

# The Particle Physics Data Grid Collaboratory Pilot

## Proposal in Response to SciDAC Announcements LAB 01-06 and LAB 01-11 and Grant Notices 01-06 and 01-11

### DOE Laboratory Contact:

Richard P. Mount, SLAC MS 97, 2575 Sand Hill Road, Menlo Park, CA 94025, Tel: 650 926 2467, Fax: 650 926 3329

### University Contact:

Miron Livny, Computer Sciences Dept., Room 5372, University of Wisconsin-Madison, 1210 West Dayton St., Madison, WI 53706, Tel. 608-262-4694, Fax: 608-262-9777

## List of Participants

### Computer Science Teams

Computer Science Department, University of Wisconsin

**Miron Livny**<sup>11</sup> (PI), Paul Barford<sup>11</sup>

Mathematics and Computer Science Division, Argonne National Laboratory

Ian Foster<sup>1</sup>, William Allcock<sup>1</sup>, Mike Wilde<sup>1</sup>

Scientific Data Management Group, NERSC, Lawrence Berkeley National Laboratory

Arie Shoshani<sup>5</sup>, Andreas Mueller<sup>5</sup>, Alex Sim<sup>5</sup>

San Diego Supercomputer Center

Reagan Moore<sup>6</sup>

### Physics Experiment Teams

ATLAS

Torre Wenaus<sup>2</sup>, Rich Baker<sup>2</sup>, Stewart Loken<sup>5</sup>, David Malon<sup>1</sup>, Ed May<sup>1</sup>, Razvan Popescu<sup>2</sup>, Larry Price<sup>1</sup>, Alex Undrus<sup>2</sup>, Alexandre Vaniachine<sup>1</sup>

BaBar

**Richard Mount**<sup>7</sup> (PI), Robert Cowles<sup>7</sup>, Andy Hanushevsky<sup>7</sup>, Adil Hasan<sup>7</sup>

CMS

**Harvey Newman**<sup>3</sup> (PI), James Amundson<sup>4</sup>, Paul Avery<sup>11</sup>, Lothar Bauerdick<sup>4</sup>, James Branson<sup>9</sup>, Julian Bunn<sup>3</sup>, Ian Fisk<sup>9</sup>, Gregory Graham<sup>4</sup>, Takako Hickey<sup>3</sup>, Koen Holtman<sup>3</sup>, Iosif Legrand<sup>3</sup>, Vladimir Litvin<sup>3</sup>, Vivian O'Dell<sup>4</sup>, James Patton<sup>3</sup>, Asad Samar<sup>3</sup>, Conrad Steenberg<sup>3</sup>

D0

Ruth Pordes<sup>4</sup>, Lee Lueking<sup>4</sup>, Wyatt Merritt<sup>4</sup>, Igor Terekov<sup>4</sup>, Sinisa Veseli<sup>4</sup>, Rich Wellner<sup>4</sup>

STAR

Matthias Messer<sup>2</sup>, Bruce Gibbard<sup>2</sup>, Eric Hjort<sup>5</sup>, Doug Olson<sup>5</sup>

Thomas Jefferson National Accelerator Facility (JLAB)

Chip Watson<sup>8</sup>, Ian Bird<sup>8</sup>, Ying Chen<sup>8</sup>

### Liaisons

GriPhyN Project – Paul Avery, University of Florida

### Institutions

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

# The Particle Physics Data Grid Collaboratory Pilot

## Table of Contents

List of Participants .....	i
Abstract .....	iii
1 The Particle Physics Data Grid Collaboratory Pilot .....	1
1.1 Background and Significance .....	1
1.1.1 Large-Scale Collaboration on Large-Scale Collaboratory Technology .....	1
1.2 Preliminary Studies.....	4
1.2.1 Major Precursor Projects leading to PPDG.....	4
1.2.2 High-Speed Transport .....	6
1.2.3 Requirements identified by experiments and their implications for Computer Science tasks.....	6
1.3 Research Design and Methods .....	8
1.3.1 Data Grid Overview and Architecture.....	8
1.3.2 Development and Deployment Process .....	9
1.3.3 Computer Science Program of Work .....	10
1.3.4 HENP Experiments Program of Work .....	14
1.3.5 Work Plan & Schedule.....	22
1.3.6 Computer Science High Speed Networking Testbed .....	25
1.3.7 PPDG Management Plan .....	25
1.3.8 Outreach and Education .....	27
1.3.9 Budget summary and justification.....	27
1.4 Subcontract or Consortium Arrangements .....	28
2 Literature Cited .....	29
3 Budget	
4 Other Support of Investigators	
5 Biographical Sketches	
6 Description of Facilities and Resources	
7 Appendix	

# The Particle Physics Data Grid Collaboratory Pilot

Scientific Discovery through Advanced Computing (SciDAC):  
National Collaboratories and High Performance Networks, LAB 01-06;  
High Energy and Nuclear Physics Research, LAB 01-11

## Abstract

The Particle Physics Data Grid (PPDG) Collaboratory Pilot will develop, acquire and deliver vitally needed Grid-enabled tools for data-intensive requirements of particle and nuclear physics. Novel mechanisms and policies will be vertically integrated with Grid middleware and experiment-specific applications and computing resources to form effective end-to-end capabilities. PPDG\* is a collaboration of computer scientists with a strong record in distributed computing and Grid technology, and physicists with leading roles in the software and network infrastructures for major high-energy and nuclear experiments. Together they have the experience, knowledge and vision in the scientific disciplines and technologies required to bring Grid-enabled data manipulation and analysis capabilities to the desk of every physicist. A three-year program is proposed, taking full advantage of the strong driving force provided by now running physics experiments, ongoing Computer Science projects and recent advances in Grid technology. Our goals and plans are ultimately guided by the immediate, medium-term and longer-term needs and perspectives of the physics experiments, some of which will run for at least a decade from 2006 and by the research and development agenda of the CS projects involved in this Collaboratory Pilot and other Grid-oriented efforts. PPDG is actively involved in establishing the necessary coordination between complementary Data Grid initiatives in the U.S., Europe and beyond.

---

\* [www.ppdg.net](http://www.ppdg.net)

# 1 The Particle Physics Data Grid Collaboratory Pilot

## 1.1 Background and Significance

Experimental research in fundamental physics is an important mission of DOE, as is the development of the information technology required to meet the science mission of the department. Never before have particle and nuclear physics research had so much to gain from state of the art information technology. Collaborations of hundreds to thousands of physicists and engineers are formed to create accelerators, detectors and analysis systems with a productive life of tens of years. These analysis systems form a complex and widely distributed fabric of computing and storage resources. Currently the world's leading operational facilities in high-energy and nuclear physics are PEP-II/BaBar at SLAC<sup>1</sup>, using electron-positron collisions to investigate the small asymmetry between matter and antimatter that gave rise to our matter-dominated universe, RHIC<sup>2</sup> at Brookhaven, using nuclear collisions to create energy densities typical of the big bang, and CEBAF at Jefferson Lab<sup>3</sup> using electron beams to reveal the partonic structure of the nucleus. In the immediate future the Fermilab Tevatron will join this group and allow the CDF and D0 experiments to address new physics frontiers. Within five years the ATLAS and CMS experiments at the CERN LHC<sup>4</sup>, in which the U.S. role is the largest of any single nation, will measure collisions at unprecedented energies that are expected to revolutionize our understanding of physics. The computing facilities for BaBar currently support analysis of a 200 terabyte object database using over one hundred data servers and close to two thousand compute servers. The facilities required for each of RHIC, CDF, D0, ATLAS and CMS will be even larger.

