Technical Report GriPhyN-2001-xxx

www.griphyn.org

GriPhyN Research for ATLAS

Year 2 Project Plan

ATLAS Application Group

GriPhyN Collaboration

&

Software and Computing Project

U.S. ATLAS Collaboration

12/21/2001

Version 3.7

Table of Contents

1 Introduction

The goal of this document is to provide a detailed GriPhyN – ATLAS plan for Year 2. Some high-level plans for Year 3 are also included.

As an application, ATLAS software is inherently fit for Grid computing with its distributed data and computing needs. As such, we have developed a coordinated approach for the various U.S. Grid projects, namely PPDG, GriPhyN and iVDGL, focused on delivering the necessary tools for the ATLAS Data Challenges (DCs).

These tools fall into several different categories. First, and mainly as part of PPDG although it will be used for GriPhyN, is Magda – the Manager for Grid-based Data. This tool is a data distribution tool/sandbox that is being used for initial work in distributed data management. It was developed to enable the rapid development of components to support users, and as other project pieces reach maturity they are easily incorporated. For example, GridFTP was recently incorporated, and current testing using the Globus replica management tools is underway, replacing original prototype code for both functions. This is detailed in Section 3. Related to this work is Adagio (Athena Data Access using Grid I/O) to enable data-access over the Grid from within the Athena infrastructure. This is described in Section 4.

Second is Grappa, the Grid Portal for Physics Applications, which is being developed as a user-friendlier interface to job submission and monitoring. It will interface to Magda in future development as well. This work is based on the Indiana XCAT Science Portals project, and is fully adaptable to new developments in Grid software and in the ATLAS/Athena framework itself. Initial work has focused on a simpler, web-based interface to job submission, with easy access to Python job scripting. This is fully described in Section 5.

In the area of monitoring, there are several on-going efforts with respect to different aspects of the problem. We are leading the joint PPDG/GriPhyN effort in monitoring to define the use cases and requirements for a cross-experiment testbed. In addition, we have been evaluating and installing sensors to capture the needed data for our testbed facilities internally, and determining what information should be shared at the Grid level, and the best ways to do this. At the application level, much work has been done with Athena Auditor services to evaluate application performance on the fly. For visualizing monitoring data, we have developed GridView to easily see the resource availability on the U.S. ATLAS Testbed. These efforts are detailed in Section 6.

We have also been working on the development of many needed management tools. Section 7 discusses Pacman, our package management system, which is a candidate for the packaging of the VDT. Section 8 addresses our security approaches, or rather, our use of other people's work in security, and Section 9 describes our approach to site management software. Section 10 describes testbed development. Section 11 describes education and outreach activities. Section 12 summarizes GriPhyN – ATLAS goals, and Section 13 provides information about project management.

1.1 High Level Goals

The high level goals for Year 2 of GriPhyN – ATLAS are

1. Provide support for analysis of ATLAS Data Challenge 1 data collections. The reasons for this approach are:
   - To drive development of GriPhyN technologies in several key areas including Grid data access and management, grid user interfaces, monitoring and security.
   - To demonstrate added value by GriPhyN to immediate ATLAS project objectives.
   - To acquire support within ATLAS for adoption of GriPhyN technologies, important for future VDT releases which will emphasize virtual data methods.
   - Coordinate GriPhyN virtual data research with other U.S. Grid projects including PPDG and iVDGL.
   - To forge GriPhyN ties with the ATLAS Data Challenge team in which participants from several Grid development teams and testbed efforts centered at CERN and the EU work to support ATLAS computing objectives.

2. Demonstrate large scale instantiation of compute resources comprising the hierarchy of LHC Computing Model Tiers 0-3 through design, development and testing of site management and software packaging tools.

3. Create ATLAS demonstrations for SC and other venues (such as CHEP) which exhibit cradle-to-grave Grid-level analysis of ATLAS high energy physics data, operative from the point of view of a physicist-user at the Tier 3 level.

4. Explore and/or design ATLAS instances of virtual data tracking and catalog architectures being developed by the GriPhyN – CMS and LIGO application teams leading to specification and development of virtual data toolkit components for ATLAS in Years 3-5.

Technical goals for Year 2 of GriPhyN – ATLAS are:

1. Provide easy access mechanisms to DC1 data using Magda, enhancements to Magda which capture metadata attributes created during DC1 production, and other collection navigational tools.
2. Support Grid file replication and data distribution efforts (GridFTP) for distribution of DC1 data from production sites (CERN and a few Tier 1 sites) and the Tier 1 at Brookhaven, and the two ATLAS prototype Tier 2 sites (part of iVDGL) at Boston University and Indiana University.
3. Register DC1 data caches at these sites with Magda.
4. Continue development of Pacman source distribution caches for VDT, ATLAS, and other external software packages as required.
5. Continue development of ATLAS remote site execution environment and startup kits.
6. Develop simple Grid job submission tools based on the Grappa portal.
7. Deploy a grid information service for ATLAS / iVDGL sites based on the MDS2 service.
8. Deploy an ATLAS software information service which describes ATLAS software

installations at Grid sites based on MDS.

9. Connect Grappa with grid information service describing Grid resources available to ATLAS users.
10. Demonstration series of various pieces of the ATLAS production and analysis chain for Monte Carlo data
11. Demonstration of Grid-based data analysis using ATLAS software at a significant number of Grid sites, beginning first with Tiers 0-2, later expanding to ATLAS Tier 3 sites, and later to non-ATLAS sites such as sites within iVDGL home to the other GriPhyN application teams.
12. Demonstrate connectivity of Grid-based data analysis jobs based on GriPhyN technology to DataGrid Testbed sites.

1.2 Terminology and Acronyms

Several acronyms are used within this document. They include the following project related acronyms:

DC	Data Challenge, defined by the ATLAS project
GG	GriPhyN – ATLAS goals as defined in this document
GM	GriPhyN – ATALS milestone as defined in this document
iVDGL	International Virtual Data Grid Laboratory Project
VDT	Virtual Data Toolkit, developed by GriPhyN and supported by iVDGL
PG	PPDG – ATLAS project goals, as defined by PPDG project plans
PPDG	Particle Physics Data Grid Collaboratory Project
EU DG	European DataGrid project

In addition, in sections below we identify work items, approximate schedules, and significant milestones to mark progress. Where appropriate, we cross reference this to the Grid planning schedule for U.S. ATLAS which include activities from other Grid projects such as PPDG, iVDGL, liaison and integration tasks associated with the EU DataGrid testbed effort, the HENP Networking Working Group, etc. Below is an example, with work area key following.

