   (a) In use by the user which requested them

   (b) Marked as not available for removal by a user with privilege to do so

   (c) Not considered to be a candidate for removal from the disk cache, based on the configuration parameters and cache policy in use

6.3.4 Pick Event Access

1. The system will maintain a disk resident cache of individual EDUs which have been requested. If this cache of 'picked' events should become excessively large then the least frequently accessed EDUs will be moved to near-line, readily accessible tape.

2. A user must be able to specify a list of Events for which a particular tier of data is to be provided. An API, will be provided for specification of this list of events.

3. A query of supporting Run and Event meta-data, which results in a list of Event numbers, must also be an acceptable way to provide a list of events to be picked.

4. In order to minimize tape mounts and maximize efficient use of tape drives, requests for Events to be picked will be batched up for execution. The parameters which control the delay and other criteria for batching up requests will be configured for the system.

5. It is highly desirable that a selected individual RAW EDU can be submitted to Farm Production for on-the-fly reconstruction and the resultant EDU also cached in the Pick Events cache

6.4 Write Access

1. Data files which are written in various ways, such as through *RIP*, Farm processing, Production filtering, Monte Carlo and User Analysis, must all be able to be entered into the system. There are, in general, two types of such data:

   (a) Private will include only minimal meta-data information and therefore may be accessible/useful only to the person or group who inserted the data.

   (b) Public will include a full complement of meta-data which would allow any user the ability to correlate the imported data with other data in the overall store. This means that any meta-data parameters, such as processing steps, parent data sets, trigger configurations, luminosities, et cetera, could be easily acquired using standard techniques.

2. It is desirable that standards be established which would make the data, at least partially, self describing. This would mean that a set of parameters needed to enter a file into the metadata would be part of each file. Even if the files are self describing, there is a need for an independent mechanism to be provided for the operations of specifying file meta-data and cataloging information.

3. The system must provide an API to specify which files, and possibly in which order are to be written together to a tape. Details of this depend on capabilities of the Storage Management software system.

6.5 Databases

Several classes of data are to be stored in one or more Databases, as opposed to simply stored in data files as the physics event data is. Classes of information stored in a Database shall include, but not be limited to

- Meta-data, which is "data about the data "
- Data Inventory
- Transaction data
- System configuration data

Specific requirements on the data stored in databases, and on the data management system for the databases, are:

6.5.1 Meta-data

Metadata describing the physics and processing content of the data are included in this category.

1. File catalog will contain a few 10**6 entries.

2. New file catalog entries must be able to be made at the rate of several per minute.

3. Must be able to retrieve the status and location of any given file within a few ms.

4. Event catalog will contain a few 10**9 entries, up to approximately 100 bytes each.

5. New entries must be made to the event catalog, without significantly degrading the live system, at an average rate of 100 Hz. This may occur in batches at high rate, rather than continuously as recorded by the online system.
6. The system must be partitionable to allow entry, update and backup to take place without significantly degrading performance.
7. There will be a creation history for each file, including per file accounting of processing details.
8. Physics data stream definition and version, both logical and physical.
9. Event Catalog with at minimum stream ID, trigger bits, filter bits, run number, event number, for each event.
10. Run information including begin time, end time, description and conditions.
11. Production code version for each processing step. Other identifying information about creation of datasets.
12. Timestamped Luminosity data

6.5.2 Inventory

The inventory contains a complete list of the locations and status of all files available from the storage *warehouse*. It is also needed to determine the most readily available source of a requested file. The inventory must include:

1. Each physical file and its location in long term storage.
2. Each physical cartridge or cassette, its location, type and status.
3. Prioritized list of devices and file systems in which to look for files
4. Catalog of picked events which are archived to tape.
5. Catalog of files and picked events available in cache.
6. Event range/map to identify the events included in each file (for some files at least)
7. Lost Data/Luminosity Data map to identify invalid Luminosity blocks.
8. The Storage Management System must be able to build and export a map of what data is on which volume, so that global optimization of access can take place.
9. The Database must be able to retrieve full information about any file within 1 second.
10. The Database must be refreshed and updated, each 24 hours, with the data exported from the SM system. This update must not significantly degrade the performance of the system for more than a few seconds.