The non-deterministic nature of quantum physics, uneasily understood during the last century, inevitably requires the measurement and analysis of billions of particle interactions to observe and understand fundamental processes. Typical current detectors, weighing over ten thousand tons, must have millions of sensitive channels and produce hundreds of terabytes of data per year. LHC experiments will rapidly reach tens of petabytes of data to be analyzed. The design, construction and data analysis for each experiment requires the combined intellect and dedicated work of international collaborations. For over two decades, particle physics collaborations have developed applications that stretched collaboratory technology to its limits, in several cases driving the creation of national and international networks.

The Computational and Data Grid community<sup>5</sup> has identified support of "Virtual Organizations" (VO) as the driving force behind the creation of Grid environments<sup>6</sup>. A High Energy or Nuclear Physics (HENP) experiment with its complex fabric of computing, networking and storage resources, demanding applications and dynamic multi-institutional structure can be viewed as an archetypical large VO. For a member of a VO to benefit from the resources of a Grid, his/her application has to be vertically integrated with the fabric of the Grid via a stack of Grid-enabled middleware.

In coordination with complementary projects in the U.S. and Europe, this proposal is aimed at meeting these urgent needs for advanced Grid-enabled technology and strengthening the collaborative foundations of experimental particle and nuclear physics. Our research and development will focus on the missing or less developed layers in the stack of Grid middleware and on issues of end-to-end integration and adaptation to local requirements. Each experiment has its own unique set of computing challenges, giving it a vital function as a laboratory for CS experimentation. At the same time, the wide generality of the needs of the physicists for effective distributed data access, processing, analysis and remote collaboration will ensure the more general applicability of the Grid technology that will be developed and/or validated by this proposed collaboratory pilot.

### 1.1.1 Large-Scale Collaboration on Large-Scale Collaboratory Technology

The PPDG collaboration is pursuing a pragmatic approach to the building of this and next generation end-to-end data analysis systems for HENP experiments using state of the art Grid technology as the underlying infrastructure.

High Energy and Nuclear Physics Experiments taking data *now* will rely on our Grid enabled software deliverables. Therefore, we must ensure that sufficient manpower is available, not only to do the necessary research and development, but also to integrate and deploy the software in a robust and efficient production system that is capable of handling the experiments data. Much of the work of the PPDG collaboratory pilot will leverage Grid software developed as research projects through other DOE funding channels. PPDG will provide the CS projects involved in the collaboration active testbeds and a technically knowledgeable and experienced user community. This community will insist that "their" software supports a smooth and effective round the clock operation. Not only are

HENP scientists known for discovering the tiniest particles of matter, they have a reputation for using innovative technology to achieve their science, and for putting the necessary intellectual and technical effort into deploying robust and functional systems to meet these needs.

Computer Scientists are leading the way in the research and development of novel information technologies to transform our society and the way we do computing. However, they need advanced applications to guide, test, evaluate and validate their work. Such applications will also help the Computer Scientists to bring their research “to market” for the general use. Realistic and challenging “use cases” and requirements of cutting-edge applications stimulate and guide the CS research and development towards “real-life” problems. Users whose career objectives rely on this software will stimulate developments that work in “real-life” environments and are maintainable. If these users are scientists the developments must, by the very nature of the objectives, be flexible, extensible to the edge of their potential and adaptable to changing needs and technologies.

Our collaboration brings the CS and physics disciplines together working closely and synergistically to benefit both sciences. We are fortunate to have a broad collaboration of six experiments and four Computer Science projects. This gives us a unique opportunity to guide, adapt and reuse our developments towards multiple applications. To promote rapid turnaround of our planned deliverables, the initial development of each integrated capability is planned on a schedule of one year or less and will be delivered to one (or a few) of the six experiments. After an initial period of production running, common Grid technologies and software components will be offered to the other experiments, with adaptations and generalizations developed as needed. By the end of the three years of the pilot, we expect many of the PPDG deliverables will have been used by four or more experiments, thus ensuring the utility and appropriateness of our Grid technology for more general common use. Both sciences will publish the results of their PPDG work, and will demonstrate both the research and the practical application of the results.

We believe that the scale of the proposed budget is justified by the opportunity to leverage the work on collaborative developments funded by the physics programs and to take advantage of the massive computing, storage and networking fabric serving the experiments. PPDG will place people in each of the six experiments just sufficient to have a decisive influence on the way the experiment exploits the Grid and to ensure that work within the experiment is made available to all PPDG members. The experiments intend to contribute at least as much additional effort to the PPDG program. By including six major experiments with differing timescales, needs and cultures, PPDG will be strongly driven to develop solutions of great generality.

On the computer science side, the PPDG budget will also be highly leveraged. For the Wisconsin group, funding is proposed at a level allowing significant developments in areas where no adequate middleware exists. We propose funding for the ANL, LBNL and SDSC groups at a level allowing collaboration with experiments on adapting, deploying and ‘hardening’ existing middleware, but only minimal new developments. Leveraged efforts will come from these groups’ involvement in the GriPhyN project<sup>7</sup>, the European DataGrid<sup>8</sup>, and, with the hoped for SciDAC funding, from the success of their proposals for work of great value to PPDG including:

- “A High Performance Data Grid Toolkit: Enabling Technology for Wide Area Data-Intensive Applications”, PIs: Ian Foster (ANL), Carl Kesselman (ISI)
- “Storage Resource Management for Data Intensive Applications”, PI Arie Shoshani (LBNL)
- “Security and Policy for Group Collaboration”, PIs: Steve Tuecke (ANL), Carl Kesselman (ISI)
- “Scientific Data Management Enabling Technology Center”, PI Arie Shoshani (LBNL)
- “DOE Science Grid Collaboratory Pilot”, PIs W. Johnston (LBNL), I. Foster (ANL)

The proposed rise in project funding in FY2002 and FY2003 matches the needs of the work plan and strategy presented here. Middleware applicable to all or many experimental needs will be first tried on one experiment. As the project progresses, most of the project’s suite of tools will come under integration and test with larger numbers of experiments, requiring more total effort. By the third year of the project, these activities will extend to other scientific users and interested industrial partners.

#### **1.1.1.1 LHC experiments – ATLAS and CMS**

The Large Hadron Collider (LHC) at CERN will open a new frontier in particle physics due to its higher collision energy and luminosity compared to existing accelerators. The two general-purpose detectors, ATLAS<sup>9</sup> and CMS<sup>10</sup>, are being constructed by large collaborations, each bringing together almost 2000 scientists from 150 institutes around the world. The experiments will exploit the full range of physics made accessible for the first time by LHC’s combination of high luminosity and 14 TeV center of mass energy. While ATLAS and CMS employ quite different

detector techniques, emphasizing different aspects of particle identification and measurement, both designs maximize the discovery potential for new physics such as Higgs bosons and supersymmetric particles, while optimizing the capability of high-accuracy measurements of known objects such as heavy quarks and gauge bosons. The detectors employ precision tracking, high resolution calorimetry, and muon measurement over a large solid angle to accurately identify and measure electrons, muons, photons, jets and missing energy.