Table 1 Example work item list and milestones

GriPhyN Code	ATLAS Grid WBS	Name	Description	Date Start	Date End
Type-X1	1.1.x	Short name for project from work area X	More detail	Year-Quarter	Year-Quarter
Type-X2	1.2.x				
Milestones					
Type-X1	1.1.x	Short name for project milestone from work area X	More detail	Date	

Table 2 Keys denoting work areas within GriPhyN - ATLAS

Type		Project area X	
GG	GriPhyN Goal	D	Grid Data Access from Athena
GM	GriPhyN Milestone	DM	Data management, Magda
CP	Challenge Problem	P	Packaging
GD	GriPhyN Demonstration	I	Interface, Grappa
		T	Testbed
		M	Monitoring
		0	Education and Outreach

2 ATLAS[1]

This section gives a basic overview to the ATLAS software environment, describes the data challenges which drive the computing goals for ATLAS, and lists the involved personnel.

2.1 ATLAS Software Overview

Athena[2] is the common object oriented framework used by the ATLAS experiment for simulation, reconstruction, event filtering, and analysis. It is based on the GAUDI[3] architecture developed by the CERN LHCb collaboration. Development of the GAUDI kernel has since become a joint, multi-experiment project as other HEP experiments have since adopted the framework. ATLAS software is still in a migratory phase from previous Fortran-based procedural codes, such as ATLSIM, ATRECON, etc, and the Fortran based HEP event simulation package GEANT-3, to the new OO framework. The state of affairs is the result of tactical decisions, made at the international ATLAS level, to use well understood, benchmarked codes for the extensive physics and detector performance studies which formed the basis of the *Detector and Physics Performance Technical Design Report*[4]. The decision resulted in the successful validation of the ATLAS spectrometer design, but a necessary consequence of the approach was to delay the transition to the new OO-based framework. As such, the core ATLAS software is today in a highly developmental phase, and so in some cases Fortran legacy codes are used for preliminary Grid toolkit evaluations.

The Athena architecture, indicated by the object diagram of Figure 2-1, supports multiple data persistency services and insulates user code from the underlying storage technology. Physicists supply algorithms which perform tasks such as track finding and fitting, vertex finding, cluster finding and reconstruction. Users interact with the Athena through use of job options files, and in the near future, by a Python scripting interface. Adagio, discussed in Section 4 below, is an effort within PPDG and the core ATLAS database groups to examine the connectivity layer between Grid and core ATLAS persistency services. The run time environment is complex, as user algorithms are dynamically linked with shared object libraries, resident on the local machine or accessible from a remote site via AFS. Other files, such as parameter files and conditions databases, need to be setup and configured properly.

A major goal of ATLAS-GriPhyN is to create the Grid interfaces for collection of Athena services (such as EventSelector, histogram, auditors, messaging and monitoring services), and to provide research tools which can be more broadly useful within the GriPhyN collaboration. For example, Pacman is a software management tool to aid in the deployment of software packages, helping maintain consistency among software distributed across multiple administrative domains on the Grid. Magda is a data management tool used for viewing, accessing, and adding to Grid-distributed data with interfaces to C, Java, Perl, and the Web. Grappa is a high-level user interface based on science portal technology, allowing physicists to launch jobs, monitor them, and interact with Grid data tools without having to learn the details of Grid programming. Such a high-level interface is more than a GUI for ATLAS: GriPhyN research entails multiple approaches for problems such as metadata management, and Grappa is designed with a software component plug-and-play architecture that allows using any or all of those different approaches.

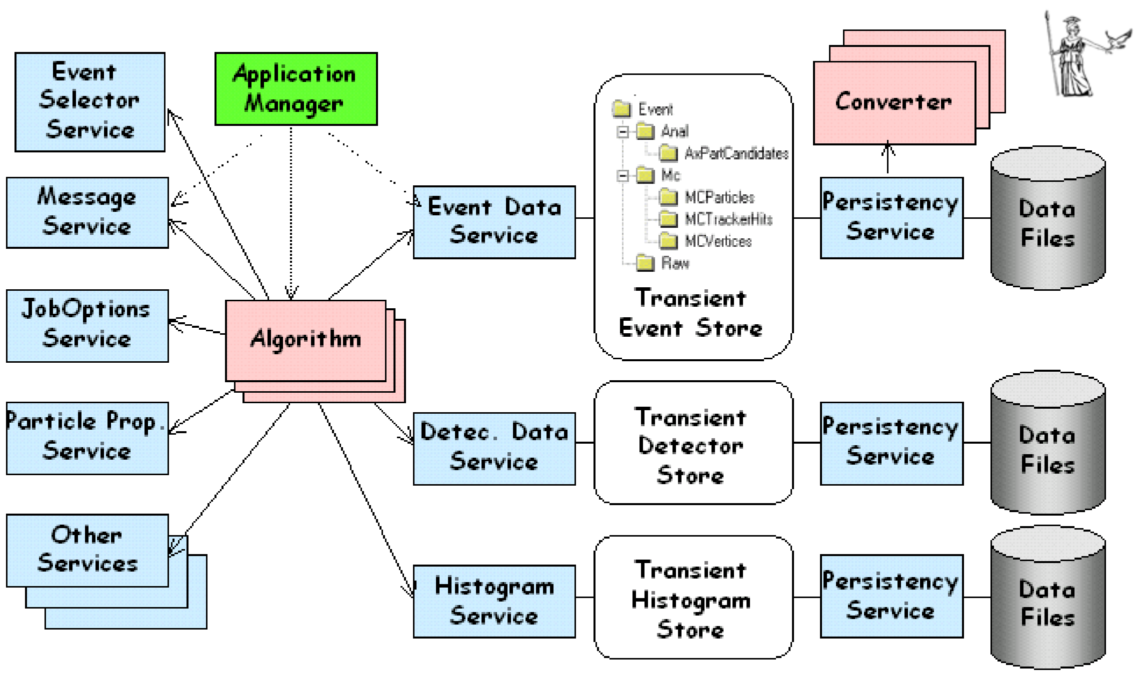


Figure 2-1 Athena Object Diagram

2.2 ATLAS Data Challenges

The ATLAS collaboration will undertake a series of data challenges* in order to validate the LHC Computing Model, which underwent an extensive CERN review concluding in January 2001. The validation will address all aspects of ATLAS software, especially its control framework and data model, and eventually choices in Grid technology. The data challenges will be executed at prototype Tier computing centers, and will be of increasing complexity and scale to exercise as much as possible Grid middleware technologies. The results of the Data

* See http://atlasinfo.cern.ch/Atlas/GROUPS/SOFTWARE/DC/dc_page.html for more information about the DC effort within ATLAS.