6.5.3 Transaction data

The transaction data is a record of how the data is being used. This information will be used to track projects and will help to optimize the system for efficient hardware usage. It must include:

1. Project definitions
2. Project status information
3. Project transaction summary
4. File migration summary
5. File usage summary

6.5.4 System Configuration data

Configuration data will be used to define the processing resources available for each task, or project. It must include:

1. Allocation of disk, tape, robot and network resources to partitions of the system (stations) and projects.
2. Users and groups
3. Priorities for various groups and classes of service.
4. Rules for optimizing the movement between tape and disk storage.
5. Disk buffer and cache policy parameters..
6. Hardware configurations, such as nodes assigned to a particular project.

6.5.5 Database Management System

1. Database shall be available 24x7 for creation/update of the meta-data and the transaction data.
2. Database shall be available 24x7 for query access by SAM users
3. Database shall be protected against power outage, network failure, or other such glitches as much as physical power supplies and network connections allow. This implies the Database Server system(s) must be considered "high availability" machines. Implementations will consider multiple servers on different power grids, and different routers/subnets.
4. Database shall be protected against loss of individual disk drives. (by mirroring and/or replication of disks?)
5. Database shall be recoverable within 12 hours from more disastrous, but extremely rare, errors, such as loss of or corruption of a disk and all its duplicated/mirrored disks.
6. Database shall prohibit users from performing disastrous functions, such as deleting all event catalog information.
7. Transactions feeding meta-data from online sources shall be recoverable, allowing them to be rerun after the problem causing the failure is addressed.

8. Shall provide tools for monitoring access and performance, allowing administrators to tune the database as needed per online data feeds and user consumption.

9. Shall provide query tools which allow for arbitrarily complex queries based on all of the information stored in the database and all of its relationships.

10. Shall provide report-writing tools, including an easy to use web interface.

11. Shall provide multi-user read/write access to the database

12. Shall be able to be backed up at least daily, without bringing down the system.

6.6 Resource Management and Optimization

1. When a disk buffer in the system is replenished there should be enough space to request several files from the same tape, to minimize tape mounts.

2. The SMS should accept lists of files to be fetched together.

3. Tape drives and robot resident tape volumes must be allocatable resources. The SMS system must support this resource allocation.

4. The system should attempt to locate requested files in any one of its managed disk caches, in level 2 or the hierarchy, before requesting that the file be retrieved from tape.

5. Files-on-demand access consists of access to files of data outside of the Farms or Freight Train access. To encourage people to use freight train access there will be a cap on files-on-demand total bandwidth.

6. Files-on-demand may either be delivered to user-managed disk, or to system-managed disk cache.

7. Preferably the SMS should accept a request to deliver a part of a file, length N bytes, starting at M bytes.

8. The system will provide algorithmic prioritization of tasks which are parameter tunable to most efficiently use available resources.

6.7 User Interface and Control

1. Interacting with the system via GUI and user API: Interacting with the system will be provided via web-based GUI's and API's. Any part of the system which is controllable or configurable through a GUI, should also be accessible through an API to allow programs and scripts to automate data processing and administrative procedures.

2. Requests for data: Users must be able to make requests for data and receive it in an organized and reliable fashion. It must be easy to check the status of a request and suspend, modify or cancel all or part of a request at any time.

3. Describing data: Data should be described "physically" by listing files by name, or "logically" by specifying constraints on parameters included within the metadata. Information contained in each filename should include: physics stream, run number, file sequence within the run, date and time of creation, EDU type, specific format information, some form of encoded processing information. The information which will be available for logical queries

will include: run number, trigger configuration, trigger patterns, physics stream, EDU type, reconstruction or other processing version, creation dates, instantaneous luminosities, and others to be specified.