Both CMS and ATLAS will filter in real time the data from the 40 million collisions/sec produced by the LHC, using dedicated hardware in the first level trigger (output 75 kHz), and farms of about 1000 online processors, to record events at a rate of approximately 100 Hz, accumulating 100 MB/sec of raw data. This will yield 10 PB of stored raw and processed data in the first year of operation, starting in 2006. The data volume is expected to increase rapidly, so that the accumulated data volume will reach 1 Exabyte (1 million Terabytes) by around 2015.

The scale of the computing challenge presented by the LHC experiments is unprecedented in terms of data volume, processing requirements, and the scale and distributed nature of the analysis and simulation tasks among thousands of physicists worldwide.

Grid technologies are essential to fully realizing the potential of the ATLAS and CMS research program. They enable a collaboration-wide computing fabric that can deliver the capability for full participation in the development and execution of the experiments' research program on the part of physicists at their home institutes. This is particularly true for distant participants such as those in the U.S. Meeting the demands of LHC analysis and simulations via a highly distributed, hierarchical computing infrastructure exploiting Grid technologies is a central element of both the CMS and ATLAS worldwide computing models.

While LHC data taking will not begin until 2006, both ATLAS and CMS already have large and highly distributed computing and software operations. These operations serve immediate and near term needs such as test beam data analysis, detector performance and physics simulation studies that support 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.

#### **1.1.1.2 BaBar experiment**

The BaBar experiment<sup>1</sup> using the PEP-II Asymmetric B Factory at SLAC is currently the world's most data-intensive scientific experiment. PEP-II is a two-ring collider, colliding 9 GeV electrons with 3.1 GeV positrons, energies chosen to maximize production and decay studies of B meson particles. It is engaged in a multi-year program to use billions of B mesons to study the minute asymmetry between matter and antimatter that has shaped our universe. BaBar acquired its first data in May of 1999 and in little more than a year PEP-II was delivering collisions to BaBar at rates above its design specifications. SLAC plans to increase the rate of collisions by at least a factor 10 in the next three years requiring the analysis of petabytes of data. The 500-physicist BaBar collaboration is international with participation almost equally divided between North America and Europe.

#### **1.1.1.3 D0 experiment**

The D0 detector<sup>11</sup> is a complex, nearly hermetic detector designed to observe proton-antiproton collisions of 2 TeV at the Fermilab Tevatron Collider. The detector is designed to focus on good identification at high  $p_T$  for leptons and jets, including b jets, and to measure missing  $E_T$  with good resolution. With these capabilities, the experiment plans a two-phase physics program over the next 7 years that will explore a variety of physics searches and measurements, including the search for the Higgs boson. The data-handling problem is challenging, with total data sets of 0.3 PB/year in the first phase and approximately 1 PB/year in the second phase. The collaboration is multinational, with more than 500 collaborators from 65 institutions worldwide. The physics analysis capabilities depend strongly on the computing tools and data handling tools that can be provided for this large data set.

#### **1.1.1.4 STAR experiment**

The primary motivating physics goal for the entire RHIC scientific program is to discover and characterize the properties of a phase of matter called the Quark-Gluon Plasma (QGP). The QGP is predicted by the standard model of particle physics (Quantum Chromodynamics) to have existed ten millionths of a second after the Big Bang. By colliding heavy ions (up to Au) at center-of-mass energies up to 200 GeV/nucleon the RHIC accelerator is expected to produce this state of matter momentarily and repeatedly. STAR<sup>12</sup> searches for signatures of quark-gluon plasma formation and investigates the behavior of strongly interacting matter at high energy density by focusing on measurements of hadron production over a large solid angle. It utilizes large volume Time Projection Chambers (TPC) for tracking and particle identification in a high track density environment. STAR will measure many observables simultaneously, on an event-by-event basis, to study signatures of a possible QGP phase transition and

the space-time evolution of the collision process at their respective energy. The goal is to obtain a fundamental understanding of the microscopic structure of hadronic interactions, at the level of quarks and gluons, at high energy densities.

#### **1.1.1.5 Thomas Jefferson National Accelerator Facility experiments**

In the traditional view, the atom's nucleus appears as a cluster of nucleons—protons and neutrons. A deeper view reveals quarks and gluons inside the nucleons. At the Thomas Jefferson National Accelerator Facility (Jefferson Lab or Jlab)<sup>3</sup>, energetic beams of electrons let physicists examine how the two views fit together. Ultimately, the process of bridging the views will yield a complete understanding of ground state nuclear matter - 99.5% of the content of our world. We will know more clearly how matter itself is put together, and how it gets its characteristic properties.

With experiments in three halls focusing on such fundamental topics as quark confinement, the proton spin crisis, and gluon excitations, Jefferson Lab has become one of the world's leading laboratories through its unique capabilities for exploring and elucidating the quark structure of matter. As one example, for more than 20 years it has been assumed, based on the available data, that the charge and magnetization distributions in the proton were proportional to one another (corresponding to  $\mu\text{G}/\text{Gm}=1$ ). New data from Hall A shows that this is not true, and is leading to a re-examination of the dynamics governing the proton's quark wavefunctions.

Jefferson Lab has an extremely active experimental program with over 1000 users from around the world. The machine capabilities coupled with advanced detector technology make Jefferson Lab's CEBAF a forefront research tool acquiring data at the rate of 1 terabyte per day.

## **1.2 Preliminary Studies**

The Particle Physics Data Grid collaboration was formed two years ago because its members were keenly aware of the need for Data Grid services to enable the worldwide distributed computing model of current and future high-energy and nuclear physics experiments. Initially funded from the NGI initiative and later from the DOE MICS and HENP programs, it has provided an opportunity for early development of the Data Grid architecture as well as evaluating some prototype Grid middleware<sup>13</sup>.

Early meetings and discussions within PPDG led to the description of an architecture for accessing and moving data in a Grid that includes components for request management and execution, storage resource management, replica and metadata catalogs. Prototype components, developed or adapted from existing middleware, were used to test the architecture in tests including Wisconsin, FNAL, ANL, LBNL and SDSC. This contributed directly to what has now been refined by the GriPhyN project to be called the Data Grid Reference Architecture<sup>14</sup>.

In addition to detailed technical discussions and tests, the previous two years have included significant cultural and high-level computing requirements knowledge exchange between the physics experiments and the Computer Science groups. It is a significant accomplishment that now, based on these preliminary exchanges and work, our collaboration has the motivation and mutual trust to move ahead to actually deploy Grid services in the mission critical activities of the experiments.

### **1.2.1 Major Precursor Projects leading to PPDG**

PPDG has benefited from strong participation by leaders of innovative projects that have pioneered in both concepts and practice of distributed computing and Grids. Among them are the three projects described briefly in the following sections.

