

Particle Physics Data Grid Quarterly Status Report

January - March 2001

16 May 2001

Contents

Introduction.....	2
PPDG Organization and workplan – excerpt from SciDAC proposal	2
Development and Deployment Process	2
Computer Science Program of Work.....	3
CS-1 Job Description Language	4
CS-2 Scheduling and management of processing and data placement activities	4
CS-3 Monitoring and status reporting.....	5
CS-4 Storage resource management	5
CS-5 Reliable replica management service	7
CS-6 File transfer services	7
CS-7 Collect and document current experiment practices and potential generalizations	7
HENP Experiments Program of Work.....	8
The ATLAS Experiment.....	8
The BaBar Experiment.....	10
The CMS Experiment	11
D0 Experiment	13
The STAR Experiment.....	16
Thomas Jefferson National Accelerator Facility Experiments	17
Work Plan & Schedule	18
Ongoing Activities.....	18
Project Activities	19
PPDG Management Plan	21
List of Participants.....	23
Reports from PPDG Teams.....	24
ATLAS	24
CMS	26
D0.....	32
LBNL Scientific Data Management Group.....	34

Introduction

The major activity in Q1 CY01 is the development of the program of work and organizational plan that was put into the SciDAC proposal. The relevant sections from the proposal are included in this report. The principle features are that *teams* of people are identified for each of the physics experiments and each of the Computer Science projects, and the work is organized around collaborative *project activities*. Each project activity includes a physics team and a CS team working to develop and deploy grid technology as a production service for the physics experiment.

In addition to the general organizational plan, this report includes descriptions of technical work from some of the physics and CS teams, which are included following the section from the proposal. Future status reports will consist primarily of reports from each team describing work on the project activities.

PPDG Organization and workplan – excerpt from SciDAC proposal

Now that Data Grid concepts are maturing and a broad interest and level of activity has developed around the world¹, especially in Europe, for implementing Data Grid ideas, the PPDG collaboration is transitioning to a role of moving Data Grid services into production environments in the high-energy and nuclear physics community. PPDG developments are scoped to be compatible with and follow the guidelines of the Data Grid Reference Architecture² as developed by the GriPhyN project. We have defined a development process, described below, that combine the programs of work for the Computer Science and experiments teams. PPDG activities are executed jointly between members of the Computer Science teams and the experiment teams where the activity will result in a deployed Grid enabled capability. During and at the completion of each activity attention will be paid to opportunities for reuse and generalization.

Development and Deployment Process

The development process described here is driven by the vision and belief that the continued coordinated search for commonality derived from actual practice will deliver the most effective solutions in an efficient way. It is founded on previous experience, including the Experiment-CS collaboration model of the Grand Challenge project that developed STACS as a service integrated with the STAR experiment.

The PPDG development work will be structured as Experiment-CS Project Activities. Many of the experiment work plans have commonality both in scope and schedule. Some Project Activities will be targeted to meet the need of one specific Experiment-CS deliverable, with coordination and cooperation between concurrent activities occurring through the PPDG management structure. In other cases the experiment and computer scientist teams will agree on a common activity, which will still follow the PPDG development model. Each PPDG Project Activity will be led by a two person team:

- the experiment(s) will appoint a representative that will provide overall coordination and prioritization of the experiments needs
- a Computer Scientist who will identify applicable Grid technologies and coordinate the involvement of one or more CS projects in the activity

Together, these two individuals will produce and track a project plan for the activity, report on its progress and be responsible to the PPDG management for its schedule and delivery.

The PPDG Collaboratory Pilot will manage, across all activities, an electronic communication forum that includes web sites, mailing lists, time-lines and document libraries. This will help all parties of the collaboration, spread over the globe, to identify issues and problems that are common to multiple experiments and conducive to common solutions. A key function of the ppdg.net repository will be the

¹ www.gridforum.org

² <http://www.ppdg.net/archives/ppdg/2001/msg00069.html>

creation of a knowledge base that gathers, into one place, a summary of current practice among each of the experiments, in the areas of data management, data transfer, job management, metadata management, etc. This repository will involve building web documents that contains links to live or copied versions of each experiment's own knowledge bases. Regularly scheduled teleconferences and occasional in-person meetings will be scheduled across all groups to maintain group dynamics across the different applications and providers, and to facilitate common solutions. In a way, we envision PPDG to operate like a Technology Grid where consumers and providers of technologies can find each other and join in collaborative research and development activities.

During the course of the proposed three-year pilot, we will go through several development cycles in which we leverage emerging Grid technologies to deliver ever increasing levels of functionality with increasingly common technology and process solutions.

An integral part of our software development cycle will be careful design and implementation of error handling mechanisms within and across modules, writing documentation, and development of a software support and maintenance program. We expect that the experiments will operate a support infrastructure that will provide the user and operational support (tier-1 support) services to their scientists. Each of the CS projects will maintain its own support infrastructure that will provide a second layer of support (tier-2 support) for software developed by the project. This support will be accessible to scientists via designated personnel at each experiment or any HENP-wide support framework. We will be open to approaches from commercial vendors to provide or develop contracted support programs. HENP specific software may be also supported through the Grids in Particle Physics Support Team, GriPPS³, currently under review by DOE. Upgrade of the software for new operating systems or environments will be the responsibility of the support group. .

Feedback and refined requirements disseminated by the PPDG effort will have the beneficial effect of focusing the out-year research and development efforts of all applicable SciDAC middleware projects, whose results can then be leveraged by the PPDG activities and deployed by the experiments. More detail on the work plan phases and schedule are given in a later section.

Computer Science Program of Work

Following a careful analysis of the integration requirements and milestones of the different HENP experiments we identified seven Computer Science research and development areas. These areas reflect the experience we gained from the first phase of PPDG. The CS teams in PPDG are organized to address the challenges in these focus areas. Each of the first six foci cover a CS area that will provide Grid enabled components to several experiments. The seventh focuses on the continuing the collection of information on experiments' practices and needs.

- CS-1. Job Description Language
- CS-2. Scheduling and management of processing and data placement activities.
- CS-3. Monitoring and status reporting
- CS-4. Storage resource management
- CS-5. Reliable replica management services
- CS-6. File transfer services
- CS-7. Collect and document current experiment practices and potential generalizations

The labels (CS-1 ... CS-7) will be referenced in the following descriptions of work.

³ Grids in Particle Physics Support Team (GriPPS) <http://home.fnal.gov/~ggraham/GriPPS/index.html>

CS-1 Job Description Language

The ability of a resource management system to effectively and efficiently manage the execution of a job depends to a large extent on what the system knows about the job. The more the system knows the easier it is for the system to plan and coordinate the execution of the job. This is especially true in a Grid environment where a clean separation between the logical requirements of the job and the physical resources used to meet them holds the key to delivering seamless services. The success of Database Management Systems (DBMS) is a clear display of the power of such a separation. Queries expressed in a logical model of the data are interpreted, optimized, and mapped by the DBMS to physical operations on the stored data. Job Description Languages (JDL) are the means by which job owners communicate the resource requirements and structure of their jobs to the resource management system they entrust to execute the job. Most batch systems like LSF and PBS, and Grid resource managers like Globus use JDLs with limited capabilities. Our work in this CS area will leverage our experience with the ClassAd language and the matchmaking framework we have been using in Condor to develop a JDL powerful and flexible enough to meet data processing and analysis needs of the HENP community in a Grid environment.

HENP jobs may consist of many tasks with complex interdependencies. A job can be viewed as Directed Acyclic Graph where nodes represent tasks and arcs inter-task dependencies. Conditions may be set on when and if the completion of one task should trigger the execution of another task. These conditions may involve complex logical dependencies between what the completing task did, the overall state of the system and what the next task needs to do. Each task may have its own set of requirements and preferences for hardware, software and data resources. In addition to CPU and memory resources, a task is likely to need storage resources to store input and output data and I/O and or networking resources to access the data stored on this resources. The type and/or size of the resources allocated to a task may determine the executable and data it uses. Today most of this information is defined procedurally by a script file and is therefore not visible to the resource manager. The script captures the steps of the job, the interdependency between the steps, the data to be accessed or created by the job and many other aspects of the job that if available in a declarative form to the resource manager can be used to provide seamless and efficient services in a Grid environment.

In the same way that Globus implemented capabilities to map jobs described in the RSL to the Job Description Languages used by PBS, LSF, Condor and others, so it will be our responsibility to do so for the JDL we will develop and the scheduling and resources management systems (for example SAM) used by the experiments. On the application and user interface side we will have to figure out how to map what we or the user knows about the work to be done to a Directed Acyclic Graph described by the new language. If we are successful with this task and reach one day the point where all experiments use the same language to describe their jobs, we may even reach the point where a job submitted by a scientist in experiment A will be seamlessly served by a computing facility owned by experiments B.

CS-2 Scheduling and management of processing and data placement activities

Most of the processing and data placement jobs to be triggered by a Physics experiment are asynchronous. A typical Grid enabled HENP job expects to experience queuing delays in acquiring resources, is likely to take a long time to execute and may have to be restarted due to hardware or software failures. A user or an application submitting such a job expects an asynchronous notification to be delivered when the job has terminated, regardless of whether it takes a minute, an hour, a day or a week to complete the job. While efficient execution of the job is clearly desirable goal; not losing the job is a must.

The focus of this CS working area will be the development and implementation of modules that provide robust and efficient job control services. We will build on our experience in building the basic job management infrastructure of the Grid enabled version of Condor and user level job control capabilities like the Directed Acyclic Graph Manager (DAGMan) that has been used to manage CMS simulation runs and the Request Executor (ReqEx) that was developed as part of the PPDG testbed. These modules will be interfaced with experiment specific applications, other batch systems already in use such as LSF, PBS, and resource management fabric as part of the planned integration activities. At first we plan to devote most of our effort to the development of reliable and recoverable job submission and resource reservation protocols,

notification services, event logging capabilities and repositories for job and resource allocation information. Once these protocols, services and capabilities are in place, we will address issues related to the efficient execution of these jobs including: job decomposition and partitioning; data query, placement and reclustering strategies; and sub-file level data access. We will explore “smarter” algorithms and policies to plan and schedule the execution of jobs. We will leverage our experience in developing and evaluating distributed scheduling policies

CS-3 Monitoring and status reporting

No distributed system can deliver efficient 24-7 services without a reliable monitoring subsystem. Therefore, collection, storage and presentation of monitoring information will be an essential part of any end-to-end Grid enabled capability that will be developed by the PPDG Collaboratory Pilot. While most Grid efforts view timely, complete, historical and well structured status information as an essential part of resource allocation, job management, and data placement strategies, our work in this CS area will also address the use of this information to detect hardware and software faults and to trouble-shoot Grid applications. Tools to analyze error and state information and to facilitate problem tracing and resolution are essential for the development, deployment and maintenance of robust production systems such as those required by HENP experiments. The seamless nature of Grid services and the autonomy of its resources make trouble-shooting extremely difficult and therefore require powerful tools and reliable information.