4. Levels of user/administrator ability/motivation: There will be multiple levels of user and administrative involvement with and understanding of the system. The system should be designed to accommodate all levels. It should be simple to learn for novice users, including self documenting user interface features. For advanced users, the system should be rich enough to allow for complex queries and advanced procedures. Several levels of administrative privileges should be allowed by the system so that it is difficult for novice administrative help to make major, unrecoverable mistakes.

5. Project status: Users and administrators should be able to monitor the status of requests.

6. There must be an application framework under which all user code operates. The framework and APIs for data access must be capable of operating in reduced form in an environment outside of Fermilab, where perhaps minimal, or no, supporting databases are available.

6.8 Data Replication, Export and Import

1. A selection of files should be able to be exported to a remote institution, where they can be read sequentially without benefit of a data handling system present at the remote institution.

2. A subset of the Database and the data handling system should be exportable to a remote institution, for use as a local data handling system.

3. An entire tape should be 'xeroxable' for export.

4. Modifications made to an exported dataset (collection of files) and to the description and meta-data which go along with it should be able to be re-imported into the system.

6.9 Monitoring, Auditing and Error Handling

1. The system will be idiot-proof:

   (a) No user process may ever jeopardize the data.

   (b) No unreasonable user requests will jeopardize the system performance.

   (c) The system will provide protections again accidental deletion of data by either users or administrators. This may take the form of confirmations, file wastebaskets or undo commands. It is understood that mistakes will happen and the system will help to ensure that they are not catastrophic.

2. A complete set of monitoring and performance tools must be provided with the system. These must provide current status for any part of the system as well as performance reports (charts and/or tables) for specified intervals, hourly, daily, weekly, et cetera. It is important to spot problems, identify bottlenecks, and report performance for the system, both to the users and to the administrators

3. The system must carry out extensive bookkeeping functions in order to ensure that the effects of missing or corrupt data are properly accounted for in physics terms. Integration with a Luminosity Database must be provided.

4. Physics auditing: A summary of important analysis parameters including missing data, data included, integrated luminosity should be available to a consumer of any project. .
5. The system must return detailed errors to user APIs, in the case of read/write failures, indicating the exact nature of the failure, and from what layer of the hardware/software system it came.

6.10 Data loss policy, actions and recovery

All reasonable actions should be taken to manage data loss. Given the total amount of data we are discussing, some amount of loss is inevitable due to hardware and/or storage media failure. Each experiment must determine the amount and nature of acceptable data loss.

1. Mechanisms shall be provided to permit adequate replication of data, in order to manage the data loss and keep it at an acceptable level.
2. Monitoring and quality assurance tools shall be provided to perform verification that the levels of data loss are indeed within acceptable limits.
3. The number of mounts for each tape will be counted. When tapes reach a threshold number of mounts they will automatically be copied and the original archived in a safe place.
4. Tapes receiving unusually large numbers of mount requests will trigger an alarm and attempts to understand the nature of the access will be made.

6.11 System Management and Configuration

1. The entire data handling system must be modular so that additional resources of all types - robot, tape library, cpu, disk, tape, network connections - can be added incrementally.
2. The parts of the system which will be used on various processing locations should be easily install-able, like a ups product.
3. Platform support and portability: The parts of the system which affect consumers should be portable to all systems supported for Run II, including various UNIX platforms and NT. There may be additional platforms off-site which also need support.

7 Dzero specifications

7.1 Storage Management System

The Storage Management System (SMS) is needed to store and manage the large quantity of data planned for Run II. It is assumed to include tape storage both on shelves and in one or more Automated Tape Libraries (ATL). The requirements for these systems are summarized in Table 1 for resources including drive bandwidths, total storage size and tape mounts per hour. For the data throughput, write bandwidths indicate data being entered into the system, and read bandwidths mean data being read from the system. The RIP numbers are for raw data originating from the DAQ. The number of tape mounts is large as we anticipate physically streaming data to tape. This may mean that, for example, 10 to 15 tapes might be appended to and exchanged among only 4 or 5 drives at any given time. The Farm category is for the reconstruction process and assumes raw data is being read in, processed and an output created which is half as large