#### **1.2.1.1 Globus**

The Globus project<sup>15</sup>, a joint effort of Argonne National Laboratory, the Information Sciences Institute of the University of Southern California, and the University of Chicago, has been working for the past five years to solve precisely the problems faced by the project proposed here—facilitating scientific collaboration within flexible “virtual organizations” by connecting globally dispersed collaborators to complex and large-scale instrumentation, data, computing, and visualization resources<sup>16</sup>. The results of the Globus project are being studied, developed, and enhanced at institutions worldwide to create new Grids and services, and to conduct computing research.

Globus components provide the capabilities to create “Grids” of computing resources and users; track the capabilities of resources within a grid; specify the resource needs of user's computing tasks; mutually authenticate both users and resources; and deliver data to and from remotely executed computing tasks. Globus is distributed in a modular and open “toolkit” form. This makes it easy to integrate services into scientific environments and

applications. Globus has been integrated with technologies such as the Condor high-throughput computing environment<sup>17,18</sup> and the Portable Batch System (PBS) job scheduler<sup>19</sup>. Both of these integrations demonstrate the power and value of the open protocol toolkit approach, and offer significant opportunities for constructing the PPDG collaborative. Experience with projects such as the NSF's National Technology Grid<sup>20</sup>, NASA's Information Power Grid<sup>17</sup> and DOE ASCI's Distributed Resource Management Project<sup>21</sup> have provided considerable experience with the creation of production infrastructures.

The Data Grid community has started to address some of the requirements associated with distributed management and analysis of large-scale data. Over the past two years, the Globus PIs have played a leadership role in establishing a broad national—and indeed international—consensus on the importance of Data Grid concepts and on specifics of a Data Grid architecture. A tightly coordinated set of projects has been established that together are developing and applying Data Grid concepts to problems of tremendous scientific importance, in such areas as high energy physics, astronomy, and climate science. For example, in addition to the current project, which is focused on the application of Data Grid concepts to the needs of a number of U.S.-based high energy and nuclear physics experiments, the Earth System Grid project is exploring applications in climate and specific technical problems relating to request management, the NSF-funded GriPhyN<sup>7</sup> project is focused on the automatic generation and management of derived data, and the EU-funded European Data Grid (EDG) project is focused on the development of an operational Data Grid infrastructure. All four of these projects have adopted a common Globus-based infrastructure.

### 1.2.1.2 Condor

The Condor project<sup>22</sup> at the University of Wisconsin-Madison has been engaged in distributed systems research for more than fifteen years. Guided by a novel viewpoint of how distributed systems are built, used and operated the project has pioneered the concepts of *distributed ownership* and *high throughput computing*. The cornerstone of the Condor approach to distributed computing is the *matchmaking* framework that enables consumers and providers of services in a distributed environment to locate each other. The framework is based on the powerful *ClassAd* language<sup>23</sup> that enables each party to describe itself and the party it would like to be matched with. Unlike most other scheduling systems where jobs and resources are pre-assigned to *queues*, matchmaking does not impose any such restrictions. Requests and offers to provide services are self-described and are free to move around and be matched. The results of this research activity have been translated to robust and effective mechanisms and policies that are deployed in Condor systems all over the world. Condor has been in production use at the University of Wisconsin since 1986. Today, Condor delivers distributed resource management services at more than 150 academic and commercial sites. At Wisconsin alone, Condor manages more than 1,400 CPUs and serves users from a wide spectrum of disciplines. The success of the Grid enabled version of Condor – Condor-G, is a clear display of the power of the Condor architecture and its ability to exploit the inter-domain services offered by Globus to give an end user a uniform view and a single access point to a large collection of Grid resources.

### 1.2.1.3 HRM and SRB

Over the previous year, we have demonstrated in a prototype system the use of the Hierarchical Storage Resource Manager<sup>24</sup> (HRM) with the Storage Resource Broker<sup>25</sup> (SRB). HRM is a middleware server that interfaces to HPSS to manage file staging requests for staging files into an HRM managed disk cache. An interface was developed from the SRB server to the HRM system to perform file staging requests. When a file is cached by the HRM, the SRB server is notified, and it takes care of transferring the file to its destination upon the client's request. Consequently, two new functions were added to SRB, a "stage" call, and a "status" call. The "stage" call allows the user to request pre-staging of files from HPSS tape to the HRM disk, and requesting the file at a later time. The "status" call is used to return to the SRB user the information provided by HRM on the status of the file, such as the time to completion of the staging request.

Most of the capabilities available today in HRM were originally developed as part of the STACS system<sup>26</sup>. STACS is a system developed under the Grand Challenge program<sup>27</sup> and is now deployed and used by the STAR experiment. Therefore, HRM is adapted from a well-tested stable component of the STACS system and was packaged to work as a Grid enabled middleware module. It accepts URLs of the files requested, accessing HPSS to stage the files to its local disk, and calling back the requesting component when a file is staged. In addition to pre-staging and call back capabilities, HRM provides the client with status capability estimating the time till staging will be done.

HRM was enhanced to provide a more general Grid interface. The HRM CORBA interface was enhanced with joint work between LBNL and Fermilab. Fermilab provided the same HRM interface to the Grid on top of their SAM system<sup>28</sup>. This demonstrated the generality of the approach, where two completely different systems provided pre-

staging to files from two different mass storage systems – HPSS and Enstore. The capabilities of the HPSS-HRM were also enhanced for its use in the CMS experiment. Specifically, in the past HRM relied on a “file catalog” that contained information about the tape ID that HPSS assigned to a file as well as the file size. The new enhancements now use a newly developed HPSS access module called HSI to extract this information dynamically for requested files. This made HRM more general and applicable to multiple experiments that use HPSS.

A version of an “on-demand” Disk Resource Manager (DRM) was developed as part of the STACS system. From this experience we gained insight on the caching policies required to manage the cache. However, this was not developed as a separate module, but rather it was developed as an integral part of the job scheduler. Recently, we started to design the functionality and the interfaces to the (on-demand) DRM as part of the current PPDG project. An early version of this DRM has been developed. We will have experience using this DRM before we need to deploy it for this proposal. This will be an iterative process as a result of deployment in real experiments.

On the application side we developed capabilities to manage the staging of files accessible via SRB and GridFTP to a locally managed disk cache. A request to process a set of files is managed by a special module capable of interacting with the remote storage, local storage and the local processing environment. The capabilities of this module have been demonstrated in production runs of CMS simulations. Condor resources at the University of Wisconsin were used to generate simulated events that were archived in the HPSS system at Caltech. This is an example of a vertically integrated capability that integrates a CMS application (CMSIM) with computing (Condor pool) and storage (HPSS and disk caches) fabric via Grid middleware (SRB and GridFTP).

