Fermilab FERMILAB-Pub-01/238-E D0 Sept. 2001

The D0 Experiment Data Grid – SAM

Lee Lueking, Lauri Loebel-Carpenter, Wyatt Merritt, Carmenita Moore,

Ruth Pordes, Igor Terekhov, Sinisa Veseli

Matt Vranicar, Steve White, Vicky White

Fermilab, P.O. Box 500, Batavia, Illinois 60510[1]

Abstract. SAM (Sequential Access through Meta-data) is a data grid and data cataloging system developed for the D0 high energy physics (HEP) experiment at Fermilab. Since March 2001 , D0 has been acquiring data in real time from the detector and will archive up to 1/2 Petabyte a year of simulated, raw detector and processed physics data. SAM catalogs the event and calibration data, provides distributed file delivery and caching services, and manages the processing and analysis jobs for the hundreds of D0 collaborators around the world. The D0 applications are data-intensive and the physics analysis programs execute on the order of 1-1000 cpuseconds per 250KByte of data. SAM manages the transfer of data between the archival storage systems through the globally distributed disk caches and delivers the data files to the users batch and interactive jobs. Additionally, SAM handles the user job requests and execution scheduling, and manages the use of the available compute, storage and network resources to implement experiment resource allocation policies. D0 has been using early versions of the SAM system for two years for the management of the simulation and test data. The system is in production use with round the clock support. D0 is a participant in the Particle Physics Data Grid (PPDG) project. Aspects of the ongoing SAM developments are in collaboration with the computer science groups and other experiments on PPDG. The D0 emphasis is to develop the more sophisticated global grid job, resource management, authentication and information services needed to fully meet the needs of the experiment during the next 6 years of data taking and analysis.

1. Overview

During the past decade the D0[2] experiment processed and analyzed 30 Terabytes of data collected during ‘Run I’. The central analysis system provided data to the analysis programs from a shared file system of 300 Gigabytes, operator mounted tapes, and a traditional batch system which scheduled jobs based on the number of compute seconds required. Allocation of the disk and compute resources across the 20 or so Physics Analysis groups was done through meetings and administrative agreement. Several dedicated FTEs were required throughout the several years of data analysis to monitor, manage and coordinate the disk space, and control and schedule the analysis jobs. Data tapes were copied and sent through US mail to collaborators around the world. Lack of access to the ancillary calibration and run information, and the lack of ability to synchronize and coordinate the data sets and code versions, made the analysis of data at anywhere but Fermilab extremely difficult. The above constraints resulted in data delivery bottlenecks that were difficult to resolve and an inability to use efficiently the available compute resources.

D0’s second major data taking period – Run 2 - started earlier this year.The experiment will archive about 1/2 Petabyte of data a year. Run 2A will last for five or six years and is expected to be followed by Run 2B with up to an eight fold increase in data. The resulting Petabytes of data that must be processed and analyzed to produce publishable physics results consist of the raw data collected directly from the detector (about ½ the dataset), 100s of Terabytes of simulated data needed to understand and check the collected data, and successive levels of “refined data sets” of sizes ranging from a few Gigabytes to a few tens of Terabytes, which are analyzed many times over to understand and extract the final physics signals (and hopefully discover the Higgs!).
For the initial processing of the data, compute farms of about 20,000 SpecInt95s are deployed. For the analysis the physics groups will consume whatever resources are readily available and from which reproducible and defendable results can be obtained. Initially a minimum deployment of 30,000-100,000 SpecInt95s is planned[3] The collaboration of over 500 physicists is spread over 3 continents. Analysis facilities in England, the Netherlands and France are already being established, as well as at several universities in the US.
Given the experience of Run 1, the immediate challenges facing us for Run 2A and the anticipated needs for Run 2B – augmented by the simultaneous need for Run 2A analyses and Run 2B data processing - the D0 experiment has developed a fully distributed data handling and management system Sequential Access Using Meta-Data (SAM)[4]. While the system design and requirements were specified several years ago[5] SAM has been developed over the past few years as a production system to meet the immediate needs of the experiment for simulated data production, archiving and processing, and for development and testing of the final data acquisition and data processing systems. It has been in production use for over 2 years with robustness, automated recovery, and monitoring features built in from the beginning. SAM is used on many D0 systems at Fermilab[6] as well as handful of offsite institutions both in the US and abroad. The SAM requirements and design for remote data access, global disk caching and automated file replication, distributed job scheduling and resource management map well onto the current and ongoing Grid requirements and research activities. The D0 physics data processing and analysis applications are providing a good example from which to gain experience in the support of a production data grid and providing information for performance evaluation and tuning and management decisions.
Now that data taking has started and the initial data processing pipelines are in place and becoming stable, we are turning our efforts, as collaborators on the Particle Physics Data Grid[7] (PPDG) project to include research, design and implementation of the more sophisticated issues of data replication, global authentication, and job and resource management to meet the collaborations data analysis needs. Clearly, the continued and increasing use of SAM by the end users is giving us valuable feedback and input to the research and development directions. Aligned with the goals and strategies of PPDG, the D0 data distributed data management system provides an available deployment of end-to-end services ready to make use of new grid components as they become available.