Most fabric and applications already in use by members of PPDG include some monitoring and status reporting capabilities (e.g. Globus GRIS and GIS and the collector and log files of Condor). As part of each of our integration activities we will work on extending these capabilities, ensure that information is reliably deposited in persistent repositories and make the recorded information accessible via Application Programming Interfaces (APIs) and Graphical User Interfaces (GUIs). We will work with applications and resource managers to extend and harden their existing monitoring and reporting capabilities and will develop portable and extensible information repositories. Netlogger and the Network Weather Service are two examples of tools that we plan to incorporate in the PPDG deliverable. The output of these tools will be transformed as needed and transferred via common protocols to the information repositories.

Special attention will be devoted to leveraging the power of semi-structured data representations provided by languages like XML and ClassAds to deliver, store and display status information. We will carefully evaluate the potential of the SRB knowledge management system implementation based on Mediation of Information using XML (MIX). The possibility of using triggers as a mechanism to alert applications on special events will be studied and the power of the matchmaking framework of Condor to provide such a mechanism will be evaluated.

In the initial phase of the developments we will adapt existing GUIs and application information display systems to accommodate the new capabilities, transfer protocols and data representation languages. As we gain experience in this area and subject to the availability of funds for GUI development we will investigate, adopt or develop a common graphical display system of status information for our Collaboratory Pilot.

CS-4 Storage resource management

Mass storage access and management

Data intensive applications, such as the experiments in PPDG, place extreme demands on mass storage systems (MSS). For example, requests for hundreds of files from several users overwhelm the ability of a system like HPSS to serve simultaneously. The response is typically a refusal to serve the client, thus forcing the client to repeatedly request the same file until the mass storage system responds. In principal, mass storage systems can be enhanced to queue the requests they cannot serve at the time, and enforce policies of how to serve the requested files. This is a complex task that is normally thought of as being outside the realm of MSS. Even if an MSS provided such a service, there are advantages to using a staging disk outside of the MSS that can be shared dynamically by the users. This is the approach taken by the Hierarchical Storage Resource Manager (HRM) middleware layer. It provides the management of a file request queue and a staging disk to provide the following functionality: 1) if the MSS is busy, file requests are queued, not refused; 2) by using advance knowledge of files requested by multiple users, it provides

files to users in an order that maximizes access from disk, thus minimizing repeated file access from tape; 3) it can reorder file access to maximize files read from the same tape, thus minimizing tape mounts; 4) it insulates the client from temporary failures of the MSS, by resuming file transfer requests when the MSS recovers; and 5) it provides status information on the length of time till a file will be staged.

HRM currently manages requests to get files out of the MSS. It will be further developed to have additional features as proposed in a SciDAC middleware proposal, called “Storage Resource Management for Data Grid Applications”. The enhancements include: 1) support for policy management; 2) support for a write capability into the MSS; and 3) support for space reservation of the staging disk. We plan to use these capabilities as they become available. The tasks that will be performed in the context of this proposal fall into 2 categories. The first is the deployment and adaptation of the HRM technology in PPDG experiments, including extensions to a Java application environment. Deployment includes the installation of the system, and verifying its correct behavior. Adaptation includes modifications to interfaces of the middleware software to work in an environment that may have a different variation of operating system or the MSS. One of the long-term goals of the HRM adaptation is to use the same basic software with an MSS other than HPSS, such as Castor developed at CERN. The second category of tasks involves providing interoperability between HRM and the other middleware components. One of our main goals of interoperability is the ability for a Globus GridFTP server to call HRM for pre-staging of a file before it is transferred by the GridFTP service. This will provide file sharing as well as access to different HRMs in a uniform way. From the GridFTP API, a file access from HRM or a disk will be identical.

Caching and staging services

For PPDG applications, the use of large shared disk caches is essential for several stages of data production and analysis. In the process of simulating event data, computation farms are scheduled, and their output needs to be staged to a temporary disk cache before being reconstructed and archived. Similarly, a temporary disk cache is needed during the collection of the experiment data. In the reconstruction phase, files may reside on a disk cache, or need to be brought from tape to a disk cache for processing. During the analysis stage, temporary (usually shared by multiple users) disk cache is needed to cache files for the analysis programs. For these reasons the Disk Resource Manager (DRM) is an important middleware service.

We see the use of a disk cache in two modes: 1) reservation-based usage; and 2) on-demand access usage. The “reservation-based” DRM is mostly needed for co-scheduling of resources in the data production and reconstructions phases, and the “on-demand” DRM is mostly for dynamic use of caches for the analysis process. For the purpose of the analysis, usually only a subset of the data needs to be accessed, and the access patterns often requires repeated access to so-called “hot files”. We plan to support both usage modes. Typically, a disk cache will be used in one mode or another, although in the long term we plan to explore the possibility of accessing a disk cache for both modes simultaneously. For our work in this proposal, we plan to take advantage of developments planned as part of the SciDAC SRM middleware proposal, called “Storage Resource Management for Data Grid Applications”. However, we note that the emphasis in that proposal is on the “on-demand” usage. Thus, we propose to develop as part of this proposal a “reservation-based” DRM. This will include the ability to negotiate a reservation of space usage for a window of time, specified as start time and duration. A second stage of the development will include an allocation capability that is based on pre-specified policy. As with HRM, one of the important tasks planned under this proposal is the deployment and adaptation of the DRM technology in PPDG experiments.

Storage Resource Broker

To ensure that our main choice of Globus based middleware technology does not limit the generality of our work, we include in our proposal a task to use another middleware technology – SRB. The SRB, or Storage Resource Broker, provides a uniform access interface to file systems, archives, and databases. The system supports replicas, aggregation of files in containers, and organization of distributed files into logical collections. The MCAT, Meta data Catalog, provides a mechanism for storing and querying system-level and domain-dependent metadata, including support for extensible catalogs, data set discovery mechanisms, and export of metadata as an XML DTD. Both technologies offer some advantages. Globus takes a layered approach to middleware, providing access to basic services, such as GridFTP and a replica catalog, and builds additional services on these for replica management. In SRB, services are organized with

respect to a collection. Our challenge is to show that the middleware software we are developing in the areas of job control, coordinated resource planning, and storage resource management work with both Globus and SRB.

To achieve this, we plan to provide similar functionality in both approaches. In particular, we plan to demonstrate that applications accessing files can do so by interacting with the SRB client in addition to Globus services, specifically for the STAR/STACS deployment. Furthermore, we will continue to work with SRB developers, so that storage resource management capabilities developed by HRM and DRM are also available through the use of SRB.

CS-5 Reliable replica management service

The file distribution service provided as part of job management requires a reliable replication management service that can move collections of files from point to point(s) with transactional integrity within the application's wide area network. While this service encapsulates the mechanics of reliable replication, it also provides its clients with the mechanisms they require to make their own guarantees of operation integrity, recovery, and cleanup. In case of failures this service needs to remove partial files from the destination and/or re-transfer files or file segments as necessary. Replication actions will be performed with transactional integrity and synchronized with space allocation and access permission checking.

Reliable replication interacts closely with the file transfer mechanism to set up transfers that efficiently utilize the network available to its clients under local and global policies and dynamic conditions.

CS-6 File transfer services

The most basic service a Data Grid provides is a data transfer service. Different layers in the software stack of a Grid-enabled application will use such a service for different purposes; some will use it for replicating data across sites while other will rely on it to stage data in preparation for running a data analysis task. Simulated events will be moved via this service to an archival site while cache managers will employ it to move data to close by caches in order to reduce I/O latency. The Globus project is developing GridFTP specifically for data transfer services on the Grid as part of the SciDAC Data Grid Toolkit Middleware proposal. Therefore, one CS focus of our Collaboratory Pilot is the interfacing GridFTP with the applications and fabric of the HENP experiments.

GridFTP encompasses a protocol, API, and service that provide efficient reliable data transfer over one or more TCP/IP connections. When transferring data it can use multiple TCP connections in parallel between the same source and destination, to achieve high throughput on a single wide-area IP route. TCP buffers and window sizes are automatically negotiated. Bandwidth can be further improved by striped or interleaved transfers across multiple servers. Users who request GridFTP services can be authenticated via Grid Security Infrastructure services or Kerberos. They can ask for partial file transfers or initiate fully authenticated third-part transfers.

Using GridFTP we plan to develop and implement a reliable end-to-end data transfer mechanism capable of controlling and coordinating the consumption of Grid resources. In a multi user environment like a Data Grid, many concurrent FTP transfers or other applications may share storage space, buffer space and communication bandwidth. Each element of the Grid must operate within the resource usage constraints as defined by higher-level resource managers. As the availability of resources changes during the transfer due changes in the load or failures, that data transfer mechanism has to be able to dynamically adjust to such events and recover,

CS-7 Collect and document current experiment practices and potential generalizations

Further meetings will be held with the experiments to better understand how they now support or plan to support the requirements above. Information to be gathered includes: a) the volume of data; what data is replicated – when and how; b) movement of data for analysis – when and how; c) experiment monitoring,

failure detection and recovery; d) job control; e) storage (disk, tape), processing (CPUs) and network resource management scenarios; and f) how replica catalogs are implemented.

This activity will be repeated annually to understand how the needs have evolved, and how the deliverables and activities already complete and underway might be adapted to other experiments, made more general, or submitted as a common component. This work area will also be used to identify new needs that will arise as a result of the ongoing work. Reports will be written documenting and synthesizing the information.

HENP Experiments Program of Work

The program of work listed below clearly exceeds the resources available to the PPDG project to deliver alone. The list of Grid services needing to be developed for each experiment is included here to show the commonality of requirements between the experiments and the overlap of these requirements with the Computer Science work areas listed above. We will work together with other Grid projects – e.g. GriPhyN, EU DataGrid - and the larger data handling groups of the experiments themselves to ensure that there is a sensible division of responsibilities, no overlap in the work being done, and that the PPDG Activities fit into the agreed upon Grid architecture.

The ATLAS Experiment

Distributed computing in ATLAS

While LHC data taking will not begin until 2006, ATLAS already has a large and highly distributed computing and software operation serving immediate and near term needs such as test beam data analysis, detector performance and physics studies supporting detector design and optimization, software development and associated scalability studies, and “Data Challenges” involving high-throughput, high-volume stress tests of offline processing software and facilities.

ATLAS is currently transitioning from FORTRAN-based software used for past production operations to new object oriented C++ software in which the U.S. has a leading role in architectural design and infrastructure development. In databases and data management, again with a leading U.S. presence, ATLAS has recently begun ramping up a development program to build the full ATLAS system. The U.S. has selected these focus areas as those best matched to U.S. expertise and to most effectively enabling U.S. physics analysis participation. Our PPDG program will provide an important complement to these efforts in providing a powerful and well integrated distributed computing capability that will empower physicists in performing analysis work at their home institutes.