Application	Write BW (MB/s)	Read BW (MB/s)	Size (TB)	Mounts/hr
R.I.P.	15	0	300	10*
Farm	7.5	15	118	4
Freight Train	5	70	12	12
Analysis-primary	1	13	12	45
Analysis-secondary	1	5	12	10
Thumbnail	1	1	12	0
Event Picking	1	38	65	240

Table 1: Approximate tape storage requirements for various delivery applications. The notation "write" means data to tape, and "read" means data from tape. Robot Mounts per hour assumes 50 GB cartridge size. Numbers are from [1] for "Season IV" with the exception of * which is a guess for the R.I.P tape mounts required, which is higher than farm mounts due to physical streaming considerations.

as the input. *freight train* processing includes production processing type activities with lots of input data, but little output. The category analysis-primary includes users selecting small samples of data from primary data sets, and analysis-secondary includes user analysis activities on *derived* or *secondary* data sets. Although *thumbnail* data will be primarily accessed from disk, there will be archival copies and backups stored on tape. The use of the robotic resources should be minimal most of the time, with some spikes of active usage. Pick events is anticipated to utilize much of any remaining resources. Although the bandwidths may not be as high as some other access modes, the random nature for the access may create many tape mounts and thus utilize a large portion of the robotic arm time.

7.2 Disk Buffers and Caches for Delivery to Processing and Analysis

In the initial phases, significant quantities (several TB) of on-line (disk) storage will be needed for raw and processed data to test and debug the detector and off-line processing. As the run matures, the system will need to include resources for long-term, on-line storage, where a summary of each event, or thumbnail, is maintained. Estimates for the size and bandwidth requirements of various delivery disk buffer and cache areas are show in Table 2. We use the term *buffer disk* to mean a resource used to match the feeds of two sets of I/O hardware, such as that from a streaming tape drive, to a processing system. For buffers the read and write bandwidths are the same. A *cache disk* is employed to store data which is anticipated to be needed by many users, and thus can reduce the number of tape mounts to get a set of heavily read files. For caches, the read bandwidths are higher than the writes.

The use of disk for various analysis applications includes some areas where the storage can be used efficiently, such as farm reconstruction and production processing, where the data can be marshalled through the processing chain very quickly. In cases like Analysis, thumbnail and event picking, disk will be used to cache data for longer periods of time, and this will reduce tape accesses. The category of unshared analysis, users or groups will be in charge of such areas, and the data may reside on the disks for long periods of time.

7.2.1 Disk Buffers for Importing Data to Tape Storage

Importing simply refers to the act of placing, or adding, data to tape storage in the system. There are many instances of areas where data enters the system including RIP, Farm processing,

Application	Write BW (MB/s)	Read BW (MB/s)	Size (TB)	Usage
Farm	15	15	0.2	Buffer
Freight Train	70	>> 70	0.2	Buffer/Cache
Analysis-shared	14	20	5	Cache
Analysis-unshared	5	5	5	Cache
Thumbnail	10	100	12	Cache
Event Picking	1	1	1	Cache

Table 2: Approximate disk Buffer/Cache requirements for various delivery applications.

Application	Write BW (MB/s)	Read BW (MB/s)	Size (TB)	Usage
R.I.P.	15	15	0.3	Buffer
Farm	5	5	0.2	Buffer
Production Processing	5	5	0.2	Buffer
Analysis	1	1	0.2	Buffer
Event Picking	1	1	0.2	Buffer
Monte Carlo	1.5	1.5	0.2	Buffer

Table 3: Approximate disk Buffer requirements for various import applications. The notations "write" means into the buffer and "read" means out of the buffer.

Production processing (freight train), Monte Carlo, and User analysis. The bandwidths required for each type of input can be estimated and these are shown in Table 3.

In each case, information needs to be supplied which allows the files to be shepherded to the correct location in the warehouse. This information includes target information, such as file family, as well as meta-data details which would might allow others to also successfully employ the data.