2. D0 Applications: Distributed Analysis

D0 applications include data collection and archiving, data processing or reconstruction of physics triggered events from raw detector information, and event analysis. The analysis activities are organized as groups by physics topic. There are of the order of 10 such groups, each including several subgroups. D0 analysis applications process a collection or “dataset” of events selected by an pre-defined set of criteria. Generally, datasets are defined through have equal or related "physics trigger” or physics event type, or were originally acquired in a single sequence or “data taking run” or common detector configuration. The application is submitted to a batch system or run interactively. SAM delivers the files of the data set to the application asynchronously from the data consumption, in order to make optimal use of the available delivery resources (e.g. all files on the same tape before all files on another tape). Tunable parameters control the amount of disk cache available for the User Group running the application, the refresh algorithms for the disk cache, preventing the delivery of unneeded files if the application falls behind in the consumption of data already delivered etc. The D0 application infrastructure provides for the unpacking of data files to deliver event objects to the end user code.

A particular physics group will typically define common “reference” data sets and will run many applications whose data sets overlap or are in common. For large datasets – of the order of 10s of Terabytes – the physics groups will coordinate their applications and process the data files in parallel as the files are delivered to a local cache so as to increase the total analysis work achieved. Traversing a large dataset may require several weeks of computation, where only a subset of the files need to be available on local disk cache on any one day. An individual physicist may define many of her own datasets for algorithm development and individual research. Users typically reprocess small datasets many times over the course of a week or a month especially for the most refined datasets where physics parameter “cuts” are being made to provide the final histograms and physics results. While many of the analysis jobs run as efficiently as sequential executions, some of the applications are benefit from being parallelized and incurring the extra overhead of combining the generated results.

This paper presents an overview of the SAM system with pointers to the more technical details published elsewhere [8].

3. The SAM system

SAM includes a rich metadata catalog, a set of distributed servers which communicate via Corba interfaces, interfaces to storage management systems and to file transfer services, command line and application program interfaces to user applications and the catalog, an interface to batch systems, and a set of services to provide management and scheduling of the job and resource requests. The meta-data catalog (which is currently centralized but given its implementation in Oracle can be distributed as needed) tracks “all information about all things” and is continuously being extended with new information. To date it includes:

- The Replica Catalog, the location(s) of each data file on disk and/or tape, contents, generation and processing history, physics signatures or “tags”;
- Job Processing History, the information about each user job - time, duration, files processed, location, resource usage ;
- Collection Catalog, the definition and history of all datasets (collections of files).
- Ancillary Information Catalog, calibration and other bookkeeping information necessary to analyze each data file;
- Configuration Information, versions, environments, users, groups etc.
- Information Services - monitoring and status information about the files, jobs and users to allow analysis of the usage and performance of many aspects of the system.

Each collection of hardware (site) – locally distributed disk, cpu etc – in the globally distributed system is known as a “station” and managed and tracked by a “station master ” server. SAM manages the movement of data files to and from archival (tape or disk) storage to the station disk caches, as well as the lifetime of each file e.g. “pinned” by application request; or subject to cache replacement policies. SAM supports

multiple locations of any file, and provides for the efficient routing of files from existing locations to the disk cache local to the users application. Additionally the user interface includes features to automatically map logical file names to physical locations based on experiment rules and conventions.
The user interfaces for SAM include command line, C++ and Python APIs for user applications. Extensive interfaces to the database which support management of the users and infrastructure, definition of datasets through a “physics-aware”, domain specific, query language etc. D0 experimenters use SAM integrated into their traditional data processing and analysis programs; and make extensive use the definition and query capabilities in the coordination and management of the physics groups.