### **1.2.2 High-Speed Transport**

There has also been development and tests of high-speed site-to-site file transfer with a goal of achieving 100 MB/sec transfer rates. Over the past two years, we have developed high-speed transfer library, GridFTP, and replica catalog and management mechanisms<sup>29</sup>. Using Grid Security Infrastructure (GSI) mechanisms<sup>30,31</sup> for authentication, they provide key building blocks for the management of data replication and movement in a Data Grid environment. Most of the testing was carried out between Caltech and SLAC using NTON. The culmination of this effort was achieved and demonstrated as Super Computing 2000 where peak rates of 1 Gbps were achieved transferring data from the SC2000 floor in Dallas to LBNL. In addition, this group achieved sustained cross-country transfer rates of more than 500 Mb/s, and was awarded the “Hottest Infrastructure” award in the Network Challenge event. These components are now being used in several HEP tools and experiments, including GDMP and some experiments at BNL.

### **1.2.3 Requirements identified by experiments and their implications for Computer Science tasks**

As a result of a series of discussions between the collaborating HENP experiments and CS groups, we identified eight requirements in common to most experiments, that stem from the four basic computing activities of the HENP community: i) data archiving, ii) simulating events, iii) event reconstruction, and iv) analysis. All the requirements described below are driven by a common underlying assumption shared by all the experiments; only a fully distributed computing environment can meet the data processing and access needs of their international collaborations. They all face the challenge of operating their computing environments as a robust and effective production service in face of all the dynamic uncertainties of performance and availability inherent to such decentralized Grid environments. From the outside everyone expects the computing facility of each experiment to look and feel like one very powerful and reliable computer. However, under the cover due to physical, economical and administrative forces these computing facilities are composed of a very large collections of distributively owned and locally managed computing, storage and communication resources that are scattered around the world. We describe each of the eight specific requirements, and delineate the planned CS research and development areas to support the needs of the first six. Given the existing Grid middleware and the immediate needs of the experiments, we decided that the last two requirements should be considered to be as longer term, and are therefore given lower priority in the first years. This review led to the definition of seven Computer Science work areas that we plan for PPDG activities. These are described in a later section.

#### **1.2.3.1 Continuous and reliable (24-7) data processing (reconstruction), data archiving, and data generation (simulation)**

This is a fundamental activity for experiment data and must operate as robustly and smoothly over the globally distributed system as within a central computer center environment. While this is a very demanding requirement for system located in one machine room and managed by a one organization, our experience so far has demonstrated that it is a much harder problem to maintain a 24-7 operation when resources are physically distributed and locally owned and managed. In order to meet this ambitious goal users and system administrators must be given ways to

declaratively specify the overall “production” process, and have this process manage the movement of the raw data to buffers, the reconstruction processing and the archiving of the raw and reconstructed data on tape.

In order to support this task the following tools and capabilities are needed: a job description language to specify (declaratively) the distributed process; a scheduler to manage the job flow using Grid resources, a replica catalog to keep track of the existence, availability and location of files/objects, storage resource managers to allocate space and release files from disk buffers dynamically, and schedulers for processing farms and file transfer services. Monitoring is an integral part of this activity as software modules (agents) and humans must be aware of the state of the distributed fabric (disks, tape silos, CPUs, networks) and applications (simulations, reconstruction, data movers) involved in the process. Each of the sub systems/processes may have its own monitoring capabilities. Some aspects of its status must be communicated to the higher levels of the control and management system.

There is a similar activity for the generation and reconstruction of simulation data in a distributed environment. However, this may not have the same stringent real time requirements as dealing with the raw data. Yet each site would benefit from the same tools to automatically manage a continuous simulation data generation and reconstruction process. Also, the two activities may share the same processing, communication and storage resources.

#### **1.2.3.2 High level specification of data distribution requests and policies**

This need comes from the practice of specifying a policy for where data should be stored and what part of the data needs to be replicated when and where. The replication requirements are usually for reconstructed data or the results of analysis jobs, but often some small percentage of raw data (and/or simulation) files are distributed across several sites as well. This process, also referred to as “file distribution service”, is usually specified based on a period of time (runs), but also based on properties of the data (streams) or usage patterns. This specification must be syntactically rich enough and easy enough to specify so that eventually the data appears transparently, and is universally available to the individual user.

In order to support this task the following tools and capabilities are needed: a job description language to specify (declaratively) the desired placement and replication actions, and a scheduler that invokes the replica management service and storage management modules.

#### **1.2.3.3 Reliable replica management**

Reliable replication is the service that invokes and manages the file transfer and insures that the replica catalog is updated properly even in the presence of network or system failures. It is important to note here that any reliable data transfer involves a step where two copies of the file exist – one at the source and one at the destination. As above, this service must robustly distribute and make the data available to the individual physicist in any time zone, and whatever the capabilities of the local system or interconnecting network.

In order to support this task the following tools and capabilities are needed: a job description language to specify the desired replication process, a scheduler that invokes the replica management service and storage resource management modules, and a reliable and recoverable replica catalog that supports reliable and recoverable remote create/delete/update transactions.

#### **1.2.3.4 Efficient and reliable file transfer**

This is, of course, a fundamental requirement that all experiments can benefit from. It is invoked by other services as needed for file/data movement across Grid storage, staging and caching devices. The efficiency requirement is the dynamic adjustment of window sizes and parallel streams based on the capacity and latency of the network link. The reliability requirement is to insure that a file transfer is an atomic operation and is fully integrated with the resource allocation and management services of the Grid fabric. In case of failures this service needs to remove partial files from the destination and re-transfer files. The file transfer service should be invoked only after space allocation is obtained and access control restrictions are checked.

#### **1.2.3.5 Storage management at site, both disk and tape**

This requirement comes from the need of each site on the Grid to manage its own storage (disk and tape) and local I/O bandwidth resources. A disk resource manager needs to adhere to (and enforce) a local policy specified by the site administrator for allocating space (and bandwidth) and revoking allocations when needed. A tape resource manager needs to throttle the use of the tape resources for both reading and writing. This service can be called by other Grid middleware services such as the scheduler or a replica manager to request resource allocation. Like any other service on the Grid, this service has to be accessible via reliable and recoverable APIs that can be invoked

remotely and support distributed transaction semantics as several such managers may be involved in a single data transfer. The experiment data handling applications need to be able to use a single interface to the data storage system – whatever system the distributed sites have locally available.

#### **1.2.3.6 Coordinated storage and computation allocation**

This requirement is to support the dynamic scheduling, flow control, and monitoring of tasks specified by job control or ad hoc analysis requirements across the globally distributed system. Its function is to coordinate the allocation and management of compute resources, communication resources and storage resources to match the needs of large number of simultaneous clients and to follow local and global policies. This service uses other services as needed, such as requesting storage allocation from storage resource managers, requesting file replication from reliable replica management services, or temporary file transfers from the reliable file transfer service.

#### **1.2.3.7 Support of a metadata catalog**

This is a requirement for the support of a distributed analysis environment. It provides a mapping from an attribute-based request (such as energy level of events and/or the number of particle produced by the event) into a set of files needed for the analysis. This activity is not emphasized in this proposal for 3 reasons: i) many experiments already have some way of dealing with this using “tag databases”; ii) many experiments pre-partition the data into smaller replicated subsets, and request entire subsets; iii) we notice that other activities address this need, notably the proposed SciDAC Scientific Data Management Enabling Technology Center and the GriPhyN project. However, this capability is needed for the support of ad hoc requests, which are essential for more flexible analysis, and we intend to incorporate this into our future plans.