7.2.2 Data loss policy

There are different levels of tolerable data loss, depending on the source and nature of the data. Raw data is most critical, since it is not reproducible. The number of cartridges or cassettes on which the data resides is proportional to the percentage size of the stream. Our figure of merit has been .5% loss per year randomly distributed over the entire data set. As an example, in a sample of 1000 cartridges we would expect to lose 5 per year. If we are *exclusively* streaming data to streams of sizes (1) 50%, (2) 5%, and (3) 1%, any loss in stream (2) or (3) will create a proportionate amount of unusable data in stream (1), due to the way we track luminosities, et cetera. This means that if we lose 10% of (3), then 10% of each (1) and (2) are unusable. For this example there would be 500 tapes for stream (1), 50 for stream (2), and 10 stream (3). If we loose one cartridge from stream (1), it represents 1/500, or .2% of the data. One cartridge lost from stream (3) represents 10% of the data.

Figure 1 shows the calculated loss estimates for 100 TB and 1 PB raw data sets for various sized streams. We significantly reduce the risk of losing data by making duplicate copies of small streams. The chances of losing any given cartridge in any given year would be 1/200, and the chance of losing any 2 given cartridges would be, about, 1/40000. It appears, from the chart, that by keeping two copies of all streams smaller than 5%, we would limit data loss to less 1% for all streams, with the loss rate significantly less for duplicated streams. The chart also indicates that in the early stages of the run, when there is little data, we may desire to have more duplication, say for all streams below 10%.

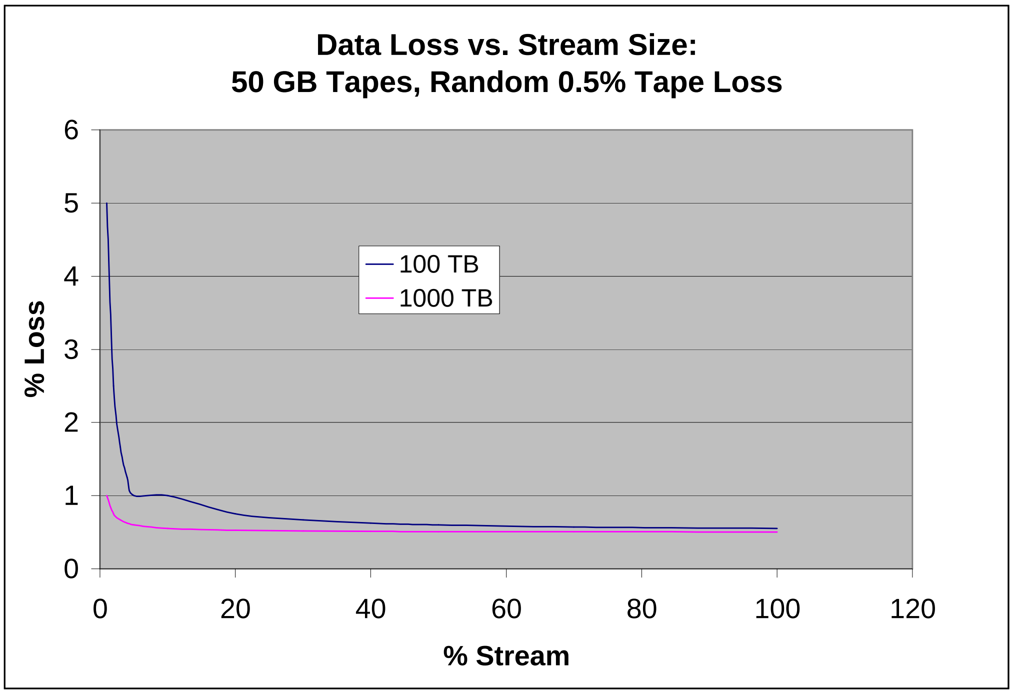

Figure 1: Estimated data loss for various stream sizes.

8 CDF specifications

This section is reserved as a placeholder for CDF to document both their agreements and differences with the Requirements listed in this document and to add additional detailed specifications which meet CDF data handling needs.