4. Technologies Used

All SAM client-server interfaces are defined in IDL and use Corba as the underlying infrastructure. We have been using Orbacus[9] until very recently. Recently we have started to investigate the use of OmniOrb[10] in an attempt to improve the performance when large amounts of data are transferred, and we are investigating some instabilities in the Orbacus nameserver. SAM servers are implemented in C++, with clients in C++, Python and Java. The data set definition interface and language is implemented in Java, C++ and python, using the TomCat[11] java servlet implementation. A sophisticated perl cgi-script is used as a general tool for web based database queries. There are many administrative scripts for system startup, shutdown, restart, status and error logging implemented in shell and python.
Access to the data is controlled through private protocols between the SAM clients and servers and user roles for access to the meta-data catalogs. As part of the Fermilab “strong authentication” project some parts of the system incorporates wrappers to integrate enable its use in a Kerberos realm.

5. Distributed Data Caching and File Delivery

SAM is responsible for providing transparent user access to data via file delivery and management over a globally distributed disk cache[12]. It includes generic interfaces to storage resources and file transfer services and has been interfaced to mass storage systems at Fermilab[13], IN2P3 and Nikhef. SAM supports file transfer via rcp, ftp and bbftp, with GridFTP next on the agenda. The central replica catalog keeps a full account of the multiple locations of the physical files and is used to determine the optimal location from which to retrieve a requested file.
When a Station Manager requests files, delivery may be done through multi-stage movement of the data between caches, or by copying of the file from another stations disk cache, if it is determined that this will provide the quickest access. Global and local configurations can be used to control algorithms to affect how file replication will occur. Each machine controlled by a station has locally mounted disks. When a file is authorized for transfer, the station’s cache manager allocates space from the available pool of local disk and issues requests to the associated station stager to deliver the files.
A station’s disk cache is allocated among the physics groups. If a cached file is requested by a second physics group any existing local replica is used. D0 applications in general have no dependence on the order of delivery of the files in a dataset and invariably are computing on the data of only one data file at a time. Thus SAM moves files in the most efficient order to achieve sequential delivery of the complete dataset. Once a file has been consumed by the users job it is eligible for deleting from the local cache if noone else is requesting to use it and as cache replacement policies dictate. Users can pin a file in cache on request, and subject to the policies of a particular physics group. From an application perspective the

management of the global disk cache is transparent. A user merely issues a request to "get_the_next_file" and when such is available a callback is executed.

6. Resource Management Research and Development

We are continuing to research and develop the global resource management and scheduling layer in SAM, in particular as mentioned above as part of PPDG. The end goals of the job control and data management are to ensure "fair" resource allocation by certain user categories, subject to the collaboration's policies, and maximization of the overall job throughput. In the data intensive world, both goals require approaches that fully incorporate the data management issues. To attain the first goal, the system uses fair share scheduling; for the second, the system performs resource co-allocation.

As discussed earlier, in general physics analysis applications are data-intensive. Given the data set sizes and storage capabilities for D0 Run 2 system, data delivery from tape is treated as expensive in terms of the number of tape mounts and tape reading time. Job scheduling decisions must be made depending on relative cost of data delivery and of increasing the job latency.. We must take account of such issues as use of tape robot arm, bandwidth of tape drive, and network bandwidth, which typically combine to result in the delivery of data being an "expensive" operation compared to the cost of delaying execution of the job such that it can be run using existing local data. The results of the comparison in cost of delivery vs delay may change as the execution of other applications may result in the delivery of useful data files. Additionally, since applications from the same physics group typically access the same sets of data, coscheduling of these user jobs may increase the total work achieved by the system.

We are approaching job and resource management in SAM using the concepts of "computational economy". Work done for a user application is understood in terms of benefit and the resources used in execution of this work incur a cost. Many different resources, with differing worth or value, will be used in the execution of a user job – tape mounts and tape reads, network transfer, local disk caches, cpu etc. The benefit to a user or physics group is defined by the files processed and takes account of the experiment policies and priorities for these groups, as negotiated by the collaboration. Given insufficient resources to achieve all requested analysis processing – a given in all HEP experiments to date - the goal is to get the most analysis done ensure the priority tasks are achieved with the necessary time, get as much analysis done as possible, and use the available resources as efficiently as feasible. Allocation of the available resources must be responsive to the sociological and political goals of the collaboration, as well as dynamic in responding to the prevailing and changing conditions. This is a challenging area of exploration in such a complex system. The SAM meta-data catalog, log file and information services (fed from instrumented servers) provide extensive information that allows us to track the performance of the system in detail, measure and analyse the results of our decision making and resource management models. Computational ecomonics is providing a good metaphor to allow us to define and manipulate our metrics quantitatively to enable decision making and actions. As an example, we are studying and balancing the relative costs of dispatching a job at a site with computing resources, but not data, available a priori versus queuing the job at a ``busy'' site where the data is available. Using a local distributed analysis cluster as an example, it is evident that the initial availability of (some) of the project's data at a machine is only one of the several factors to affect the job placement.