Associated major ATLAS milestones

- Calorimeter test beam analysis: Summer/Fall 2001
- ATLAS physics workshop: Sep 2001
- Mock Data Challenge 1: Feb - Jul 2002, 1% scale
- Computing TDR preparation: May - Nov 2002
- Trigger/Data Acquisition TDR: Summer 2002
- Mock Data Challenge 2: Jan - Sep 2003, 10% scale
- Physics Readiness Report: Jan – Jun 2004
- Full chain test: Jul 2004
- 20% Processing farm prototype: Dec 2004

Grid Services in ATLAS

Year 1: Production distributed data service – CS-1, CS-2, CS-3, CS-4, CS-5, CS-6

The distributed data services described here are to exist between CERN, the Tier 1 Facility at BNL, and a few other U.S. institutes. Likely early participants are ANL, LBNL, Boston U, Indiana U, and U Michigan. The objective is a multi-point U.S. Grid (in addition to the CERN link) providing distributed data services as early as possible.

- Distributed file and replica catalogs (CS-5)
 - Cataloging files resident on disk and in mass storage. Based on logical names in an agreed global Grid namespace. Serving read-only files at CERN, the Tier 1 at BNL, and select U.S. institutes. Supported by web, shell and API tools to browse, query and manage catalogs.
- Deployment of data transport services in production, serving the distributed data service (CS-6)
- Production distributed data service providing managed distribution and replication of data files (CS-4, CS-5, CS-6)
 - User initiated data replication via web or command line. Simple replica cache management. Service is integrated with hierarchical mass storage: HPSS system at the BNL Tier 1; collaboration with CERN and EU DataGrid for CERN mass store support.
- Remote job submission service testbed (in use by developers and 'friendly users') (CS-1, CS-2)
 - Supports remote submission of jobs to the Tier 1 from a number of external centers. Requires limited deployment of distributed authentication services.
- Grid instrumentation service: basic monitoring of distributed data service (CS-3)
 - Monitors activity, usage, performance, and error conditions. Web based display and reporting tools.
- ATLAS “data signature” design supporting Grid requirements in place. Prototyping begun.
 - A specification of data set content and characteristics complete enough to, in principle, regenerate it. Used in tests of dataset equivalence, current validity, etc.
- U.S. ATLAS distributed computing services architecture: requirements gathered, design begun. (CS-7)

Year 2: Production distributed job submission service – all CS areas

The major new functionality to be delivered in year 2 is a distributed job submission service.

- Broad deployment of user-level Grid authentication in support of “Grid user” based functionality of production remote job submission service and enhanced distributed data service. (CS-1, CS-2)
- Distributed data service enhancements (CS-4, CS-5, CS-6)
 - Extend service to several additional U.S. institutes. Integrate replica management into core database software supporting production. Add replica cache management service. Incorporate user-level storage resource reservation. Add cost estimation service. Provide C++ and Java APIs for catalog services. Automated, event-driven (e.g. “update available” signal) replica updating implemented to ensure consistency across file instances.
- Production remote job submission service (CS-1, CS-2)
 - Extension of simple year 1 system to production use by the general community; requires full deployment of user-level Grid authentication.

- Extension of Grid instrumentation service (CS-3)
 - Full monitoring of distributed data service and basic monitoring of job submission service. Improved display and reporting tools.

Year 3: Transparent distributed processing services – all CS areas

The principal goal in year 3 is to enhance the distributed data and processing services with user interfaces, specification languages, and ATLAS infrastructure integration to provide transparency to the user of the distributed nature of processing and analysis, both in 'batch' and interactively.

- Production distributed processing service (CS-1, CS-2, CS-3)
 - Extension of remote job submission service to provide transparency (to the distributed nature of the processing) to the ATLAS offline analysis user. Distributed services integrated directly into ATLAS software infrastructure.
- Integrated distributed data management services (CS-1, CS-2, CS-3, CS-4, CS-5, CS-6)
 - Integration of distributed data services with ATLAS database and data management infrastructure to provide (policy-constrained) user transparency to data locality. Cost estimators supporting policy control integrated. Run, event, and event feature (tag) metadata integrated with PPDG catalogs
- ATLAS “data signature” deployed in support of coherence/consistency of file replicas and transparency in data set requests.

The BaBar Experiment

Status

In late 2000, the BaBar collaboration proposed a “computing model” to address the rising need for data analysis capabilities and the need to achieve maximal involvement of all collaborators. The model calls for the establishment of one or more Tier-A computing centers that, like SLAC, will offer data analysis facilities to all BaBar collaborators. Each Tier-A center will focus on offering analysis facilities for a “vertically integrated” (raw data plus all derived data products) fraction of the total data set. During 2001, the Lyon computer center of IN2P3 will begin to function as a BaBar Tier-A center. Grid services supporting rapid and reliable data transfer are urgently needed to support this function. In the slightly longer term as the division of data between the Tier-As becomes more complex, job management that will automatically dispatch an analysis sub-task to the correct Tier-A center and bring output back to the submitter will become essential. In providing these needs, PPDG will collaborate with the staff of the IN2P3 computer center and with French members of the EU DataGrid project.

The creation of additional BaBar Tier-A centers is under discussion, currently most strongly focused on a center at RAL in the UK. An additional center would increase the urgency of the need for automation and robustness in both data transfer and job management.

BaBar is already using data-transfer utilities developed by the collaboration to distribute data to “Tier-B” centers supporting the analysis of a few percent of the data by a relatively small user community. These tools are too labor-intensive for long-term viability, but they allow a strategy of Grid middleware deployment focusing first on areas where the existing tools are weakest, and where CS effort is available to collaborate on the deployment. A Globus-SLAC collaborative effort on early deployment of the Globus Replica Catalog has begun. BaBar intends to focus its first PPDG efforts on the deployment of robust and functional Globus-based replica catalog services in support of SLAC-Lyon transfers.

The next focus of BaBar-PPDG will be the introduction of early Globus and BaBar tools using re-try and integrity checking to increase the reliability of transfers. This will be complemented by the development of (largely) BaBar-specific tools to automated the selection of files to be transferred, and deployment of (largely) generic tools to discover and request storage and network capacity and schedule transfers

accordingly. By early 2003, PPDG will enable the automated exchange of a terabyte a day (thousands of files per day) between SLAC and the European Tier-A centers.

In 2001, physicists will be required to determine the location of the data they wish to analyze and log in to the Tier-A to submit the appropriate jobs. During this time BaBar will work with the Condor and Globus teams to determine the work needed to move towards automation, such as an interface between Globus and the BQS batch system used in Lyon. Starting in 2002, distributed job management will be progressively deployed and enhanced to meet the evolving needs of BaBar analysis.

As CS-developed tools become mature, normally about two years after the first attempt to use them in production, BaBar itself will take over the support responsibility for the BaBar collaboration, sharing this responsibility with other major user communities for the large fraction of the tools that have wide applicability.

Grid Services in BaBar

Goals and Tasks Year 1

- Goal: Successfully replicate 1/3 of the total BaBar dataset at IN2P3 Lyon.
- Deploy Globus Replica Catalog services in production. (CS-5)
- Start tests in a production environment of Globus and BaBar middleware enhancing transfer reliability and integrity. (CS-5, CS-6)
- Start pre-production work on distributed job management. (CS-1, CS-2)

Goals and Tasks Year 2

- Goal: Replicate a large fraction of the BaBar data at two Tier-A centers.
- Deploy reliability/integrity middleware in production. (CS-5, CS-6)
- Start production tests of storage and network resource discovery and scheduling. (CS-2, CS-4)
- Start production tests of distributed job management. (CS-1, CS-2)

Goals and Tasks Year 3

- Goal: Replicate large fractions of BaBar data at Tier-A centers.
- Goal: Seamless job submission and retrieval environment across Tier-A centers.
- Deploy storage and network resource discovery and scheduling in production. (CS-2, CS-4)
- Deploy distributed job management in production. (CS-1, CS-2)

Begin work on higher level optimization – automated decisions on bringing the job to the data or *vice versa*. (CS-1, CS-2, CS-4)

The CMS Experiment

Status and Milestones:

CMS has completed the second major object-oriented software development cycle, the “functional prototype” phase, in the development of its software framework (CARF), reconstruction code (ORCA, now in its fourth major release) and its interactive graphics analysis environment (IGUANA). Persistent objects are handled using Objectivity as the baseline ODBMS, and transparent access by users to a distributed federation of objects, where each user is able to seamlessly use a private schema. The third major software development cycle, leading to “fully functional software” is now underway.

CMS has been active and has led the development of the “Tiered” computing model adopted by all four LHC experiments. Members of CMS in the U.S. and in Europe, working with PPDG, GriPhyN and the EU DataGrid projects have produced the Grid Data Management Pilot system (GDMP). GDMP is now used to

support large scale distributed production of simulated and reconstructed events among an increasing number of sites (currently 8) in the U.S., Europe and Asia. Production cycles of one to two months are scheduled two to three times per year, in support of the development of the experiment's high-level trigger and physics reconstruction and selection (PRS) activities, and for studying in depth the detector's capabilities for discovering new physics. The Spring 2001 production cycle will use approximately 1000 CPUs and will result in an estimated 20 Terabytes of data stored.

A brief list of the major CMS data production and analysis milestones is given below.

- Dec 2001 Trigger and Data Acquisition System Technical Design Report (TDR)
- Dec 2002 5% Complexity⁴ Data Challenge
- Dec 2002 Software and Computing TDR
- Dec 2003 Physics TDR: Support interactive analysis
- Dec 2004 20% CPU⁵; ~Full Complexity

Grid Services in CMS

CMS's deliverables from PPDG involve the commissioning and extension of the general tools currently under development, particularly GDMP and the Globus infrastructures (information, security and metadata catalog), into production systems for distributed file generation, distribution and access in the first year. Support for access to extracted object sub-collections, as well as simultaneous access by multiple workgroups and individuals doing reconstruction and analysis should start in the first year, synchronous with initial developments underway in CMS, and should become an increasing focus in the second and third years.

1. Distributed Production File Service – CS-5, CS-6, CS-2

Distributed data generation, distribution, and archiving between FNAL, Caltech, UCSD, Florida, Wisconsin, INFN and possibly other U.S., European and Asian sites; based on a GDMP production version and Globus tools. CMS will build on its existing Computer Science collaborations with the Globus, Condor and LBNL Storage management teams. This provides generalizable extensions to the existing prototypes and satisfies the needs for CMS simulation production for the Dec 2002 milestones. The CMS requirements include:

- (a) Global namespace definition for files
- (b) Specification language and user interface to specify: a set of jobs, job parameters, input datasets, job flow and dataset disposition specification language, online user interface and forms⁶
- (c) Pre-staging and caching files; resource reservation (storage, CPU; perhaps network)
- (d) HRM integration with HPSS, Enstore and Castor, using GDMP
- (e) Globus information infrastructure, metadata catalog, digital certificates; PKI adaptation to security at HENP labs
- (f) System language to describe distributed system assets and capabilities; match to requests; define priorities and policies.