#### **1.2.3.8 Estimation of request execution**

This is an important requirement needed to eliminate requests that require more time or resources than a client wishes or is allowed by the policy of the experiment to spend or utilize. An accurate estimation feedback may cause a researcher to change his/her request and affect the actions of the schedule and resource allocation services. However, an accurate estimation in a distributed Grid system is very complex task that will require much experience with running actual systems. Therefore, we decided that as a first step we will design every component to support estimation of services they provide. With this basic capability, and experience with deployed systems, one can attempt to generate a global estimate per request.

### **1.3 Research Design and Methods**

Now that Data Grid concepts are maturing and a broad interest and level of activity has developed around the world<sup>32</sup>, 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 Architecture<sup>33</sup> 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.

#### **1.3.1 Data Grid Overview and Architecture**

The PPDG project will work within the Data Grid architecture as defined by Deelman, Foster, Kesselman and Livny<sup>33</sup>. To quote from that document:

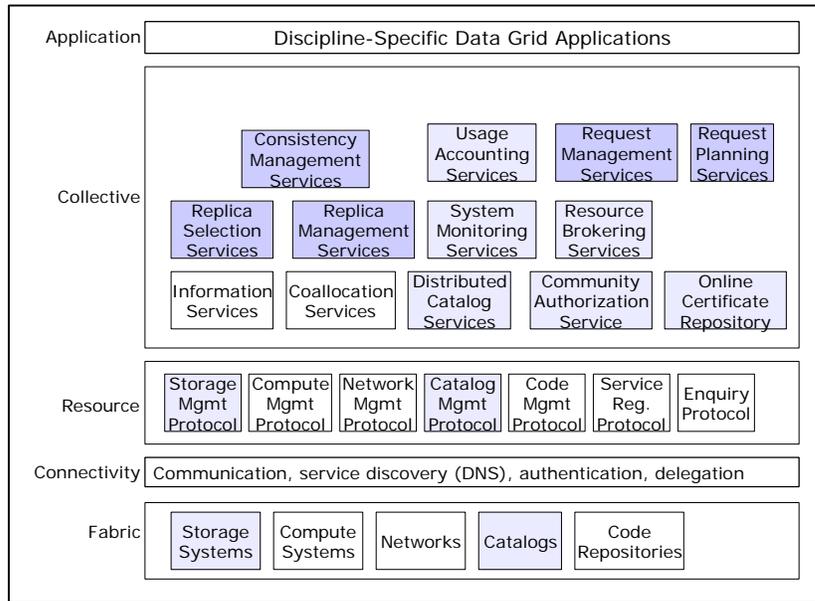
A Data Grid is more than a network: it layers sophisticated *new services* on top of local policies, mechanisms, and interfaces, so that geographically remote resources (hardware, software and data) can be shared in a coordinated fashion. A Data Grid provides a new degree of *transparency* in how data-handling and processing capabilities are integrated to deliver data products to end-user applications, so that requests for such products are easily mapped into computation and/or data retrieval at multiple locations. (This transparency is needed to enable sharing and optimization across diverse, distributed resources, and to keep application development manageable.)...

We need *data catalogs* to maintain information about data itself (metadata), the transformations required to generate derived data (if employed), and the physical location of that data. We have the various *resources* that we must deal with: storage systems, computers, networks, and code repositories. We

require the application-specific *request formulation tools* that enable the end user to define data requests, translating from domain-specific forms to standard request formats, perhaps consulting application-specific ontologies. We require code that implements the logic required to transform user requests for virtual data elements into the appropriate catalog, data access, and computational operations and to control their execution: what we term ... “*request planning*” and “*request execution.*”

Figure 2 in that document (shown here, where the shading indicates some of the elements that must be developed specifically to support Data Grids) identifies components of particular relevance to data grids.

The PPDG Computer Science work areas are designed to match existing and planned components being developed by the contributing CS groups, thus ensuring PPDG deliverables will merge well into the other developments of the field. The experiment applications will provide production quality demonstrations of Grid technologies as they continue to emerge and help validate and solidify the architecture and software systems.



### 1.3.2 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 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, GriPPS<sup>34</sup>, 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.

### **1.3.3 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.

#### **1.3.3.1 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<sup>17</sup> 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.

#### **1.3.3.2 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

#### **1.3.3.3 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 GIIS 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.

#### **1.3.3.4 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<sup>25</sup>, 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.

#### **1.3.3.5 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.

#### **1.3.3.6 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,

### **1.3.3.7 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.

### **1.3.4 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.

#### **1.3.4.1 The ATLAS Experiment**

##### **Distributed computing in ATLAS**

While LHC data taking will not begin until 2006, ATLAS<sup>9</sup> 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.

### 1.3.4.2 The BaBar Experiment

#### Status

In late 2000, the BaBar<sup>1</sup> 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)

#### 1.3.4.3 The CMS Experiment

##### Status and Milestones:

CMS<sup>10</sup> 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% Complexity<sup>†</sup> Data Challenge
- Dec 2002 Software and Computing TDR
- Dec 2003 Physics TDR: Support interactive analysis
- Dec 2004 20% CPU<sup>‡</sup>; ~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

---

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

<sup>‡</sup> CPU power relative to the initial production system at CERN in 2007.

- (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 forms<sup>§</sup>
- (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.
- (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

### 1.3.4.4 D0 Experiment

#### Status and Milestones:

The D0 experiment<sup>11</sup> has developed a fully distributed data access system, SAM<sup>35</sup>, 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 cataloguing and querying, 5. dataset definition and processing history.

---

<sup>§</sup> User interfaces are assumed to be browser-based, including a GUI, command-line and scripting capabilities.

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.

#### **1.3.4.5 The STAR Experiment**

##### **Status**

The STAR<sup>12</sup> 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.

### **1.3.4.6 Thomas Jefferson National Accelerator Facility Experiments**

#### **Data Grid Status and Milestones**

Jefferson Lab<sup>3</sup> 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 Portal<sup>36</sup> 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:

- 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 (silo, 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)

### 1.3.5 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.

#### 1.3.5.1 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](http://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.

### 1.3.5.2 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.

<b>Project Activity</b>	<b>Experiments</b>	<b>Yr1</b>	<b>Yr2</b>	<b>Yr3</b>
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

### 1.3.5.3 Evaluation and Continuing Support

During the last phase of PPDG we will:

- Transition all technologies that remain in service to a sustaining support mode. Support agreements will be explored with commercial vendors as well as with the collaborating CS groups, proposed GriPPs team, and the experiments.
- Document the pros and cons of the approaches used by this project and propose follow-on projects after it's completion.
- Attempt to further generalize the common approach to multiple physics applications into recommendations for common approaches to multiple science areas.
- Explore with software vendors possible commercialization of appropriate components of PPDG software.