We plan to develop a formal language to allow us to describe the requirements and acceptable tradeoffs of a job and of the available resources. This PPDG work is planned as a collaboration with the University of Wisconsin Computer Science Department most notably with the Condor team[14]. The Job and Resource Definition Language must have rich enough semantics to communicate the full vocabulary needed to support the dynamic scheduling, flow control, and monitoring of tasks specified by job control or ad hoc analysis requirements across the globally distributed system. Its function is to allow the job management system to coordinate the allocation and management of compute resources, communication resources and storage resources to match the needs of the large number of simultaneous clients and to follow local and global policies. We are working with the Condor group to apply and extend the ClassAd language[15] and the matchmaking framework to meet these needs.

7. Conclusion

As data pours in, the collective needs of the D0 analysis jobs will far outstrip the available resources, especially as the physicists are freed from detector commissioning and start mining the vast data sets for new physical phenomena. We anticipate that the complexity of the events collected for Run 2 will result in more computationally intensive applications as time progresses. There is anecdotal evidence from Run 1 that such new analyses were attempted. If we can solve the issues of configuration management , versioning and tracking of all parameters, the experimenters will feel more comfortable recalculating intermediate results from re-delivered data files, and not storing all intermediate datasets.

As D0 experimenters return to their home institutions having completed commissioning of the detector, the need for PhD theses increases, and the number of analyses explodes, SAM will be an essential component in the experiments ability to deliver timely and reliable physics analyses and results. It will be relied on to provide fast, robust and complete data access, information cataloging and management of the computational resources. We can already see features that will become useful as the overall load of the system increases. As an example, since resources required and available are captured and the job execution scenario determined, it is possible to return to the user an estimate of the length of time to scheduling or completion of the job. Although this is an area of ongoing research and not a service we plan to incorporate soon, we decided that we will design every component of the SAM system to support estimation of the services they provide.

The design of SAM allows us to extend and increase the sophistication of the services it offers as we continue with our research and development as part of PPDG. Working with the Computer Science group at the University of Wisconsin we enable us to take advantage of their research and experience and deliver a system which can be used as a model and test bed for other high energy physics experiments data handling.

8. Acknowledgements

This work is done as part of the SAM and the D0 Particle Physics Data Grid project. We acknowledge input and support from many people in the Computing Division at Fermilab and the D0 Collaboration.

References

1. This work is sponsored by DOE contract No. DE-AC02-76CH03000
2. http://www-d0.fnal.gov - the D0 experiment.
3. http://runiicomputing.fnal.gov/runIIdocs/NAG/cdrun2needs.ps CDF/D0/CD Run II Data Management Needs Assessment Document

4. SAM Overview and Operation at the D0 Experiment", submitted to The International Conference on Computing in High Energy and Nuclear Physics (CHEP 2001), September, 2001. home page: http://d0db.fnal.gov/sam
5. http://d0db.fnal.gov/sam/doc/requirements/sam_vw_req_arch.ps - Requirements for the Sequential Access Model data access system, J. Bakken et al
6. ~20 stations listed at http://d0db.fnal.gov/sam_data_browsing/
7. Particle Physics Data Grid Proposal – www.ppdg.net/docs/scidac01_ppdg_public.pdf
8. SAM overview and operations at the D0 Experiment, Resource Management in SAM - the D0 Data Grid to be presented at the Computing in High Energy and Nuclear Physics conference, September 3 - 7, 2001 Beijing, P.R.China
9. http://www.ooc.com/ob/
10. http://www.uk.research.att.com/omniORB/omniORB.html
11. http://jakarta.apache.org/tomcat/
12. SAM for D0 - a fully distributed data access system", I. Terekhov et al, presented at VII International Workshop on Advanced Computing and Analysis Techniques in Physics Research (ACAT 2000), Oct. 2000, Batavia, IL, in proceedings.
13. Enstore mass storage system http://isd.fnal.gov/enstore/; SARA mass storage system http://www.sara.nl/; IN2P3 HPSS http://doc.in2p3.fr/hpss/;
14. http://www.cs.wisc.edu/condor/ - Condor High Throughput Computing
15. http://www.cs.wisc.edu/condor/classad/