⁴ The number of major processing and data handling components (boxes), relative to the Tier0 system to be fully commissioned at CERN by 2007.

⁵ CPU power relative to the initial production system at CERN in 2007.

⁶ User interfaces are assumed to be browser-based, including a GUI, command-line and scripting capabilities.

- (g) Monitoring tools and displays to: locate datasets, track task progress, data flows, and estimated time to task completion; display site facilities' state (utilization, queues, processes) using SNMP; flag bottlenecks and redirect tasks using a request redirection protocol
- (h) Develop digital signatures characterizing each dataset, how it was processed, which parts of the signature (if any) are now invalid (calibration, software version etc.) mandating preprocessing.
- (i) Object-collection extraction, transport and delivery
- (j) Synchronization between DB metadata catalog (for files) and the Globus replica catalog

2. Distributed Interactive Analysis Service – CS-1 and CS-2

CMS physicists will need to have good access to the generated simulation data for analysis for algorithm and trigger development. In parallel with production Grid data distribution, distributed interactive analysis services must be developed and deployed. The following work extends the deliverables of the first year to add request cost estimation, enhanced user interfaces for data definition and system monitoring, as well as extending the integration of the developed services into the CMS analysis applications. Components of the Distributed Interactive Analysis services that we will concentrate on include:

- (a) User interface for locating, accessing, processing and/or delivering (APD) files for analysis
- (b) Tools to display systems' availability, quotas, priorities to the user
- (c) Monitoring tools for cost estimation, tracking for APD of files; request redirection
- (d) Display file replicas, properties, estimated time for access or delivery
- (e) Integration into the CMS the distributed interactive user analysis environment

3. Object-Collection Access – Extensions to CS-1, CS- 2

As the simulation and test beam data sizes grow the need for providing object level data definition and delivery services will increase. Our later focus will be to develop this as part of our PPDG deliverables and architecture, hopefully taking advantage of the algorithms and modules developed for file level query, delivery and access. As for all CMS deliverables the goal is to provide robust, fault tolerant services to a world wide community of physicists integrated with the CMS data processing and analysis applications. Components of the object collection extensions that we plan to work on include:

- (a) Object collection extraction, transport and delivery
- (b) Integration with ODBMS
- (c) Metadata catalog concurrent support

D0 Experiment

Status and Milestones:

The D0 experiment has developed a fully distributed data access system, SAM⁷, which has been extensively exercised on Monte Carlo and Cosmic test data during the construction of the detector upgrade. The SAM system has been deployed at NIKHEF in the Netherlands and IN2P3 in France to catalog, store and access files of simulated data. Data files are stored at these sites and transparently shipped over the network to Fermilab and placed in mass storage. As part of PPDG we are also collaborating on specific activities with D0 institutions in the Netherlands and England to extend the performance and functionality of the system for our European colleagues. The system will be maintained and extended to meet the evolving needs of data processing and global physics analysis. The SAM system includes features that are common to the HEP experiments including: 1. data replication, 2. disk cache management, 3. resource management, 4. metadata cataloging and querying, 5. dataset definition and processing history.

⁷ <http://d0db.fnal.gov/sam/>

At present, the SAM system performs job control (allocation and scheduling) for the data delivery jobs but not yet completely for the “real” data processing jobs of scientific applications. For the latter, the SAM system uses interfaces to the abstract batch system. The batch system uses its specific, opaque logic to schedule and run data processing jobs using SAM-supplied additional constraints and requirements. In its initial resource management, the SAM system strives to allocate processing (computing) resources together with the data delivery resources.

D0 has met its major data access milestones prior to the start of data taking. The milestones below enable us to plan and prioritize our work over the next few years and are associated with the extensions to provide more intelligent and robust system, as well as provide services for the very large data sets that will be available after several years of data taking.

- Mar 2001 Initial data taking, detector commissioning, data processing and analysis.
- Dec 2001 Definition of and initial implementation for formal job description language.
- Dec 2001 Initial support for globally coordinated physics analysis and transparent access to the data. Production support for regional analysis at Michigan State University, the University of Texas at Arlington (UTA), University of Lancaster (UK) and the University of Maryland, as well as IN2P3 and NIKHEF
- Mar 2002 Enhanced global performance monitoring and resource utilization integrated into SAM. Integration of Grid authentication and transparent mapping into Fermilab Kerberos authentication domain.
- Mar 2003 Enhanced resource and job dispatch management incorporating feedback from performance and resource utilization monitoring.
- Jun 2004 Enhanced services - including Grid/PPDG services as available - Dataset reclustering, restreaming, event and sub-event selection.

Grid Services in D0

We will concentrate our efforts in PPDG on extending the facilities for intelligent global data, job and resource management, and integrating these into the production SAM services for the benefit of the D0 physics community, including collaborating on specific activities with D0 institutions in the Netherlands and England to extend the performance and functionality of the system. We will build on the architecture and design of the existing and already planned versions of SAM, and work with our Computer Science and other Experiment PPDG collaborators to: define interfaces between the Grid fabric and the D0-SAM specific data system; extend the SAM system to use and help bring to production Grid middleware and PPDG deliverables as they become available; and extract SAM services to become components potentially reusable by other experiments.

Formal Job Description and Resource Management Language - Year 1 (CS-1)

For efficient data access, D0 requires a unified approach to the allocation and scheduling of all resources pertaining both to the data delivery and to the processing. Depending on the priorities set by the experiment, such scheduling must aim to achieve primary goals: maximizing the number of processing jobs; and implementing the experiment policies for sharing resources and establishing priorities among competing access modes and research groups within the experiment. Distributed analysis is a parallel distributed activity of recurring processing of certain data. To illustrate, consider dispatch of such a parallel job on a farm (cluster) where the set of nodes whose disks hold data does not, at any given time, correlate with the set of the nodes whose processors are idle.

Since the application is data-intensive, the comprehensive job control system must balance (a) the “expensive” data delivery onto the nodes that don’t have the job data with (b) forcing the job to wait for the processors where the data is already present. The decisions must be made depending on relative “costs” of data delivery and of increasing the job latency. These costs incorporate experiment policies, local

prioritization of system resources, and the conditions at external, “global” mass storage systems, all of which being dynamic.

The main CS project being proposed lies in co-scheduling and co-management data processing and delivery activities, in other words, comprehensive resource management subject to the above D0 requirements. D0 SAM will present its (dynamic) requirements in a formal job description and resource management language, to be understood by the Grid. In order to allow for inclusion of both the optimality and policy considerations, D0 envisions that the language will use economics concepts similar to “benefit”, “cost”, “value”, “fair share”, etc..

The deliverables will include:

- a) Formal Job Description language defining the metric(s) to be optimized in the course of the comprehensive resource management
- b) Software Components from Computer Science partners with solutions to the optimization problem.

Global Monitoring of Resource and System Performance and Utilization - Year 1 and 2 (CS-3)

SAM already includes system performance, monitoring, and resource utilization statistics collection and display. These are currently implemented in a pragmatic and sometimes ad-hoc fashion. We plan to take advantage of the opportunity to work closely with a Computer Science group to enhance the design and implementation to be scalable, flexible and reusable. We will reintegrate the final product back into the SAM system to increase our ability to support the system and reduce the human resource overhead needed to support 24x7 operations. System performance and resource utilization statistics will be fed into the global resource and job management framework and used to improve the optimization and aggregate work done by the system. We will investigate existing Grid performance measurement tools - such as Netlogger. We will investigate data definition languages and message protocols and hope to take advantage of other PPDG developed display and web presentation tools. The deliverables will include

- a) Acquisition – appropriation, extension and/or development - of Grid performance measurement and resource utilization tools and integration into SAM
- b) Acquisition of monitoring and statistics display components and their integration into SAM.
- c) Access to and interpretation of the statistics by SAM resource management

Enhanced Integrated Production Experiment wide analysis job and data placement - Year 1, 2 and 3 (CS-1, CS-2)

Throughout the project we will work to consolidate and deploy the global distributed data access and analysis system as a production service to the D0 Collaboration. We will concentrate on robustness in job dispatch, data placement, and distributed cache and resource management, and the support of fully transparent user control and response. This is clearly a challenging project in its own right - to provide isolation of the applications from faults, restarts, and bottlenecks in the global system as well as provide the user with accurate, complete, timely and simple to diagnose information about the state and errors encountered. The experiment analyses will be able to make most advantage of our distributed processing environment if the system operates at a high degree of availability - as a production, fault tolerant, dynamically reconfigurable and extensible global system. Anticipated deliverables include.

- a) Production global data delivery for D0 collaborators.
- b) Grid based fault tolerant, restart and error response services integrated into SAM
- c) Integration of Globus authentication services and automatic translation of Grid authentication to the Fermilab authentication realm.

Enhanced Data Reclustering and Restreaming Services - Year 3 (CS-2)

As the stored data size grows we will need to enhance our data delivery and job dispatch optimization techniques. The D0 physics community will be loath to tolerate any deterioration in the service provided - in terms of speed and ease of access. We will develop extensions for the placement and selection of the data. Placement of the D0 data is controlled by instructions to the SAM system. Events are streamed - with data in the same stream being co-located. Once analysis of the data is advanced, we can profit from analysis of the data access and job execution profiles. We may well gain benefit from implementing different streaming criteria from those initially chosen, to more optimally reflect the experiments data access patterns. Associated with this, we will explore the tradeoffs between the latencies introduced and resources used by transparent reclustering of the data, and the performance enhancements obtained by providing the user analyses more efficient run-time access to the data. Deliverables are

- a) Algorithms to define the best placement and delivery of the data given complex data selection criteria and complete knowledge of the state of the SAM system.
- b) PPDG components to automatically recluster and restream D0 event data.
- c) Enhanced facilities for the selection of data by Event, Sub-Event and more complex query definitions.

The STAR Experiment

Status

The STAR experiment at RHIC, developed over the past decade, began taking data in the summer of 2000. In the first run it acquired about 5 million events that occupy about 4 TB of storage. These data have been reconstructed several times and each version of the summary data (DST) occupies 0.5 TB of storage. Beginning in June of 2001 STAR will begin its second data taking run which is expected to last until March 2002. This second run will generate about 50 times more data than the first run, or around 200 TB of raw data. There will be an equivalent run each year and plans are being made that will result in even greater data volumes.