Challenges will be used as input for a Computing Technical Design Report due by the end of 2003 for ATLAS.

The first three data challenges (DC) that will be run starting in December of 2001, and will last until December of 2003. The DCs are designed to have physics content in order to draw a large group of data analysts from the physics community, thus providing a better check and exercise of the software. Data Challenge 0 (DC0), which runs in December 2001 and January 2002, is essentially a test of the continuity of the code chain. DC0 will provide ATLAS with continuity tests of several key data paths:

Generators --> full simulation --> reconstruction --> analysis;
Generators --> fast simulation --> analysis;
Physics TDR[†] data --> reconstruction --> analysis.

A by-product of these continuity tests will be a reference set of scripts and related job options files used to validate each link in the three chains. Each of these recipes essentially defines a transformation of one data product into another, and each of these "standard" recipes will be of interest to the ATLAS collaboration at large. These standard job options and scripts will provide a foundation for our prototyping of transparency with respect to materialization, and serve as a basis for our initial transformation catalog. Only modest DC0 samples will be generated, and essentially all in flat “traditional” sequential file format. DC0 production will likely take place at CERN only, though Tier 1 sites will be testing their software environments with DC0 executions in preparation for DC1.

Data Challenge 1 (DC1) will run during the first half of 2002, and will be divided in two phases. In the first phase several sets of 10^7 events for high-level trigger studies will be generated. The second phase will be oriented to physics analysis, with several types of sets generated, some as large as 10^7 events. The second phase will also focus on testing new software, including Geant4, the new event data model and the evaluation of database technologies such as Root-I/O. The production of DC1 data will involve CERN and also sites outside of CERN, such as the Brookhaven Tier 1 facility. Several hundred PCs world-wide will participate.

Data Challenge 2 (DC2) runs for the first half of 2003. The scope will depend on what was accomplished in DC1, but the main goal will be to have the full new software in place. We will generate several samples of 10^8 events, mainly in OO-databases, and with large-scale physics analysis using Grid tools.

All the Data Challenges will be run on Linux systems operating according to ATLAS specifications, and with the compilers distributed with the code if not already installed locally in the correct versions. The DCs are summarized in Table 3 below. Subsequent to the planned DC0-DC2 challenges will be Full Chain and 20% scale processor farm tests. The detailed plans for these challenges will depend on the results of the first three DCs.

Our approach within GriPhyN is to design goals and milestones in coordination with, and in support of, the major software and computing activities of the international ATLAS Collaboration. Hence the attention paid to these DCs. This work is being done in close conjunction with specific Grid planning[5] underway within the U.S. ATLAS Software and

[†] TDR = technical design report for detector and physics performance, as previously mentioned

Computing Project, which includes planning for the Particle Physics Data Grid Project (PPDG) and iVDGL.

Table 3 Schedule and Specifications for ATLAS Data Challenges

Name	Date	Events #, size	CPU SI95-sec	Data Volume	Description
DC0	December 01 to February 02	10^5 2.5 MB	10^8	1 TB	Continuity check of ATLAS software
DC1	February 01 to July 02	10^7 2.5 MB (larger if higher luminosity or if hits and digits written out)	Simulation: 3×10^{10} Recon: 6×10^9	Simulation: 20 TB Reconstruction: 5 TB (Multiples of this if pileup is assumed and hits are written out.)	Major test of production capabilities; 1% scale relative to final system. Grid tools to be used in analysis phase.
DC2	January 03 to September 03	10^8	10^{12}	100 TB, but perhaps as much as 50% of the full scale	10% scale test. Large scale production deployment of multi-tiered distributed computing services.
Full Chain Test	July 04	10^8	10^{12}	TBD	Test of full processing bandwidth, from high level trigger through analysis. High throughput testing of distributed services.
20% Processing Farm Prototype	December 04	10^9	10^{13}	Up to 0.5 PB	Production processing test with 100% complexity (processor count), 20% capacity system relative to 2007 level. High throughput, high complexity testing of distributed services.

2.3 Personnel

The ATLAS – GriPhyN team, Table 4 , involves participation from a number of individuals from ATLAS affiliated institutions and from computer scientists from GriPhyN university and laboratory groups. In addition, there is significant joint participation with PPDG funded efforts at ANL and BNL.

Table 4 ATLAS – GriPhyN Application Group

Name	Institution	Affiliations	Role	Work Area
Rich Baker	BNL	PPDG, ATLAS	Physicist	Testbed, monitoring
Randall Bramley	IU	GriPhyN	Computer Scientist	Grappa
Kaushik De	UTA	ATLAS	Physicist	GridView, Testbed
Daniel Engh (start 2/02)	IU	GriPhyN, ATLAS	Physicist	Athena – Grappa, grid data access
Lisa Ensman (till 4/02)	IU	GriPhyN, ATLAS	Physicist	Athena – Grappa
Dennis Gannon	IU	GriPhyN	Computer Scientist	Grappa
Rob Gardner	IU	GriPhyN, ATLAS	Physicist	Project lead, physics contact
John Huth	HU	GriPhyN, ATLAS	Physicist	Management
Fred Luehring	IU	ATLAS	Physicist	ATLAS applications
David Malon	ANL	PPDG, ATLAS	Computer Scientist	Athena Data Access
Ed May	ANL	PPDG, ATLAS	Physicist	Testbed coordination
Jennifer Schopf	ANL	GriPhyN, Globus, PPDG	Computer Scientist	CS contact, Monitoring
Jim Shank	BU	GriPhyN, ATLAS	Physicist	ATLAS applications
Shava Smallen	IU	GriPhyN	Computer Scientist	Grappa
Jason Smith	BNL	ACF, ATLAS	Physicist	Monitoring, Testbed
Valerie Taylor	NU	GriPhyN	Computer Scientist	Athena Monitoring
Alex Undrus	BNL	ATLAS	Physicist	Software Librarian
Torre Wenaus	BNL	PPDG, ATLAS	Physicist	Magda
Saul Youssef	BU	GriPhyN, ATLAS	Physicist	Pacman, ATLAS app.
Dantong Yu	BU	PPDG, ATLAS	Computer Scientist	Monitoring