Since a decision on model of data access was made only recently there has not been sufficient time for an assessment of data handling requirements, in the light of the decision, to be fully developed.

References

[1] Lee Lueking, "Flows and Controls in the Seasons of SAM"

`http://www-d0.fnal.gov/~lueking/sam/five_seasons.ps`

[2] Drew Baden et. al., "CDF/CD/D0 Joint Data Management Needs Assessment Report", June 4, 1997, D0 Note 3197

`http://fncduh.fnal.gov:8080/workinggroups/runII/NAG/cdrun2needs.ps`

[3] Mike Diesburg, et.al., "D0 Reconstruction Input Pipeline System Requirements", October 1997 and Liz Buckley-Geer, et.al., "CDF Reconstruction Input Pipeline System Requirements", October 1997

`http://www-hppc.fnal.gov/rip/D0.txt` and
`http://www-hppc.fnal.gov/rip/CDF.txt`

[4] DRAFT Run II Mass Storage Requirements, June 1998.

`http://fncduh.fnal.gov:8080/workinggroups/runII/1998jun02/100bullet.txt`

[5] SAM Report to von Ruden Review panel, October 1997.

`http://www-d0/~lueking/sam/cdr2dma/data_access.ps`

A Glossary

The goal of this glossary is to define terms used in the Data Handling Projects for Run II, to provide a reference for checking consistency of use of these terms in the documents, and to provide a more precise description of these terms within the their context. There is an attempt to define these terms in relation to the physics analysis domain in which they are being used.

- **Analysis Group** - Group of physicists who organize to coordinate their access the same dataset or files for the purpose of analysis.
- **API** - Application Programming Interface. Set of callable routines through which access to event data can be controlled and data read or written.
- **Application Framework** - Software to whose API a user or physicists application code code is written, and under which the program is executed.
- **ATL/Robot** - Automated Tape Library where the mounting and dismounting of tapes in the tape drives is done without human intervention though an API.
- **Auditing** - Information gathered about the processes running and the parameters associated with them, the transactions executed, and the user requests queued and serviced.
- **Bookkeeping** - Stored information describing the production, location and history of files.
- **Buffer** - Temporary disk storage of data to smooth the flow of data when the instantaneous rate of the input and the output are different. The policy of what data is written to a buffer does not include the assumption that it will be reused - cf. Cache.
- **Cache** - disk storage used to store more frequently accessed data in order to reduce the latency and increase the throughput of access. The Cache reduces the effective access time to data stored on tape. Cached data, by definition, is expected to be a subset of the data that will be (frequently) reused and whose size allows it to be stored on the available disk.
- **Cache Policy** - decisions on how to manage the cache.
- **Catalog** Database of semi-static information about Events or Files that does not include the actual Event or File within it. A catalog does not necesarily include the location of the event or file.
  - **Event Catalog**
  - **File Catalog**
- **Consumer** - User Process that receives data from cache or tape.
- **DA** - Data Acquisition, collects the raw data from the detector
- **Data** - In the context of SAM data and event are used interchangeably. In general if the term Data is used it could additionally apply to calibration constants, meta-data that applies to more than a single event, detector or run conditions that apply to a whole run or sequence of runs etc.
- **Data Streams - Event Streams -** Organization of Events based on anticipated access patterns, such that access to the data is facilitated and can be achieved as fast and as efficiently as possible.

  - **Physical Data Streams -** Data Streams determined by the Trigger and Level3 Event information and physically located together on disk or tape.
  - **Logical Data Streams -** Data Streams determined by the expected selections and queries used to access the data (e.g 4 jet stream).

- **Database** - Records of information organized for rapid search and retrieval.

- **Database Management System** - (DBMS) A set of programs that control the organization, storage and retrieval of a database.