Grid Services in STAR

There are three types of computing activities in STAR that are targeted to benefit from Data Grid services. These are: A) bulk file replication between BNL and LBNL, B) job control and management of various production computing activities (simulations, DST production, mini-DST production), and C) coordinated storage and computation suitable for data analysis. For each of these activities STAR will participate in the collection and documentation of current practices (CS-7). While all of these activities will ultimately include each of the CS work areas, we envisage a phased approach that optimizes the cost/benefit and also acknowledges the schedule of middleware development

Year 1 – Site-to-site bulk file replication – CS-5, CS-6

The primary activity in the first year is the integration and deployment of Grid services for a site-to-site file replication service between Brookhaven and Berkeley labs. This will begin with the file transfer service (CS-6) followed by integration of replica services (CS-5) and storage resource management (CS-4) as this additional middleware functionality becomes available. In the following years (2 & 3) we will integrate higher level services (CS-3, CS-1, CS-2) after they have been developed in the context of other experiments involved with PPDG.

Year 2 – Job control and management of production computing activities – CS-1, CS-2

The main activity in the second year is the integration and deployment of job control (CS-1) and scheduling (CS-2) for production computing activities in STAR. This will first be applied to production simulations at Berkeley lab and then to data processing at Brookhaven lab. We expect that the CS-1 and CS-2 services

are developed in the context of other experiments in PPDG and STAR will participate in extending these services as a second round activity. Following the initial deployment in STAR we will include the storage resource management (CS-4) and monitoring (CS-3) services.

Year 3 – Coordinated storage and computation – CS-1, CS-2, CS-4, CS-5

The main activity in the third year is to integrate and deploy Grid services for coordinated storage and computation activities in STAR. This will be used primarily for data I/O intensive physics analysis activities. This will be deployed first utilizing the job control (CS-1), scheduling (CS-2) and storage resource management (CS-4) services. After initial deployment the replica services (CS-5) will be included as part of automating the management of secondary storage.

Generalization of services – CS-7

Following deployment and utilization of all of these Grid services, STAR will participate in the generalization of these services for the benefit of other experiments.

Thomas Jefferson National Accelerator Facility Experiments

Data Grid Status and Milestones

Jefferson Lab experiments have thus far acquired over 250 terabytes of data, held in a StorageTek silo (300 terabytes) with a large disk cache (25 terabytes). The first stage reconstruction of the data is currently carried out on an array of 150 nodes in the lab's batch analysis farm, with access to data managed by custom Java-based silo and disk pool management software (JASMine). Physics analysis is carried out both at the laboratory and at collaborating universities around the world, with datasets still being moved in some cases via physical tapes. Detector simulation data is generated off-site and moved to the silo either over the network or by tape.

The laboratory has embarked on an upgrade of its analysis infrastructure, and is moving toward a multi-tier model, similar to that proposed for LHC. This upgrade will be necessary to support future upgrades at the laboratory, including an energy upgrade and a new experimental facility that will increase data production by an order of magnitude. One component of this infrastructure upgrade will be a Jefferson Lab Data Grid that combines in-house silo and disk cache management software with components from PPDG. The enhanced capabilities will be integrated into a next-generation analysis framework for the CLAS collaboration (CEBAF Large Acceptance Spectrometer, Hall B) and used for the future Hall D program.

In collaboration with the Lattice Hadron Physics Collaboration, including MIT, Jefferson Lab is prototyping the use of web technologies to build a simulation and data analysis meta-facility that will give access to distributed batch systems and data management resources. Already the Lattice Portal⁸ provides the ability to submit and control batch jobs and retrieve data files from the Jefferson Lab silo. This software will be extended to multiple sites, with file transfers between the sites handled by components from PPDG.

Grid Services at Jefferson Lab

Near-term major milestones in this project include:

- Sept 2001 Replicated data services (raw and reconstructed data) between Jefferson Lab, MIT, and ODU (CS-4, CS-5, CS-6)
- Feb 2002 Automated policy based replication (push) of raw data (a subset) and reconstructed data to several universities involved in running experiments (CS-2, CS-3, CS-4, CS-5)

To achieve these milestones, Jefferson Lab will in the first year of this proposal:

⁸ <http://lqcd.jlab.org/>

- work with developers within PPDG involved in standardizing interactions between client applications and the disk resource manager (protocols, application programming interfaces, etc.), as a first step towards integrating this capability into the CLAS framework (CS-4, CS-5, CS-6)
- deploy and integrate the Globus developed GridFTP component, integrating the server piece with the existing disk management software to support both file retrieval and authenticated uploading of files into the disk cache (CS-6)
- begin a study of selecting datasets for analysis based upon data characteristics rather than filenames

Additional tasks to begin as prototyping work in the first year and move into full development in the second and third years include:

- managing the flow of datasets to and from off-site batch jobs (CS-1, CS-2)
- migrating jobs and/or data between sites (load balancing), taking into account load, network bandwidth, etc. (CS-1, CS-2)
- monitoring the state / health / load of the integrated system (silos, disk, compute, network) with interactive web interfaces (CS-3)
- generating trend presentations (for capacity planning) on the web (CS-3)
- supporting easy integration of additional university sites (a deployable package, including documentation, etc.) (CS-4, CS-5, CS-6)

Work Plan & Schedule

The Experiments collaborating on PPDG are at different phases in their life-cycle. BaBar, D0 and Star have well developed data handling and processing systems that are already in production use. Their PPDG deliverables address specific needs to extend the already existing services, often to accommodate the requirements of increasingly active European-American analysis efforts. The Thomas Jefferson Laboratory program is designed to extend the facilities offered by the laboratory to its general community. The list of PPDG deliverables given by Atlas and CMS reflect the fact that the experiments are in the early stages of development of their global data handling and processing systems. Clearly the deliverables and programs of work that describe how Grid services will be applied to their computing activities exceed the resources available to the PPDG project per se. We are including the list here to show the commonality of requirements between the experiments and the overlap of these requirements with the Computer Science work areas listed above. The decision of the U.S. leaders of the experiment specific data processing projects to include these lists in the PPDG proposal reflects their enthusiasm and commitment to work together on common developments on PPDG and other the Grid development projects.

Ongoing Activities

All Computer Science and Experiment groups will participate in the following activities

- *Dissemination of Information.* We will setup communications forums including mail groups, document libraries (requirements, specifications, project plans, APIs, etc) – all contained at www.ppdg.net.
- *Document current practices* – create a joint document describing individual experiments current practices.
- *Project Activity definition* – all CS teams and experiments will meet to refine and define a vision of the types of common solutions we want to wind up with in production by the culmination of the project. *Activity refinement meetings* will be held again in January of 2002 and 2003 to review status and plan the next year's Activities. This will include knowledge gained and accounting for the expected rapid advances in hardware and software that will be taking place as we proceed.

- *Design of mechanisms for technology delivery* that achieves a prudent level of commonality between experiments and between CS technologies in the area of how the deployed CS technologies are packaged, installed, and maintained. This task will try to reduce costs and leverage economies of scale, but will also allow for differences between different technologies or experiments (typically based on history) that merit departures from a common approach. This activity should encompass the issues of configuration management, patching, install scripts and conventions, etc. Ideally, all delivered CS technologies should look like members of a single technology product suite.
- *Middleware testbed design and deployment* – for each experiment we will define one testbed system to be used by the experiment for the deployment and testing of the PPDG deliverables.. In addition we will create a CS test to serve as a reference platform for the project to view and experiment with a working example of the base CS technologies – Condor, Resource Management, and Globus. We will maintain the concept of testbeds throughout the project in order to give application teams quick but accurate access to emerging technologies and feature sets in a controlled environment.
- *Design of initial deployment ordering* – for each app we will design the initial integration of technologies that can actually be brought into production use. For the initial deployment, we will try to keep the number of new untried technologies to a minimum, building instead on components that already exist. This step will take the form of: a functional specification of user-visible features; an internal design specification of components to be deployed, interfaces to be used between the components, and details of any new integration or adaptation code or new features that need to be developed. For this (and all subsequent) deployment designs, we will devote affixed amount of time to the identification of common solutions across multiple application teams.
- *Project Deployment Cycles* Each Computer Science working area will consist of approximately 3-4 major cycles of development and deployment. We will write overall plans for these, and the anticipated deliverables for each. Reviews will be organized by the executive team during each project deployment cycle to ensure project wide coordination and provide opportunity for input and feedback.
- *Investigate Applicable Commercial Technologies:* We will continue to understand and investigate relevant commercial developments in areas of Grid technology . We will work with such commercial interests on technologies potentially mutually beneficial for PPDG.
- *Participate in Grid Standards Activities.* We will contribute to and conform with the standards efforts in the Global Grid Forum, IETF, and W3C.

Project Activities

Each Project Activity will follow a traditional project plan, both for the initial and all subsequent deployments. For each Activity the two project leaders (experiment and CS) will develop and publish a detailed plan to complete all necessary development and integration and to phase them into production. Regular status of each activity will be presented to the collaboration. Each Project Activity will consist of

- Deliverables assessment analysis and deployment plan
- Specification and design (including commonality assessment and planning)
- Execution of deployment plan – including documentation and testing.
- Operation of service and performance analysis
- Analysis of future needs and potential for adapting the deliverable to other experiments.

As stated previously, not all the experiment Grid services listed above will be addressed by PPDG Activities. The development methodology will be applied to those activities that are well defined

collaborations between one or more experiments and one or more CS groups towards a well defined deliverable and goal. The Project Activities are grouped into the seven defined CS areas of work.

The work plan attempts to accommodate the individual experiment needs as well as allow some serialization of the CS work to allow sufficient resources for a full deployment cycle, including analysis, design and development, to take place. The work plan is designed to show a process for transferring and extending a deliverable from one experiment to the next. Following the first round of deliverables we will evaluate their applicability for adaptation and reuse for other experiments. After each activity some support and maintenance will be required for existing deployments by the continuing activities in the area.

The Project Activities for the three years show a progression. The first year concentrates on the extension and integration of existing software into the experiment systems and deployment of reliable, robust existing services. In the second year the focus is on extending the functionality of services and the transition of first year deliverables to other experiments. By the third year it is hoped that the experiments will be moving towards the integration of a common infrastructure.