3 Manager of Grid-based Data – Magda

Magda (MAnager for Grid-based Data) is a distributed data manager prototype for Grid-resident data. Magda is being developed by the Particle Physics Data Grid as an ATLAS/Globus project to fulfill the principal ATLAS PPDG deliverable of a production distributed data management system deployed to users and serving BNL, CERN, and many U.S. ATLAS Grid testbed sites (currently ANL, LBNL, Boston University and Indiana University). The architecture is illustrated in Figure 3-1. The objective is a multi-point U.S. Grid (in addition to the CERN link) providing distributed data services to users as early as possible. Magda provides a component-based rapid prototyping development and deployment infrastructure designed to promote quick in-house development of interim components later replaced by robust and scalable Grid Toolkit components as they mature.

These work statements refer to components of U.S. ATLAS Grid WBS 1.3.3.3 (Wide area distributed replica management and caching) and WBS 1.3.5.5 (Infrastructure metadata management).

The deployed service will be a vertically integrated suite of tools extending from a number of Grid toolkit components (listed below) at the foundation, through a metadata cataloging and distributed data infrastructure that is partly an ATLAS-specific infrastructure layer and partly a generic testbed for exploring distributed data management technologies and approaches, to primarily experiment-specific interfaces to ATLAS users and software.

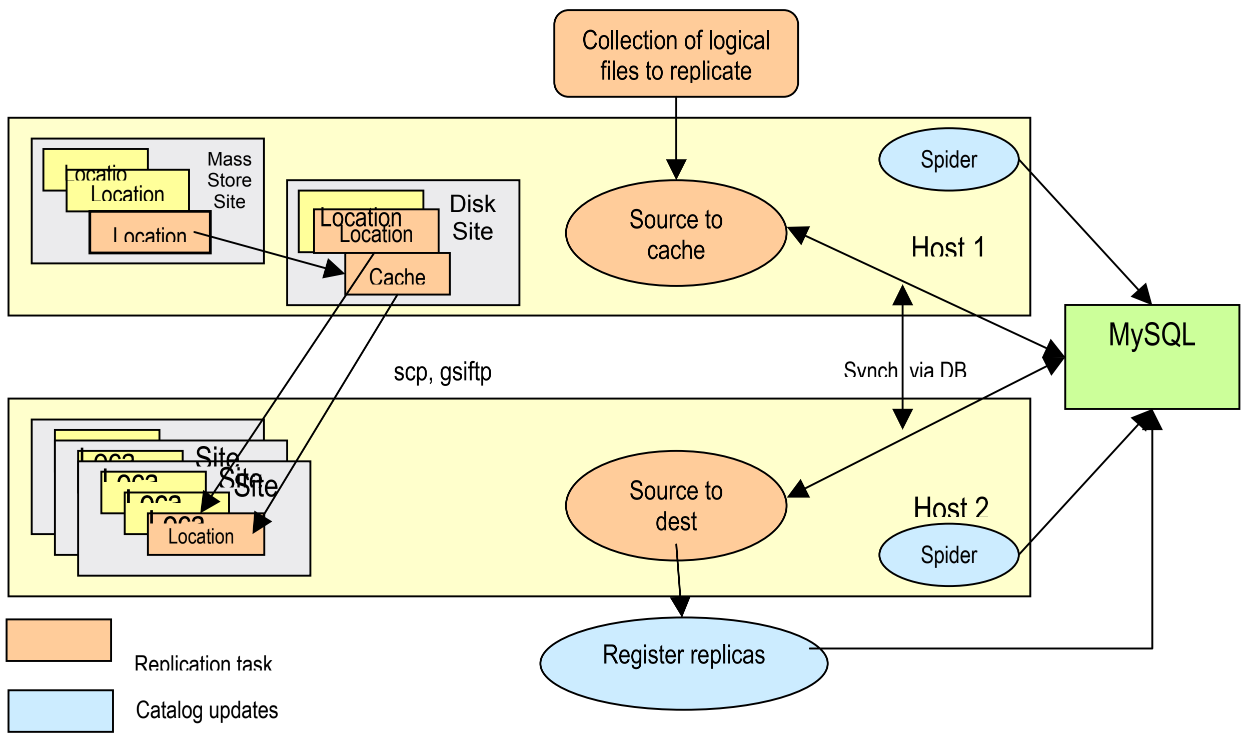


Figure 3-1 Magda Architecture (PPDG)

Grid Toolkit tools in use or being integrated within Magda include Globus GridFTP file transfer, GDMP replication services, Globus replica catalog, Globus remote execution tools, and Globus replica management.

Magda has been in stable operation as a file catalog for CERN and BNL resident ATLAS data since May 2001 and has been in use as an automated file replication tool between CERN and BNL mass stores and U.S. ATLAS Grid testbed sites (ANL, LBNL, Boston, Indiana) since summer 2001. Catalog content fluctuates but is typically a few 100K files representing more than 2TB of data. It has been used without problems with up to 1.5M files. It will be used in the forthcoming ATLAS Data Challenges DC0 (Dec 2001-Feb 2002) and DC1 (mid to late 2002). In DC1 a Magda version integrated with the GDMP publish/subscribe data mirroring package (under development within PPDG and EUDG WP2) will be deployed. The principal PPDG milestone for Magda is fully functional deployment to general users as a production distributed data management tool in June 2002. The principal GriPhyN/iVDGL milestone is Magda-based delivery of DC1 reconstruction and analysis data to general users throughout the U.S. ATLAS Grid testbed within 2 months following the completion of DC1.

In addition to its role in early deployment of a distributed data manager, Magda will also serve as a development tool and testbed for longer term R&D in data signatures (dataset and object histories comprehensive enough to permit on-demand regeneration of data, as required in a virtual data implementation) and object level cataloging and access. This development work will be done in close collaboration with GriPhyN/iVDGL, with a GriPhyN/iVDGL milestone to deliver dataset regeneration capability in September 2003.