- **Dataset -** Collection of events that has a meaning from a Physics perspective. A dataset can be of varying types and sizes - e.g. Raw Dataset - collection of Raw Events; Top Dataset - collection of events that are Top Candidates; Run Dataset - collection of events from a single data acquisition Run; Run Ia Dataset - all events from Collider Run Ia; Calibration Dataset - related collection of calibration events.
  - **Primary Dataset -** A dataset output from the initial production or farm processing.
  - **Derived Dataset -** A dataset obtained from further processing of events after initial reconstruction on production systems.

- **Data Tier -** Event data is classified according to its content into tiers where each tier holds a particular subset of the whole event - e.g. Raw Event, Analysed Event. Lower tiers contain more information for each event - e.g. the RAW event data and higher tiers contain a refined level of event information - e.g. EDU50. The top most tier contains an event catalog.

- **Event -** Any information associated with a single Collision in the Experiment Detector for which data is read out and archived.
  - **Raw Event -** Event information as recorded from the online/level 3 system.
  - **Reconstructed Event -** Event information after processing by the reconstruction program and production systems.
  - **Analysed Event -** Event information output from one or more physics analyses being performed on the event.

- **Event Clustering -** see Data Clustering

- **Event Data Unit -** Event Data Unit (EDU) is data identified by a unique combination of Run Number, Event Number, Type of Event Data Unit. Any type of data record or data from any data tier, which can be read as a unit from some physical devices and is associated with a particular event.
  - **EDU50 -** Event Data Unit of size around 50KBytes.
  - **EDU250 -** Event Data Unit of size around 250KBytes. This is expected to be the size of a Raw Event.
  - **EDU5 -** Event data unit of size arund 5KBytes. This term is useful as it is expected that enough spinning disk will be available to store all Events if the size per event is around 5KBytes. It is expected that Thumbnails will be EDU5s.

- **Event Management System -** Hardware and Software System to manage and coordinate access to the meta-data and events.

- **Event Picking or Pick Event -** "Random access" user requests for particular events or selections of events. This mode of event/data access can result in vastly different patterns of access to the data than the Freight Train and Production data access modes. Not understanding the quantity of and need for this type of event access could result in constructing a system unable to deliver the needed throughput in this area.

- **Export** - Making tapes previously available through the ATL or Mass Store no longer available, and only available external to the system.

- **File** - Collection of bytes on disk or tape treated as a single unit and identified by a file name and physical location. A file may be a program, a document, a database, or some other collection of bytes.

- **File Family** - Collection of files distributed across more than one tape volumes and potentially written in parallel to more than one tape volume. Once a volume contains data from a file family, the rest of the volume will contain only files from that same family.

- **Freight Train -** Mechanism and process by which large datasets are processed in a coordinated fashion. A Freight Train consists of the organized delivery of a large dataset over a period of up to several weeks or months to multiple analysis programs. The data provided by a Freight Train is delivered at a rate determined by the data access and delivery system and not by the ability of the analysis programs to absorb and process it. Thus, a more cpu intensive analysis program may not receive all the data of the dataset in a single passage of the freight train. Mechanisms are in place to monitor which events/files are successfully accessed, and for resending those that were missed.

- **GUI -** In the context of SAM, the Graphical User Interface is imagined to provide integrated control and monitoring of the user requests and the processing system that will satisfy them.

- **HSM -** Heirachical Storage Management System. Provided by that Run II Joint Project. SAM is a user of the Run II HSM and does not construct or operate it.

- **Import** - Making tapes available through the ATL or Mass Store that were previously outside access through and control by the system.

- **Latency -** Time between request for data and the delivery of data to the requesting program.

- **Luminosity** - a measure of the number of potential proton/anti proton collisions per second. Used to normalize measured event rates to the total proton/anti-proton cross section or to each other.

- **Luminosity Block -** Time stamped record which records the luminosity for a time interval for each trigger.

- **Luminosity Record** - data which summarizes the luminosity integrated over a given time period. Different triggers may have different luminosities due to prescales and special beam conditions.

- **Map -** Relationship between information in an Event or File Catalog to the Physical Location or other attribute of an Event or File.