### 1.3.6 Computer Science High Speed Networking Testbed

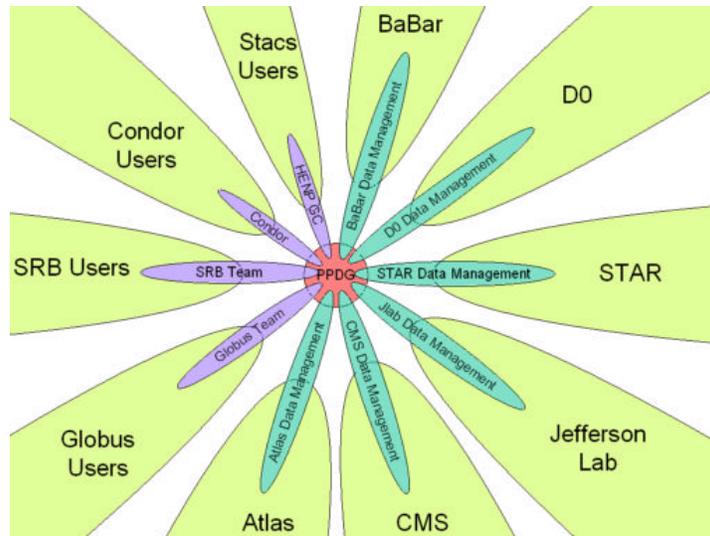
The CS Teams propose to purchase and share a small but powerful dedicated development framework on which to perform the measurements, tests, and experiments needed to achieve the network file transport speeds that the participating experiments require, over the wide area links that they will be using.

This work requires the ability to exercise total control over, and make very low-level changes in, the operating system and network interfaces of a stable performance test platform, and to perform tests under very controlled conditions where the effects of changes and improvements can be accurately and unambiguously measured. All of this experimentation must take place over real wide-area links, as the problems these links pose to transport protocols are not observable on high-speed local area networks.

To perform these tests, we propose to set up two small clusters of four nodes each, at two distant points connected by ultra-high-speed wide area networks. These points may be in laboratories that are connected by existing high speed nets such as at ANL and UW; they can later be moved to points on newer, faster networks such as SurgeNET that are proposed for funding under the SciDAC program.

### 1.3.7 PPDG Management Plan

U.S Physics Experiment computing management and Computer Science project management have joined the PPDG proposal with a common vision to work together to develop and apply Grid software to the benefit of each experiment, and with a commitment to adapt, extend and reuse information technology and software used in one experiment for the benefit of other experiments. The PPDG management plan is designed to provide sufficient coordination of the specific project deliverables and milestones, as well as show our commitment to a process of working together towards the common goal.



The PPDG management challenge is described graphically in the figure, where the circles and ellipses show the relationship between, and approximate relative sizes of, the proposed PPDG-funded team, the teams in the physics experiments concerned with Grid applications and data management, and the Physics Collaborations and CS projects. The obvious challenge, optimal coordination of all Grid-related efforts in spite of the time pressures and physics-focused mission of the experiments, is also an outstanding opportunity. The collective resources that the experiments and CS projects can bring to bear on Grid enabled developments of common interest provide a many-fold leverage of the proposed PPDG funding. Working together closely on these developments will provide both communities with the best environment to meet their scientific goals.

The PPDG management strategy has top-down and bottoms-up components. The bottoms-up strategy involves exploiting the mission-oriented drive of the physics experiments, part of which is already directed towards strengthening collaborations with PPDG-CS and with U.S. and European Grid projects outside of PPDG\*\*. The involvement of the CS projects is driven by their desire to test and evaluate their technology in a real-life setting.

The top-down strategy must ensure the steering of the powerful bottoms-up forces to ensure that, while the individual goals of each of the experiments and the CS projects are met, most developments are of value to other PPDG collaborators, and with little or no additional generalization, to a wide user community. In addition, PPDG must join in the effort to steer collaborations of individual PPDG members with other Grid projects such that a common Grid architecture emerges and all funding is used effectively.

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:

---

\*\* Existing and planned collaborative activities involving PPDG, GriPhyN, and European Grid projects are described in the letters of support attached to this proposal.

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.

### 1.3.7.1 Coordination with other Data Grid projects

PPDG is working with the other major Data Grid projects in the U.S., Europe, and Japan to ensure effective coordination and communication among all parties. Coordination and consultation have already taken place pair wise between projects, with invitations to individuals from one project to join meetings of technical committees of another project or in other way to provide advice and consultation. In some cases these meetings resulted in exchange of Grid technologies and joint activities. A first meeting of all projects to organize more general coordination among the projects was held on March 4 in Amsterdam in connection with the first Global Grid Forum meeting. Although more discussion is needed, the group agreed to work toward a common architecture for all projects to permit tools to work well together. This point is important since the LHC experiments at least are connected to all of the projects and will not benefit from them if the tools are incompatible. Mechanisms were also discussed to avoid, where possible, duplication of effort in developing redundant tools and to permit the total resources of the projects to be used as efficiently as possible. Another meeting is planned on the time scale of June, 2001.

### 1.3.8 Outreach and Education

PPDG plans to involve students in our activities, especially in the testing and deployment phases of the project. We will leverage existing summer student programs that include outreach to minority communities (e.g. at ANL, the U of Wisconsin, SLAC and Fermilab<sup>37</sup>) to employ and involve students in the application activities. The PPDG focus on end-to-end solutions provides a good base for summer students to learn and contribute in practical terms to the use of the Grid in the context of a physic experiment. We will coordinate with existing ATLAS<sup>38</sup> and CMS<sup>39</sup> construction project outreach to include PPDG activities and work in their programs. Through the Quarknet program physicists mentor high school teachers. We will participate in this program to increase teacher and student awareness and understanding of distributed computing and Grid middleware.

### 1.3.9 Budget summary and justification

Funding is requested under Office of Science Notice LAB 01-06, under Office of Science Notice 01-06 and under Office of Science Notices 01-11 and LAB 01-11 (HENP SciDAC). The requested total funding for FY2001, FY2002 and FY2003 is \$3.5M, \$4.0M and \$4.5M. The ramp up reflects the expectation that the rapidly rising needs for data-intensive collaboratory tools in particle and nuclear physics and advances in Grid technology will drive an increasingly productive interdisciplinary collaboration.

The distribution of funding across the teams, management and equipment, in FY2001, is:

<b>CS teams</b>	<b>\$K</b>
Wisconsin	500
ANL	350
LBNL	230

SDSC	70
<b>Physics experiment teams</b>	
ATLAS	400
BaBar	400
CMS	400
D0	400
Jefferson Lab	200
STAR	200
<b>Project management</b>	
Pordes, FNAL	100
Livny, Wisconsin	50
Olson, LBNL	100
<b>Test equipment</b>	
ANL	50
Wisconsin	50
<b>Total \$K FY2001</b>	<b>3500</b>

The distribution of funds across institutions for each team is determined by each team individually and is reflected in the budget pages for each institution. The budget pages for years FY2002 and FY2003 reflect the increased overall funding request of \$4M and \$4.5M using the same distribution across teams as in FY2001. However, as the project progresses the distribution of funds will be evaluated and may be modified for the optimum benefit to the overall goals of the Collaboratory Pilot.