In mid 2002 Magda development in PPDG will give way to an emphasis on developing a distributed job management system (the PPDG ATLAS Year 2 principal deliverable) following a similar approach, and building on existing Grid tools (Condor, DAGman, MOP, etc.). This work will be done in close collaboration with GriPhyN/iVDGL development and deployment work in distributed job management and scheduling.

ATLAS GriPhyN/iVDGL developers plan to integrate support for Magda based data access into the Grappa Grid portal now under development (see Section 5).

3.1 Magda references:

Magda main page: http://ATLASsw1.phy.bnl.gov/magda/dyShowMain.pl

Magda information page: http://ATLASsw1.phy.bnl.gov/magda/info

PPDG BNL page: http://www.usATLAS.bnl.gov/computing/ppdg-bnl/

3.2 Magda Schedule

Table 5 Magda work items and milestones as related to GriPhyN

GriPhyN Code	ATLAS Grid WBS	Name	Description	Start	End
GG-DM1	1.3.3	Deployment of basic services	Setup of Magda infrastructure for use in DC1 at GriPhyN / iVDGL sites; use of Pacman	Y2-Q1	Y2-Q4
GG-DM2	1.3.5	Metadata development	Magda metadata management	Y2-Q2	Y2-Q4

GriPhyN Code	ATLAS Grid WBS	Name	Description	Start	End
			interfaces for GriPhyN		
GG-DM3	1.3.3	Job submission interfaces	Interface to Grappa portal and other grid user interfaces	Y2-Q3	Y3-Q1
GG-DM4	1.3.3	Virtual data extensions	Development of Virtual Data signature tools	Y2-Q4	Y3-Q4
Milestones					
GM-DM1	1.3.3	Magda population	Construction of Magda database cataloging DC1 data will occur automatically during DC1 production.	8/1/02	
GM-DM2	1.3.5	Metadata interface	Complete interface to Grenoble metadata catalog with DC1 attributes	4/1/02	
GM-DM3	1.3.3	Job submission interfaces	Functional extension of Magda with Grappa interface	9/1/02	
GM-DM4	1.3.3	Data set regeneration	Data set regeneration using Virtual data tools developed within PPDG and GriPhyN	9/1/03	

4 Grid Enabled Data Access from Athena – Adagio

An important component of the U.S. ATLAS Grid effort is the definition and development of the layer that connects ATLAS core software to Grid middleware. Athena is the common execution framework for ATLAS simulation, reconstruction, and analysis. Athena components handle physics event selection on input, and support event collection creation, data clustering, and event streaming by physics channel on output. The means by which data generated by Athena jobs enter Grid consciousness, the way such data are registered and represented in replica and metadata catalogs, the means by which Athena event selectors query metadata, identify logical files, and trigger their delivery--all of these are the concern of this connective layer of software.

Work to provide Grid-enabled data access from within the ATLAS Athena framework is underway under PPDG auspices. Prototype implementations supporting event collection registration and Grid-enabled Athena event selectors were described at the September 2001 conference on Computing in High Energy and Nuclear Physics in Beijing (cf. Malon, May, Resconi, Shank, Vaniachine, Youssef, "Grid-enabled data access in the ATLAS Athena framework," Proceedings of Computing in High Energy and Nuclear Physics 2001, Beijing, China, September 2001). An important aspect of this work is that the Athena interfaces are supported by implementations both on the U.S. ATLAS Grid testbed (using the Globus replica catalog directly), and on the European Data Grid testbed (using GDMP, a joint EDG/PPDG product).

4.1 Adagio Schedule

Table 6 Adagio work items and milestones related to GriPhyN

GriPhyN Code	ATLAS Grid WBS	Name	Description	Start	End
GG-D1		Athena Grid Registration	Grid registration of Athena products	Y2-Q2	Y2-Q4
GG-D2		Athena Grid Input	Grid-aware Athena input specification	Y2-Q2	Y3-Q2
GG-D3		Athena Runtime Grid Access	Run-time access to grid-managed data	Y2-Q3	Y3-Q3
Milestones					
GM-D1		Athena Grid Registration	Registration of Athena data products in grid replica management services	6/1/02	
GM-D2		Athena Metadata Registration	Registration of Athena data products in metadata services	9/1/02	
GM-D3		Athena Logical-file-based input	Logical-file-based input specification in Athena Job Options	9/1/02	
GM-D4		Athena Grid Event Selection	Grid-enabled Athena event selection services	3/1/03	

5 Grid User Interface – Grappa

Grappa is an acronym for Grid Access Portal for Physics Applications. This work supports U.S. ATLAS Grid WBS 1.3.9 (Distributed Analysis Development) work breakdown deliverables. The preliminary goal of this project was to provide a simple point of access to Grid resources on the U.S. ATLAS Testbed. The project began in May 2001.

5.1 Grid Portals

While there are a number of tools and services being developed for the Grid to help applications achieve greater performance and functionality, it still takes are great deal of effort and expertise to apply these tools and services to applications and execute them in an everyday setting. Furthermore, these tools and services rapidly change as they become more intelligent and more sophisticated. All of this can be especially daunting to Grid application users who are mostly interested in performance and results but not necessarily the details of how it is accomplished. One approach that has been used to reduce the complexity of executing applications over the Grid is a *Grid Portal*, a web portal by which an application can be launched and managed over the Grid[6]. The goal of a Grid Portal is to provide an intuitive and easy-to-use web (or optionally an editable script) interface for users to run applications over the Grid with little awareness about the underlying Grid protocols or services used to support their execution[7].

5.2 Grappa Requirements

5.2.1 Use Cases

In order to understand submission methods and usage patterns of ATLAS software users, information (specifications of environment variables, operating system, memory, disk usage, average run time, control scripts, etc.) will be collected from physicists and used to formulate scenario documents, understandable by physicist and non-physicist alike. Those scenario documents will guide the further design of our Grid Portal for the submission and management of ATLAS physics jobs. One such scenario has been developed for ATLSIM[8], the Geant3-Fortran based full simulation of the ATLAS detector. Others will be developed to provide a complete understanding of how ATLAS users will want to use the Grid.