Project Activity	Experiments	Yr1	Yr2	Yr3
CS-1 Job Description Language – definition of job processing requirements and policies, file placement & replication in distributed system.				
P1-1 Job Description Formal Language	D0, CMS	X		
P1-2 Deployment of Job and Production Computing Control	CMS	X		
P1-3 Deployment of Job and Production Computing Control	ATLAS, BaBar, STAR		X	
P1-4 Extensions to support object collections, event level access etc.	All			X
CS-2 Job Scheduling and Management - job processing, data placement, resources discover and optimization over the Grid				
P2-1 Pre-production work on distributed job management and job placement optimization techniques	BaBar, CMS, D0	X		
P2-2 Remote job submission and management of production computing activities	ATLAS, CMS, STAR, JLab		X	
P2-3 Production tests of network resource discovery and scheduling	BaBar		X	
P2-4 Distributed data management and enhanced resource discovery and optimization	ATLAS, BaBar			X
P2-5 Support for object collections and event level data access. Enhanced data re-clustering and re-streaming services	CMS, D0			X
CS-3 Monitoring and Status Reporting				
P3-1 Monitoring and status reporting for initial production deployment	ATLAS	X		
P3-2 Monitoring and status reporting – including resource availability, quotas, priorities, cost estimation etc	CMS, D0, JLab	X	X	
P3-3 Fully integrated monitoring and availability of information to job control and management.	All		X	X
CS-4 Storage resource management				

P4-1 HRM extensions and integration for local storage system.	ATLAS, JLab, STAR	X		
P4-2 HRM integration with HPSS, Enstore, Castor using GDMP	CMS	X		
P4-2 Storage resource discovery and scheduling	BaBar, CMS		X	
P4-3 Enhanced resource discovery and scheduling	All			X
CS-5 Reliable replica management services				
P5-1 Deploy Globus Replica Catalog services in production	BaBar, JLab	X		
P5-2 Distributed file and replica catalogs between a few sites	ATLAS, CMS, STAR, JLab	X		
P5-3 Enhanced replication services including cache management	ATLAS, CMS		X	
CS-6 File transfer services				
P6-1 Reliable file transfer	ATLAS, BaBar, CMS, STAR, JLab	X		
P6-2 Enhanced data transfer and replication services	ATLAS, BaBar, CMS, STAR, JLab		X	
CS-7 Collect and document current experiment practices and potential generalizations	All	X	X	X

PPDG Management Plan

PPDG plans a two component management structure with:

1. An **executive team** composed of Ruth Pordes (PPDG Steering Committee Chair), Doug Olson (Steering Committee Physics Deputy Chair) and Miron Livny (Steering Committee Computer Science Deputy Chair). All three members of this team have extensive experience in managing software development and deployment projects and will each make PPDG a principal activity. They will be tasked to track the goals and work of the physics-CS activities teams and provide guidance to ensure overall coherence of the PPDG Collaboratory Pilot. Miron Livny will steer the project deliverables towards maximal commonality (and wide usability) in the components of each experiment's vertically integrated Grid software. The team will work together to advise the Steering Committee to best meet the short and long term goals of the Collaboration.
2. A **Steering Committee (SC)** comprising one representative from each physics experiment and one representative of each Computer Science group. The project PIs and the executive team will be ex officio members of the Steering Committee. In the interests of keeping the SC small and effective, the project PIs and members of executive team may also act as experiment or CS group representatives if they wish. The preliminary membership of the Steering Committee is:

Ruth Pordes, Chair/DO Rep.

Miron Livny, Project PI/Computer Science Deputy Chair/U.Wisc CS Team Rep.

Doug Olson, Physics Deputy Chair

Richard Mount, Project PI/BaBar Rep.

Harvey Newman, Project PI
Lothar Bauerdick, CMS Rep.
Torre Wenaus, ATLAS Rep.
Chip Watson, JLab Rep.
Matthias Messer, STAR Rep.
Ian Foster, ANL CS Team Rep.
Arie Shoshani, LBNL CS Team Rep.
Reagan Moore, SDSC CS Team Rep.

The steering committee is structured to provide decisive management and to balance the Computer Science, software technology, physics-experiment needs and budgetary constraints.

Clearly, the success of such coordination and additionally the success of coordination with other HENP Grid projects will be based on an open discussion and decision making process. Both the Steering and Executive committees will publish agendas and minutes of their meetings and are charged with ensuring that successful communication and coordination exists throughout the Collaboratory Pilot through the three years of the project.

List of Participants

Computer Science Teams

Computer Science Department, University of Wisconsin
Miron Livny¹¹ (PI), Paul Barford¹¹

Mathematics and Computer Science Division, Argonne National Laboratory
Ian Foster¹, William Allcock¹, Mike Wilde¹

Scientific Data Management Group, NERSC, Lawrence Berkeley National Laboratory
Arie Shoshani⁵, Andreas Mueller⁵, Alex Sim⁵

San Diego Supercomputer Center
Reagan Moore⁶

Physics Experiment Teams

ATLAS

Torre Wenaus², Rich Baker², Stewart Loken⁵, David Malon¹, Ed May¹, Razvan Popescu²,
Larry Price¹, Alex Undrus², Alexandre Vaniachine¹

BaBar

Richard Mount⁷ (PI), Robert Cowles⁷, Andy Hanushevsky⁷, Adil Hasan⁷

CMS

Harvey Newman³ (PI), James Amundson⁴, Paul Avery¹¹, Lothar Bauerdick⁴, James
Branson⁹, Julian Bunn³, Ian Fisk⁹, Gregory Graham⁴, Takako Hickey³, Koen Holtman³, Iosif
Legrand³, Vladimir Litvin³, Vivian O'Dell⁴, James Patton³, Asad Samar³, Conrad Steenberg³

D0

Ruth Pordes⁴, Lee Lueking⁴, Wyatt Merritt⁴, Igor Terekov⁴, Sinisa Veseli⁴, Rich Wellner⁴

STAR

Matthias Messer², Bruce Gibbard², Eric Hjort⁵, Doug Olson⁵

Thomas Jefferson National Accelerator Facility (JLAB)

Chip Watson⁸, Ian Bird⁸, Ying Chen⁸

Liaisons

GriPhyN Project – Paul Avery, University of Florida

Institutions

¹Argonne National Laboratory, ²Brookhaven National Laboratory, ³California Institute of
Technology, ⁴Fermi National Laboratory, ⁵Lawrence Berkeley National Laboratory, ⁶San Diego
Supercomputer Center, ⁷Stanford Linear Accelerator Center, ⁸Thomas Jefferson National
Accelerator Facility, ⁹University of California at San Diego, ¹⁰University of Florida, ¹¹University
of Wisconsin

Reports from PPDG Teams

ATLAS

US ATLAS Grid Testbed

In ATLAS, a primary focus of PPDG efforts in this quarter has been establishing a US ATLAS grid testbed involving our PPDG sites (ANL, BNL, LBNL) and several university sites (Boston U, LBNL, Indiana U, U of Michigan, U of Oklahoma, and U of Texas-Arlington). The effort has been organized by PPDG collaborator Ed May at ANL. Each site is running a Globus 1.1.3 gateway and hosting a standard set of grid tools and ATLAS applications. This testbed will be used for PPDG tool development and testing. This is intended to be a continuously running test and development vehicle for PPDG, GriPhyN and ATLAS. A status display of the test grid can be found at <http://heppc1.uta.edu/kaushik/computing/grid-status>. We run a weekly technical coordination by phone/VRVS conference.

The implementation of the ATLAS US grid testbed at ANL-HEP is described at <http://www.hep.anl.gov/globus>.

The Berkeley effort has focused on bring up a testbed node at the NERSC-PDSF. We have installed Globus on a gateway machine so that other ATLAS can submit jobs to the PDSF LSF. ATLAS members can now work on PDSF using certificates issued by Argonne and Berkeley as well as from a number of other top-level domains (NPACI, NCSA, the DOE Science Grid and NASA). Working with Ed May at Argonne, we have resolved the problems associated with installing certificate-signing policies on ATLAS machines.

The Tier 1 site at BNL deployed a dedicated Linux farm (currently two nodes, but expandable to meet demand) to serve requests from other US ATLAS Grid sites via LSF queue. The Tier 1 disk storage resources are available for Grid data transfers. We have been actively working on enabling Grid access to the Tier 1 HPSS storage and we expect significant progress during the next quarter.

Early grid connectivity to Europe is vital for the Tier 1 center. During the quarter, BNL demonstrated the ability to exchange authentication credentials with our European collaborators. Globus requests from Italy were successfully executed at BNL.

Further information on the US ATLAS grid testbed can be found at <http://www.usatlas.bnl.gov/computing/grid/>.

Distributed data services testbed

At ANL, testbeam data is in use to provide a development and test environment for PPDG distributed data services. ATLAS Tilecal testbeam data for 1998 and 1999 stored at NERSC's HPSS service hpss.nerisc.gov was copied (replicated) at the HPSS service at BNL, hpss.rcf.bnl.gov. Perl scripts were written to gather replication information data. This information was used to populate a MySQL database at BNL to provide access to the replication data and meta-data describing the ATLAS Tilecal testbeam raw data and Objectivity database files. The schema is shown at <http://gate.hep.anl.gov/globus/replication.html> Each raw data replication is about 1400 files or 150GB in total. We are currently building a complete Objectivity database at BNL, again about 1400 files or 200GB in total. We will use these data to test the PPDG replication management and transport services.

Rapid prototyping tool for distributed data services

At BNL, a rapid prototyping tool 'DBYA' was developed for design studies and component prototyping for the distributed data service that is the principal ATLAS year 1 PPDG milestone. A MySQL database is used for cataloguing, metadata and the management of distributed data stores. Perl scripts populate the databases and drive the autogeneration of C++ and Java interface code based on database schema. Browsing and querying is available via web interface and command line. The system presently catalogs about 30k files on disk and in mass store (BNL HPSS and CERN staging system), including the main simulation data repository for ATLAS, the tile calorimeter test beam data replicated at BNL, and the ATLAS software repository. It is presently in use for developing experiment-specific catalog loading infrastructure, and exploring the organization and schema for metadata describing both the data and the

infrastructure of the distributed data service. The web interface for the system is at <http://atlassw1.phy.bnl.gov/dbya/dyShowMain.pl> and further information is available at <http://atlassw1.phy.bnl.gov/dbya/info>

CMS

We have made progress in several PPDG-related areas. These include: building a distributed file service based on GDMP, commissioning a prototype Tier2 center, beginning development of a remote data analysis service and optimized access to event TAGs, performing large scale distributed simulations, studying load-balancing strategies among Grid sites by extending the MONARC simulation system, developing an execution service for a large number of processors, specifying CMS' Virtual Data requirements, modeling the CMS Grid workloads, developing a framework for coordination among Grid projects. Parts of the work were done in coordination and collaboration with the GriPhyN and European projects. A detailed review of the recently released Grid Architecture documents of Foster, Kesselman et al. also was conducted.

1) *Installation, Configuration and Deployment of Prototype Tier2 Center*

A prototype Tier2 center has been set up and commissioned. It is distributed in two halves each with 20 dual CPU nodes at Caltech's CACR and SDSC, interconnected by CALREN2. Later this year we intend to double the number of nodes, and we hope to upgrade the Caltech-San Diego network connection using NTON.