From the description of the proposed effort given above it is clear that the goals are not only ambitious but that all of the work planned is highly leveraged to take advantage of other efforts. The CS activities build on, adapt and deploy middleware that either already exists in some form or is primarily developed under other projects (Globus, Condor, SRB, STACS, GriPhyN). The effort in the physics experiments is directly on activities that must be carried out for the experimental programs and it is just the additional effort to develop/deploy and integrate common Grid enabled solutions across different experiments that is funded under PPDG, and even this covers only part of the effort the experiments are putting into the PPDG work.

It should be clearly understood that the primary purpose of the PPDG Collaboratory Pilot is to adapt, develop and deploy end-to-end Data Grid solutions that are generalized and common across each of the experiments. Based upon the centuries of experience brought to bear by the PPDG participants it is understood that the only way to accomplish this task of common solutions on the necessary short time scale is with such a highly leveraged and carefully managed effort with broad and dedicated participation by many experiments that need, and several of the key CS groups involved with, Data Grid development.

#### **1.4 Subcontract or Consortium Arrangements**

All funded participation in PPDG is via direct funding to each institution and there are no subcontracts or funded consortium arrangements.

## 2 Literature Cited

---

- <sup>1</sup> <http://www.slac.stanford.edu/BFROOT/>
- <sup>2</sup> [www.rhic.bnl.gov](http://www.rhic.bnl.gov)
- <sup>3</sup> [www.jlab.org](http://www.jlab.org)
- <sup>4</sup> <http://lhc.web.cern.ch/lhc/>
- <sup>5</sup> Foster, I. and Kesselman, C. (eds.). *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, 1999.
- <sup>6</sup> Foster, I., Kesselman, C. and Tuecke, S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *Intl. J. Supercomputer Applications*, (to appear) 2001. <http://www.globus.org/research/papers/anatomy.pdf>.
- <sup>7</sup> [www.griphyn.org](http://www.griphyn.org)
- <sup>8</sup> [www.eu-datagrid.org](http://www.eu-datagrid.org)
- <sup>9</sup> <http://atlasinfo.cern.ch/Atlas/Welcome.html>
- <sup>10</sup> <http://cmsinfo.cern.ch/Welcome.html/>
- <sup>11</sup> <http://www-d0.fnal.gov/>
- <sup>12</sup> [www.star.bnl.gov](http://www.star.bnl.gov)
- <sup>13</sup> [http://www.ppdg.net/docs/ppdg\\_qtrly\\_dec00.pdf](http://www.ppdg.net/docs/ppdg_qtrly_dec00.pdf)  
[http://www.ppdg.net/docs/ppdg\\_qtrly\\_sep00.pdf](http://www.ppdg.net/docs/ppdg_qtrly_sep00.pdf)  
[http://www.ppdg.net/docs/PPDG\\_HENP\\_april00\\_public.pdf](http://www.ppdg.net/docs/PPDG_HENP_april00_public.pdf) - Appendix A  
<http://gizmo.lbl.gov/ppdg/Documents/110399.html>  
<http://gizmo.lbl.gov/ppdg/Documents/082599.html>
- <sup>14</sup> GriPhyN Data Grid Reference Architecture, E. Deelman, I. Foster, C. Kesselman, M. Livny, [http://www.phys.ufl.edu/~avery/griphyn/meeting\\_20dec00/foster\\_architecture.pdf](http://www.phys.ufl.edu/~avery/griphyn/meeting_20dec00/foster_architecture.pdf)
- <sup>15</sup> [www.globus.org](http://www.globus.org)
- <sup>16</sup> Foster, I., Kesselman, C., Tsudik, G. and Tuecke, S. A Security Architecture for Computational Grids. In *ACM Conference on Computers and Security*, 1998, 83-91.
- <sup>17</sup> Livny, M. High-Throughput Resource Management. In Foster, I. and Kesselman, C. eds. *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, 1999, 311-337.
- <sup>18</sup> Moore, R., Baru, C., Marciano, R., Rajasekar, A. and Wan, M. Data-Intensive Computing. In Foster, I. and Kesselman, C. eds. *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, 1999, 105-129.
- <sup>19</sup> Johnston, W.E., Gannon, D. and Nitzberg, B., Grids as Production Computing Environments: The Engineering Aspects of NASA's Information Power Grid. In *Proc. 8th IEEE Symposium on High Performance Distributed Computing*, 1999, IEEE Press.
- <sup>20</sup> Stevens, R., Woodward, P., DeFanti, T. and Catlett, C. From the I-WAY to the National Technology Grid. *Communications of the ACM*, 40(11):50-61. 1997.
- <sup>21</sup> Beiriger, J., Johnson, W., Bivens, H., Humphreys, S. and Rhea, R., Constructing the ASCI Grid. In *Proc. 9th IEEE Symposium on High Performance Distributed Computing*, 2000, IEEE Press.
- <sup>22</sup> <http://www.cs.wisc.edu/condor/>
- <sup>23</sup> Rajesh Raman, Miron Livny, and Marvin Solomon, "Resource Management through Multilateral Matchmaking", Proceedings of the Ninth IEEE Symposium on High Performance Distributed Computing (HPDC9), Pittsburgh, Pennsylvania, August 2000, pp 290-291. <http://www.cs.wisc.edu/condor/doc/gangmatching.ps>

- 
- <sup>24</sup> <http://gizmo.lbl.gov/ppdg/>
- <sup>25</sup> <http://www.npaci.edu/DICE/SRB>
- <sup>26</sup> <http://gizmo.lbl.gov/stacs/>
- <sup>27</sup> <http://www-rnc.lbl.gov/GC/>
- <sup>28</sup> <http://grid.fnal.gov/ppdg/hrm.html>
- <sup>29</sup> Allcock, B., Bester, J., Bresnahan, J., Chervenak, A.L., Foster, I., Kesselman, C., Meder, S., Nefedova, V., Quesnel, D. and Tuecke, S., Secure, Efficient Data Transport and Replica Management for High-Performance Data-Intensive Computing. In *Mass Storage Conference*, 2001.
- <sup>30</sup> Foster, I., Kesselman, C., Tsudik, G. and Tuecke, S. A Security Architecture for Computational Grids. In *ACM Conference on Computers and Security*, 1998, 83-91.
- <sup>31</sup> Butler, R., Engert, D., Foster, I., Kesselman, C., Tuecke, S., Volmer, J. and Welch, V. Design and Deployment of a National-Scale Authentication Infrastructure. *IEEE Computer*, 33(12):60-66. 2000.
- <sup>32</sup> [www.gridforum.org](http://www.gridforum.org)
- <sup>33</sup> <http://www.ppdg.net/archives/ppdg/2001/msg00069.html>
- <sup>34</sup> Grids in Particle Physics Support Team (GriPPS) <http://home.fnal.gov/~ggraham/GriPPS/index.html>
- <sup>35</sup> <http://d0db.fnal.gov/sam/>
- <sup>36</sup> <http://lqcd.jlab.org/>
- <sup>37</sup> <http://sist.fnal.gov/>
- <sup>38</sup> <http://pdg.lbl.gov/atlas/outreach.html>
- <sup>39</sup> <http://cmsdoc.cern.ch/cms/outreach/html/>  
<http://quarknet.fnal.gov/>