Our initial analysis has shown that in addition to job launch, the portal must provide the ability to enter and store parameters and user annotations (notes, images, graphs) for re-use, single point authentication, real-time viewing of output and errors, and the ability to interface with mass storage devices and new Grid tools as they become available. Grappa will satisfy most of the requirements by integrating existing technologies and making them accessible via a single user interface. Tools such as the Network Weather Service, Prophesy, NetLogger are examples of existing software that Grappa will use for job management, as well as GriPhyN tools for coherent data management (Grid WBS 1.3.3.5), data distribution (Grid WBS 1.3.3.7), and data access management (Grid WBS 1.3.3.9).

Conceptually, Grappa lets physicists easily submit requests to run high throughput computing jobs on either on simple Grid resources (such as remote machine) or more advanced Grid resources such as a Condor scheduling system. Job submission allows simple parameter entry and automatic variation. Users interact from scripts or Web interfaces, and can specify resources by name or requirements. Application monitoring and job output logs are returned to the script or Web browser. A major goal of Grappa is to allow users to manage ATLAS jobs and data with an interface that does not change, while the underlying Grid tools and resources are developed within GriPhyN.

5.3 Grappa and Existing Tools

5.3.1 XCAT Science Portal

One Grid Portal effort underway at the Extreme! Computing Laboratory at Indiana University is the XCAT Science Portal which provides a script-based approach for building Grid Portals. An initial prototype of this tool has been developed to allow users to build personal Grid Portals and has been demonstrated with several applications. A simplified view of the architecture is illustrated in the Figure below. Following is a brief description.

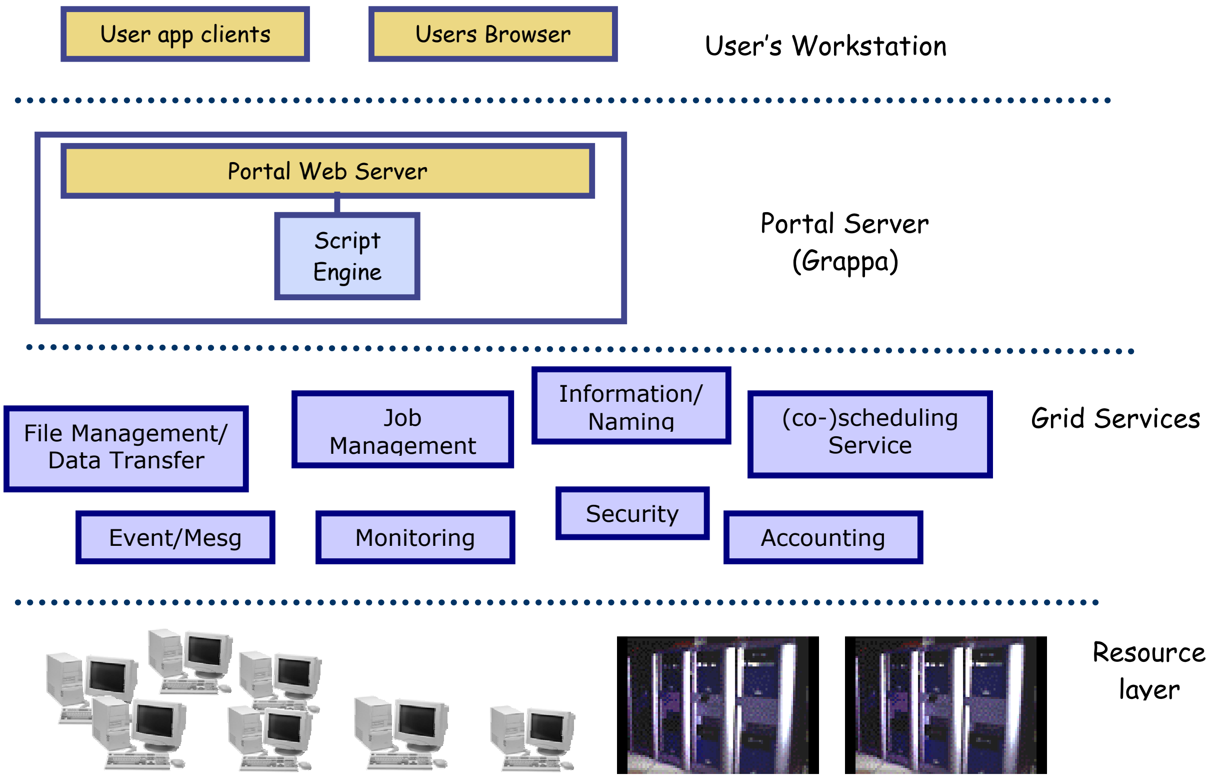


Figure 5-1 XCAT Science Grid Portal Architecture

Currently, a user authenticates to the portal using a GSI credential; a proxy credential is then stored so that the portal can perform actions on behalf of the user (such as authenticating jobs to a remote compute resource). The user can access any number of *active notebooks* within their notebook database. An active notebook encapsulates a session and consists of HTML pages describing the application, forms specifying the job's configuration, and Java Python scripts for controlling and managing the execution of the application. These scripts interface to Globus services in the GriPhyN Virtual Data Toolkit and have interfaces following the Common Component Architecture (CCA) Forum's specifications, which allows them to interact with and be used in high-performance computation and communications frameworks[9]. For a fuller description of the diagram and the XCAT Science Portal, see Ref. [6].

Using the XCAT Science Portal tools, Grappa is currently able to use Globus credentials to perform remote execution, store user's parameters for re-use or later modification, and run ATLSIM and Athena – ATLFAST based on the current scenario documents.

5.4 XCAT Design Changes for Grappa

Grappa will continue to build on top of the XCAT Science Portal technology, with interfaces added to the tools and services being developed by GriPhyN and other data Grid projects. While the requirements of ATLAS applications continues to be assessed (see Section 5.1), the underlying XCAT Science Portal will be redesigned to follow a cleaner three-tier architecture, partitioned as indicated by dotted lines in Figure 5.1.

Grid Services are to be separated from the *Grid Portal* giving greater flexibility to integrate new tools as they become available, tools such as Magda, described in Section 3, and monitoring systems described in Section 6. Based on initial considerations of design requirements, the following other types of Grid Services requirements are likely candidates.

- **Job Configuration Management**: Service for storing parameters used to execute a job and user annotations for re-use (Grid WBS 1.3.5.1). This feature is currently implemented in the current XCAT Science Portal but will need to be redesigned as a Grid Service in order to facilitate sharing of job configurations among users.