For the Caltech half (operations at San Diego were similar):

- a) The Nodes were purchased from Datel systems in San Diego, built around Motherboards by SuperMicro (www.supermicro.com). RAID arrays by Winchester Systems (www.winsys.com) were selected on the basis of performance and price. The Ethernet Gigabit/Fast switches are HP2524s. A Dell 4400 server provides the data service and network interface to the system, and is equipped with Gigabit Ethernet cards by SysKconnect.
- b) Benchmarking of RAID arrays: Extensive testing has been performed on the Winchester arrays. These currently include the latest Ultra160 RAID controllers. Current rates are 70 MB/sec read and write to each of the 0.5 TByte arrays. The filesystem being used is reiserfs journaling.
- c) PBS batch system installed. Will install Condor-G by the end of March.
- d) Installation of Globus has been completed
- e) Gbit connections to CACR HPSS system were completed
- f) Node cloning was completed after debugging of the procedure.
- g) The full CMS software installation has been completed on the Tier2 prototype and made ready for production (CMSIM simulation and ORCA reconstruction). Objectivity/DB 6.0 and AMS servers have been fully set up and tuned. 500k CMSIM events and 150k fully reconstructed events were produced during the first half of March.
- h) The latest version of the Globus software was installed.
- i) The latest version of GDMP was installed, tested, and deployed in the transfer of minimum bias events both from CERN and from Jasper, to the Tier2.
- j) In mid-March a 1 TByte disk server was purchased from ASA Computers, and installed in the Tier2. This device is built around 3ware ATA RAID controllers, which are hosted in a motherboard that includes dual 1GHz Pentium CPUs, 1 GByte RAM and dual SysKconnect Gbit Ethernet cards. The RAID controllers are capable of RAID 0,1,10 and 5. Performance tests are planned to compare this array with those from Winchester (the price per TByte being around four times cheaper for the ATA RAID).
- k) System management continues to be a significant aspect of work on the Tier2. Tasks include debugging of systems and network problems, ordering hardware (e.g. heavy duty rack for the ASA Server, console switches), installing new hardware (e.g. new scratch disks), monitoring hardware

(e.g. controlling rack temperatures, placement of components, air flow etc.), liaising with the vendors for equipment returns and replacements, organizing engineer visits, registering new users, answering user's questions, and so on.

- l) The TQS system (see below) is being developed on the Tier2.
- m) Locations of the HEP equipment in the CACR machine room were rationalized by moving items into spare space in the Tier2 racks.
- n) Use of the Tier2 by "remote" collaborators, e.g. at UC Davis and UCLA, is increasing.

2) *Grid Data Management Pilot (GDMP)*

Continued progress was made with GDMP in terms of stability and bringing it closer to production quality software. Large-scale stability testing of GDMP has been performed between CERN and UC San Diego exercising the entire system. Objectivity database files were staged from the CERN mass storage system to the CERN disk, transferred to San Diego, and attached to the local federation. 150GB was transferred during the evaluation and the results were fed back to the development team. Many other sites also took part in GDMP testing. Around 250GB of data was transferred to Caltech and almost 150GB to FermiLab from CERN. Some 100GB have been transferred from INFN to CERN and around 20GB was transferred from Moscow to CERN showing how GDMP behaves on a slow network link. We gained very important experiences from these tests and the result is the new GDMP release (version 1.2.2). This latest version keeps track of the state of the system at a particular time to initiate fault recovery mechanisms. It also has quite a few bug fixes which became visible only after the large scale tests described above.

Interfacing the Globus replica catalog with GDMP is underway. The part related to writing into the Catalog is complete and the current work is on enabling searches for files in the catalog and allowing their deletion. This will in the end provide a high level abstraction to hide the details of the Globus replica catalog for ease-of-use as well as for providing the option of choosing some other mechanism than Globus, without having to change the application's interface to this.

Coordination of an effort to develop implementations of the HRM APIs defined by PPDG has begun. FermiLab is working on an implementation for Enstore and LBNL has just come up with another one for HPSS. We started discussions with people at CERN to convince them to do the same for CASTOR as well, so that integration of GDMP with the tape systems at FNAL, CERN, Caltech and elsewhere can be made location-independent. The plan is then to integrate this API within GDMP to provide a complete end-to-end solution, including staging files in from the tape to disk, transferring them over the WAN and seamlessly staging them out from disk to tape at the destination.

3) *Development of a Remote Analysis Service*

This will be based on existing technology: IGUANA/ORCA on the server and Java Analysis Studio (JAS, from SLAC) on the client. We are planning to use a thin client downloadable from a Web browser. IGUANA/ORCA will be encapsulated as a "black box" inside a server running on the proto-Tier2 system. ORCA databases (probably an existing large Muon sample) will be hosted on the master node and on the slave nodes (probably the Tag will be split 20 ways and we will put each piece on each of the 20 slaves and run an AMS there). Remote JAS clients will be able to request execution of arbitrarily complex queries across the Tag (and deeper), and get the histogram/plot results returned and rendered in the client.

The choice of JAS was made to eliminate the software distribution burden on clients without the expertise or the OS environment to install IGUANA. So JAS is a "lightweight" client, with IGUANA being the "heavyweight" client, that requires linking with LHCXX, ORCA, etc. on the local machine. JAS only requires JDK 1.2 or later, so it is suitable e.g. for someone to use on a laptop or other resource/bandwidth constrained client where running a remote IGUANA/X11 session is impractical.

JAS already implements remote analysis capabilities through RMI, and has a data interface module (DIM) on the server that knows about tags and HTL histograms. See <http://www-sldnt.slac.stanford.edu/jas/> for more.

The separation between Java and C++ is currently somewhere in the middle of the server side, which is implemented using Java and some JNI calls.

The goal is, however, to eventually implement the server side using pure C++, and use a language-neutral transport, such as CORBA, XML-RPC or SOAP, instead of RMI. This would have several benefits including much better scalability, better integration with ORCA/CARF, and importantly, make the work directly useful to other non-Java analysis systems such as IGUANA, Root or Lizard. Initial prototype already running on the Caltech Tier2 system.

4) *Large Scale Distributed Simulations*

These continue on the X-Class 256 CPU Exemplar, where we are generating one million full QCD background events for Higgs--> gamma gamma research.

Additional runs on the Wisconsin Condor have been proceeding. All previously identified CMSIM problems with these runs have been fixed. Combined usage of Linux and Solaris parts of Condor flock together will be tested in April.

Full CMS software has been installed and tested both on the RoadRunner and the Los Lobos Linux clusters in New Mexico. Some tests have been completed involving running processes on these clusters and writing data across the WAN to the Caltech tier2.

A script-based system for CMSIM production has been created for the Maui resource manager installed on the LosLobos cluster, and this is working fine. An ORCA script system is in preparation.

Takako Hickey's scheduler system (TQS; see below) has been installed on the LosLobos cluster. A script-based system for Takako's scheduler system has been installed and successfully tested.

The full CMS software suite has been installed and tested on Chiba City Linux cluster in Argonne (ANL). Some tests have been completed involving running processes on this cluster and writing data across the WAN to the Caltech tier2. After solving some technical problems on Chiba a script-based system for CMSIM and ORCA production will be tested in April.

The full CMS software suite has been installed and tested on new Platinum Linux cluster in NCSA. for using it in Data Terascale Facility prototype tests. Some tests have been completed involving running processes on this cluster and writing data across the WAN to the Caltech Proto-Tier2. The Linux clusters at Caltech (pTier2), NCSA(Platinum), Wisconsin(Condor Flock), and New Mexico (Los Lobos), along with the HPSS mass storage system at Caltech and the UniTree system at NCSA will be used to test a "DTF Prototype MicroGrid" during the next month. In the MicroGrid, a coordinated simulation and reconstruction run will be launched using Condor-G, scheduled, and data flow controlled, all from Caltech.

5) *Prototype Distributed Production System for Simulated Events*

The Fermilab CMS group is working with the University of Wisconsin CMS group and the Condor team on developing a prototype distributed Monte Carlo production system. The system is designed to act as a testbed for PPDG technologies while starting to meet real needs for CMS Monte Carlo production efforts. The goals of the system are automation of tasks, robustness and "monitorizability". The production problem is divided into two parts: job management and file management.

Condor G is being used as the primary Grid tool for the job management section. The most significant piece of software required for automatic job management is a system to translate physics requests into runnable jobs to be submitted to the Condor G manager. A working prototype of this software has been written at Fermilab. It continues to be developed. Fermilab has started work on the Condor G portion by setting up a test installation on local machines.

The file management portion of the system is still in the planning stage. The planning has consisted of identifying which tools come closest to meeting our needs and understanding which extensions to them will be necessary. We have chosen GDMP to manage file transfers and cataloging. We will require the ability to transfer plain files, i.e., file not part of an Objectivity database. This capability is part of the planned future of GDMP, but is not yet complete. We will also need to use GDMP to read and write files from the mass storage system at Fermilab. For this part of the project we are working on extending the HRM interface itself to include writes as well as reads. We will then add this interface to

the HRM-GDMP work we started during the previous quarter. Finally, for the monitoring portion of our system, we plan to test the Globus replica catalog as used by GDMP for plain file cataloging.

6) Distributed Systems Modelling

An evaluation of a job scheduling system based on a self-organizing neural network [1] (SONN) has been made using the MONARC Simulation system. The simulation toolset was extended and allows one to dynamically load any scheduling module for each Regional Center. This dynamic scheduling system should be seen as an adaptive middle layer software, aware of currently available resources and using the past experience to provide an effective use of resources and Optimize specific users' requirements. In the case studies used in these simulations [2], the SONN correctly identified the correlations and also performed a search in an unknown part of the parameter space, thereby achieving a significant improvement in the turn-around time for the jobs submitted in this distributed system. Currently work is proceeding to include such an algorithm into a distributed agent-based system using the JINI technology and evaluate it in the framework of a real prototype Tier2 center. Such an approach for a hierarchical scheduling system requires integration with local job scheduling and monitoring components. The aim is to build a prototype for a flexible federated management architecture.

In addition, two large-scale simulations of data replication between Regional Centers were performed. In the first case [3], full data reconstruction was modelled as occurring at the Tier0 center, and after each job was finished the reconstruction results were modelled as being replicated to several Tier1 Centers. The contention and the limit on the I/O in the data base servers could be observed as the number of replications occurring in parallel rose as the production load increased. In the second case [4] Event selection at one Tier1 center was modelled. In this simulation the resulting Analysis Object Data (AOD) files were modeled as being distributed to 25 Tier2 Regional Centers. Both of these studies were undertaken as an evaluation of possible Regional Center architectures.

7) Partitionable Execution Service for Distributed Processors (TQS)

Work is well advanced on a prototype partitionable execution system and a toolset that aids high energy physicists in effectively using computational resources distributed worldwide. The initial system design has focussed on the following features that are not addressed adequately by other scheduling tools:

- a) Keeping track of large number of long running jobs, as a single set.
- b) Supporting collaboration among multiple physicists.
- c) Conserving limited network bandwidth.
- d) Maintaining high availability.
- e) Placing jobs according to where the data is, as well as where CPU is available
- f) Tolerating partition failures that are common in wide-area networks.