- **Meta-data -** Set of information used to describe Events, Files or Datasets for the purposes of allowing location of and access to the data, selection of subsets of the data etc.

- **Monitoring** - Gathering and presentation of information about the running system, the processes executing and the data flowing.

- **MSS** - Mass Storage System - system that manages a large amount (Terabytes or Petabytes) of data on tape and provides for writing and reading of this data. IN general an MSS includes an ATL, can include a hierachy of storage on Roboticly accessibly tapes, tapes residing on shelves and/or caching of the data on local or distributed disk. HPSS - A Mass Storage System in production use at Fermilab. Described at http://www.sdsc.edu/hpss/. Enstore - a Prototype mass storage system at Fermilab, based on the DESY OSM model. Will be described at http://hppc.fnal.gov/enstore.

- **On-Demand Data -** Data access mechanism for moderate-sized datasets to be processed by single users or small groups of users. Usually confined to a small set of files and used by analysis jobs that run in a few hours or days (depending on the total amount of data in the request). One important usage of this access method is the debugging of code for analysis jobs, such as for eventual use with Freight Trains.

- **Physics Data Stream - Physics Event Stream -** Collection of data, specified by the Trigger value. In D0 there will be up to 15 Physics Data streams.

- **Pick Event - see Event Picking**

- **Post-reconstruction analysis and production -** Processing done on a Primary or Derived Dataset.

- **Processing Unit -** Program that performs calculations on a Dataset.

- **Production Farm -** System of physically independent computing resources used in a coordinated fashion through the use of control software, for the reconstruction or analysis of event data. Typically Production Farms have been used for the processing of the Raw Events to produce output Data Summary Events, and are best suited for situations in which the "CPU used per event" is much larger than the I/O needs per event. Nowadays, production farms are used for reprocessing of data or "production analysis" where a large body of any Event Data must be processed by a single executable.

- **Project** - Organization of coordinated processing of files this will occur in a coordinated fashion such that the events are read only once from the storage management system and are delivered in parallel to all the analyses programs that want to access them.

- **Query Optimizer -** A program and/or business process that receives user requests and attempts to evaluate how long they will take and which h/w resources the request will require. It is planned to merge the requests from many users to optimize use of the resources.

- **Reconstruction** - Processing of the raw data, and associated calibration data, to make one or more derived data sets.

- **Request Manager -** (is this true?) Program that handles the user requests and allocates a "token" to authorize a program to access and obtain resources or data.

- **RIP** - Reconstruction Input Pipeline project which provides data from the data acquisition system buffers to the Mass Storage System, and from the Mass Storage System to the Production Farms.

- **Run** - Run I, Ia, Ib, IIa, IIb, etc identify periods of Collider operation - typically of months to a years duration. A data acquisition Run identifies a set of data acquired sequentially with a single set of calibration and detector constants. A Run is identified by an increasing Run Number which is typically stored in each Raw Event. The Run Number is used to identify temporally connected EventData. In stable detector and collider operation, Runs last between a few hours and a day.

  - **Physics Run** - Sequential set of events identified by a single number known as the "Run Number". In general, all events from a single physics run are associated with a single set of calibration data.

- **Run Conditions** - Parameters associated with a Physics Run e.g. Luminosity, state of the detector elements etc.

- **Shelf (Storage) -** Tapes not currently stored in the Automatic Tape Library or Robot. It is expected that the HSM system will provide mechanisms for moving tapes in both directions between Shelves and the Robot(s).

- **Station** - Computer system on which data processing and/or analysis is executed.

- **Storage Management Software (SMS)** Software to manage the Heirachical Storage Management System (HSM).

- **Thumbnail -** A Thumbnail (Event) is sufficient on which to perform most physics analyses. It is expected/hoped that the experiment can define the size of the Thumbnail such that the whole of the Run II dataset can be stored on attached disk.

- **Token -** A program requests and obtains a "token" in order to gain authorization to use a resource and access/retrieve data. Once the use of the resource is complete the program must return the token so that it may be allocated to another user/program.

- **Warehouse -** Sum of all the stored data.