- **Authentication Service**: The ability to authenticate using Grid credentials. The XCAT Science Portal currently supports a GSI and MyProxy interface for this.

- **File Management**: Service to stage input files (Grid WBS 1.3.3.9), interface with mass storage devices (Grid WBS 1.3.3.12), access replica catalog tools, etc. This will likely be the combination of several Grid Services. For example, Magda provides replica catalog access and GridFTP can be used to stage input files.

- **Monitoring:** Stores status messages, output, and errors in real time such that they can be retrieved and/or pushed to Grappa and then displayed to the user. The XCAT Science Portal can currently interface to the XEvent service (also developed by the Extreme! Computing Lab). Other monitoring services as those described in Sections 6 are likely to be accessed as well.

A second XCAT Science Portal redesign goal is support of multi-user access to the Grid Portal so that each user does not have to maintain their own web portal server but can still manage their own data separately from other users. A third goal more sophisticated parameter management interfaces, since ATLAS applications are controlled by a large number of parameters.

5.5 Grappa and Virtual Data

Grappa could be well placed to be a user interface to virtual data and in this role is similar to the NOVA work begun at Brookhaven Lab on "algorithm virtual data", AVD[10] and which will be pursued in Magda development. If virtual data with respect to materialization is to be realized, a data signature fully specifying the environment, conditions, algorithm components, inputs etc. required to produce the data must exist. These will be cataloged somehow (Virtual Data Language), somewhere (Virtual Data Catalog), or the components that make them up are cataloged and a data signature is a unique collection of these components constituting the 'transformation' needed to turn inputs into output. Grappa could then interface to the data signature and catalogs and allow you to 'open' a data signature and view it in a comprehensible [illegible]n, edit it, run it, etc. Take away the specific input/output data set(s) associated with a particular data signature and you have a more general 'prescription' or 'recipe' for processing inputs of a given type under very well defined conditions, and it will be very interesting to have catalogs of these -- both of the 'I want to run the same way Bill did last week' variety and 'official' or 'standard' prescriptions the user can select from a library.

5.6 Grappa Schedule

Table 7 Grappa work items and milestones

GriPhyN Code	ATLAS Grid WBS	Name	Description	Start	End
GG-I1		Multi-user Portal	Extend portal server to multi-user	Y2-Q1	Y2-Q4
GG-I2		Use scenarios	Analyze user scenario documents	Y2-Q1	Y2-Q1
GG-I3		Job launch extension	Grappa using other launch tools	Y2-Q2	Y2-Q4
GG-I4		Evaluation of Grid based file management systems (GFMS)	Continued evaluation and monitoring of developments in Grid-based file management systems such as Magda, SRB, Globus replica catalog service	Y2-Q2	Y2-Q4
GG-I5		Implement interface to GFMS	Design, prototype, implement and test Grappa interface to suitable GFMS	Y2-Q2	Y2-Q4
GG-I6		Condor DAGman interface	Implement DAGman functionality	Y2-Q2	Y2-Q4
GG-I7		Parameter management	Explore large parameter set management	Y2-Q1	Y2-Q4
Milestones					
GM-I1		Condor-G functionality	Demonstrate use of Condor-G from Grappa	7/1/02	
GM-I2		GFMS Evaluation	First evaluation of GFMS complete	4/1/02	
GM-I3		GFMS Interface	First release of GFMS interface with Grappa	7/1/02	

6 Performance Monitoring and Analysis

Performance monitoring and analysis is an important component necessary to insure efficient execution of ATLAS applications on the Grid. This component entails the following:

- Instrumenting ATLAS applications to get performance information such as event throughput and identifying where time is being spent in the application
- Installing monitors to capture performance information about the various resources (e.g., processors, networks, storage devices)
- Developing higher level services to take advantage of this sensor data, for example, to make better resource management decisions or to be able to visualize the current testbed behavior

- Developing models that can be used to predict the behavior of some devices or applications to aid in making decisions when more than one option is available for achieving a given goal (e.g., replication management)

Many tools will be used to achieve the aforementioned goals. Further, the performance data will be given in different formats, such as log files or data stored in databases. Additional tools will be developed to analyze the data in the different formats and visualize it as needed. The focus of this work will be on the U.S. ATLAS testbed.

We are leading the joint PPDG/GriPhyN effort in monitoring to define the use cases and requirements for a cross-experiment testbed. As part of the joint effort we have been gathering use cases to define requirements for the information system needed for a Grid-level information system, in part to answer questions such as these. The next step of this work will be to define a set of sensors for every facility to install, and to develop and deploy the sensors and their interface to the Globus Metacomputing Directory Service (MDS) and other components as part of the testbed. The services, needed to make execution on compute Grids transparent, will also be monitored. Such services include those needed for file transfer, access to metadata catalogs, and process migration.

Details about the Grid-level monitoring are given in Section 6.1 along with information about the visualization of this data using GridView. In addition, we have been evaluating and installing sensors to capture the needed data for our testbed facilities internally, and determining what information should be shared at the Grid level, and the best ways to do this, as detailed in Section 6.2. At the application level, much work has been done with Athena Auditor services to evaluate application performance on the fly, as described in 6.3. Section 6.4 discusses some higher level services work in prediction. Section 6.5 discusses work plans for a grid telemetry system.

6.1 Grid-level Monitoring

At the Grid-level, several different types of questions are asked of an information service. This can include scheduling-based questions, such as what is the load on a machine or network or what is the queue on a large farm of machines, as well as data-access questions like – where is the fastest repository I can download my file from?

We are defining a standard set of sensors to be installed on the testbed in order to address these types of questions, and to interface with the Globus information service, MDS. In addition, we are developing additional sensors as needed to conglomerate data on local farms, for example, and advertise this summary data to the grid.