Early this year the emphasis was on tailoring the system for the Caltech Tier2 prototype. For this, two features that were not in the initial implementation were added:

- a) Queuing of jobs when suitable processors are not available.
- b) Ability to run jobs under a different user ID given proper certification.

These features are currently implemented at a basic level. Further design/development/testing are ongoing.

This past month the system has been successfully installed on the (whole) Tier2 prototype. A single system manages processors that are located in two geographically distributed locations (Caltech and UCSD). The system is now being tested with CMS CMSIM/ORCA runs.

Looking to the future, three major items are being planned:

- a) Design and development of hierarchical servers to increase scalability.
- b) Studying data-aware allocation algorithms using the MONARC simulation system.
- c) Integration with the GDMP data mover system, and the Globus metacomputing toolkit.

8) *CMS Requirements Document: "CMS Virtual Data Requirements"*

This document, currently a work in progress, describes the long-term vision of CMS for the grid system it will use from 2006 onwards during LHC running. The document is intended as input for all Grid development projects (GriPhyN, EU DataGrid, PPDG) of which CMS is a customer'. The communication of a single coherent long term vision is an important element in the CMS efforts to help in establishing close synchronization and collaborative ties between these projects.

The document follows the computer science convention of only describing the requirements for, not the implementation of, the grid system. As such it is complementary to the more architectural documents produced in GriPhyN and the EU DataGrid. The main contribution made by the document is that it describes a (proposed) 'CMS virtual data grid system' in 2006 in great detail, including the data model, quantitative aspects, and exact interfaces between the Grid software components and the non-grid CMS software components. The description of these interfaces is an important step towards developing a view of the relation between the Grid projects and the CMS-internal software efforts.

The GriPhyN project timing has driven the creation of this document. A "pilot" version of this document, called 'CMS virtual data needs', was released for the December GriPhyN architecture meeting. In January 2001 it was concluded that major additional work was needed to complete the full document, now called 'CMS virtual data requirements'. Some other planned work was pushed forward in time because of this. It was decided that the timely creation of a complete CMS GriPhyN requirements document had higher priority, first because the many computer scientists in GriPhyN and the EU DataGrid urgently needed documentation that explained LHC physics requirements to them in computer science terms, and second because the timely availability of a complete document would allow for much more effective discussions in the EU DataGrid Workshop in March 2001 and the GriPhyN all-hands meeting in April 2001.

At the end of January 2001 a major draft (V3) was released inside CMS for review. On February 19 the first completed draft (V6), also incorporating CMS comments on V3, was released to the GriPhyN and EU DataGrid projects. The next revision is foreseen to be ready in early April, between the EU DataGrid Workshop and the GriPhyN all-hands meeting. The final version will be released after the all-hands meeting.

The document also incorporates comments and feedback from many CMS/GriPhyN participants in the time period Dec 2000 to March 2001. Other important sources are early results from the CERN Hoffmann review and results of the MONARC, RD45, and GIOD projects.

9) *Modeling CMS Grid Workloads*

As a companion document to the requirements document described above, a detailed model of a CMS grid workload, including both production and "chaotic" physics analysis, is being developed. The core model, called HEPGRID2001, is object based, using the virtual data concept of GriPhyN. By extending this model with a baseline simulation of an object-to-file mapping system, a file-based model will also be created. The grid projects are the main customers for these models.

The paper "HEPGRID2001: A Model of a Virtual Data Grid Application" (accepted for presentation at HPCN Europe 2001) documents this work. In addition to the paper there is a grid workload generator in C++. In comparison to the MONARC models, the HEPGRID2001 model contains less hardware details and more workload details.

10) *Collaboration with US and EU Grid Projects*

Significant time was spent in coordination activities between the US and the EU grid projects. In particular there have been contacts and coordination with the members of WP2, WP8, and the ATF of the EU DataGrid project, both in e-mail and during visits and conferences. There is also joint work on GDMP between Caltech and DataGrid WP2. CMS is increasingly organizing its Grid feedback activities to the grid projects to happen as a joint activity between all responsible CMS grid project participants both in the EU and the US.

A joint meeting of the Grid project managements was held in Amsterdam in March, with the goal of developing coordination and possibly joint management of the Grid projects, so that a single compatible Grid infrastructure and de facto standard software components are developed serving the LHC program as well as other HENP experiments. As a result of this meeting, one of us (HN) began work on formulating a Grid Project Coordination Framework, in collaboration with G. Wormser (IN2P3) and M. Mazzucato (INFN).

11) Technical Review of GriPhyN and EU DataGrid Architectures

Some time was spent reviewing the architectural documents being created in GriPhyN and the EU DataGrid, and in giving feedback to them. Major points that have been raised in the architectural feedback are as follows. First there is a need for close coordination between the Grid projects and for compatibility between their deliverables. Second, there is a need to develop further a vision of the more application-specific higher-level grid components which will be needed in vertically integrated HEP Grid systems. Third, a file-based grid does not match well to the CMS needs, and some way to deal in the grid with the finer granularity of events and persistent objects has to be developed.

12) Optimized Tags on Proto-Tier2

A new "sliced tag" implementation of HepODBMS was finished and successfully installed and run at the Caltech proto-Tier2. A detailed performance analysis of the current tag implementation with the sliced tags will follow. The results will be submitted to CHEP01.

The first part of the integration of Bitmap Indices based on HepODBMS tags is finished and installed at Tier2. We expect that Bitmap Indices combined with sliced tags my result in better performance for certain queries which only reference a small number of attributes of the tag. Benchmarks will be based on synthetic and on real CMS tag data and will be performed in April.

In addition some initial planning for a prototype tag analysis server for remote analysis on the Proto-Tier 2 was done.

=====

[1] http://monarc.web.cern.ch/MONARC/sim_tool/Publish/SONN/note01_009.pdf

[2] http://monarc.web.cern.ch/MONARC/sim_tool/Publish/SONN/

[3] http://monarc.web.cern.ch/MONARC/sim_tool/Publish/CMS/RecRep/

[4] http://monarc.web.cern.ch/MONARC/sim_tool/Publish/CMS/SelRep/

D0

There has been little specific D0 PPDG effort this quarter. Efforts have concentrated on preparing the new SCiDac proposal to DOE, attendance of the First Global Grid Forum in Amsterdam and following the EU DataGrid Workshop.

The D0 SAM team has been concentrating on preparations for the start of Run II data taking (<http://www.fnal.gov/pub/news/tevplot.html> <http://www.fnal.gov/pub/news/dzerocoll.html>) and support of the local and remote users. Work relevant to PPDG:

- a) Support for Monte Carlo production at NIKHEF. Bbftp has been integrated with SAM and deployed on the d0 Central Analysis System. Sustained file transfer rates of above 20 Mbit/sec on SURFNET are measured by our collaborators from NIKHEF to Fermilab [1]
- b) The first version of resource management in SAM has been released for use by the experiment. In it, SAM's data delivery infrastructure has been integrated with an abstract Batch System, with LSF as the first system being adapted [2]. The integration allows to coordinate user jobs dispatch with data delivery and perform fair share resource allocation in a flexible framework of benefit accounting. The experience we gain from the use and support of such integration this will feed into the PPDG proposal activity for definition of a formal Job Description Language.
- c) SAM Test Harness. We have developed a framework for the simulation of the distributed data delivery and SAM system. This is a set of scripts driven by configuration files that can start, run, stop and stimulate errors in the distributed set of SAM servers and daemons, run analysis projects, stimulate the delivery of raw data files, interface to the farm reconstruction job scripts etc. The Test Harness keeps comprehensive statistics and information about the errors encountered. It has been used to find problems within all layers of the system and is ready to be deployed on testbeds if they become available.

The hours accounted to PPDG for this quarter are minimal (~40 hours); The D0 PPDG work has in the main come from the base program. The Fermilab PPDG hours have come in general as part of the CMS contributions. We anticipate this will change in the coming quarter.

[1] From: "Kors Bos" <k.bos@nikhef.nl>
 To: "Willem van Leeuwen" <a03@nikhef.nl>; "Lee Lueking" <lueking@d0mino.fnal.gov>
 Cc: "Vicky White" <white@fnal.gov>; <veseli@fnal.gov>; <terekhov@fnal.gov>; <ruth@fnal.gov>; "Kors Bos" <bosk@atlas.nikhef.nl>
 Subject: RE: Numbers from sam/bbftp transfers
 Date: Tuesday, March 27, 2001 2:05 PM

Hi Lee,

Willem is not on-line now and will not be at NIKHEF tomorrow so let me try to give you some numbers which I have to quote from memory. Willem will give you more precise answers when he's back on-line.

We have seen an increase of a factor between 5 and 7 roughly in transfer speed. This corresponds with what we measured before: you go faster linearly up to about 7 or 8 streams and then it flattens off. Where we saw on the SURFnet statistics 4 Mbit/sec in the past we now see above 20 Mbit/sec. The transfer of a set of files (reco only) from one cpu with 200 tbar events with 2.0 min.bias takes roughly one minute now. As

far as we can tell there almost never is a problem with SAM on our side and the files go to d0mino but from there on there still are sometimes problems. But there you know as well as we do because we have always told you if we saw the files not moving into SAM but staying in the cache area.

[2]http://d0db.fnal.gov/sam/doc/design/resource_smbs_goals.html,
<http://d0db.fnal.gov/sam/doc/design/fairshare.html>

LBNL Scientific Data Management Group

Two activities took place during this quarter, one were enhancement to HRM, and the second the development of a basic minimal-DRM.

The HRM was packaged and modified for use by the CMS project at Caltech. The main modification was in eliminating the need for a pre-loaded file catalog. The HRM previously needed to use a file catalog in order to find the file_size of the file requested and its location on tape (tape_id). The size information is needed to verify that the file was properly staged to the HRM disk (i.e. eliminate errors). The tape_id information is needed in order to order the requests to HPSS to minimize tape mounts.

In the new version of HRM we use the HSI interface to get the tape_ID and file_size information. HSI is an alternative interface to HPSS developed by the PROBE project. By using HSI to get the tape_ID and file_size, the file catalog is built dynamically for the files requested from HRM. This makes the use of HRM simpler in that the file catalog does not have to be created ahead of time, or maintained.

The minimal DRM was developed and is now being tested. It supports permanent and volatile files, in that permanent file are not subject to release. Upon request of a file it pins the file. It has a time out associated with the pin. If another request for the same file is made, the time stamp gets updated and the new time out assigned.

The minimal-DRM currently supports only blocking calls, and provides no call backs. Also, it is not yet mutli-threaded to allow multiple file requests to be processed concurrently. These features will be added next.