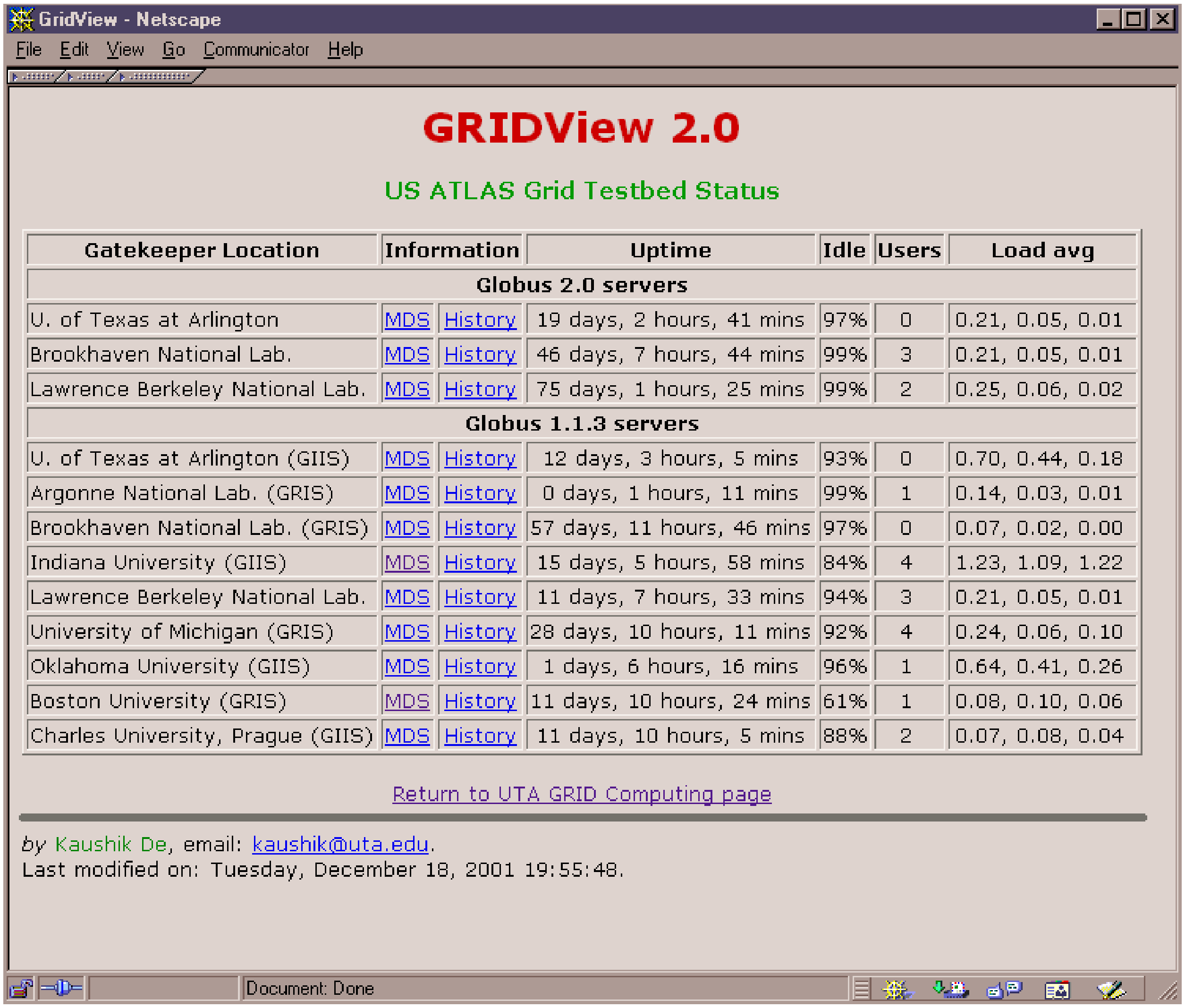
One area that has received a great deal of attention in the group already is monitoring network resources. There are two main types of network sensors – passive sensors that sniff on a network connection or active sensors that create network traffic to obtain information about network bandwidth, package loss, and round-trip time. There are many tools available for network monitoring, iperf, Network Weather Service, pingER and so on. We need to support the deployment of these testing and monitoring tools and applications, in association with the HENP network working group initiative, so that most of ATLAS major network paths can be adequately monitored. The network statistics should be included in Grid information service so that Grid software can choose the optimized path for accessing the virtual data.

In order to make better use of the data advertised by various sensors or tools, GridView was developed at the University of Texas at Arlington (UTA) to monitor the U.S. ATLAS Grid, first released in March, 2001. GridView provides a snapshot of dynamic parameters like CPU load, up time, and idle time for all Testbed sites. The primary web page, a snapshot shown in Figure 6-1 below, can be viewed at:

http://heppc1.uta.edu/kaushik/computing/Grid-status/index.html

GridView has gone through two subsequent releases. First, in summer 2001, MDS information from GRIS/GIIS servers was added. Not all Testbed nodes run a MDS server. Therefore, the front page continues to be filled using basic Globus tools. MDS information is provided in additional pages linked from this front page, where available.

Recently, a new version of GridView was released after the beta release of Globus 2.0 in November 2001. The U.S. ATLAS Testbed incorporates a few test servers running Globus 2.0 as well as every Testbed site running the stable 1.1.x version. GridView provides information about both types of systems integrated in a single page. Globus has changed the schema for much of the MDS information with the new release, but GridView can query and display either type. In addition, a MySQL server is used to store archived monitoring information. This historical information is also available through GridView.



GRIDView 2.0

US ATLAS Grid Testbed Status

Gatekeeper Location	Information		Uptime	Idle	Users	Load avg
Globus 2.0 servers						
U. of Texas at Arlington	MDS	History	19 days, 2 hours, 41 mins	97%	0	0.21, 0.05, 0.01
Brookhaven National Lab.	MDS	History	46 days, 7 hours, 44 mins	99%	3	0.21, 0.05, 0.01
Lawrence Berkeley National Lab.	MDS	History	75 days, 1 hours, 25 mins	99%	2	0.25, 0.06, 0.02
Globus 1.1.3 servers						
U. of Texas at Arlington (GIIS)	MDS	History	12 days, 3 hours, 5 mins	93%	0	0.70, 0.44, 0.18
Argonne National Lab. (GRIS)	MDS	History	0 days, 1 hours, 11 mins	99%	1	0.14, 0.03, 0.01
Brookhaven National Lab. (GRIS)	MDS	History	57 days, 11 hours, 46 mins	97%	0	0.07, 0.02, 0.00
Indiana University (GIIS)	MDS	History	15 days, 5 hours, 58 mins	84%	4	1.23, 1.09, 1.22
Lawrence Berkeley National Lab.	MDS	History	11 days, 7 hours, 33 mins	94%	3	0.21, 0.05, 0.01
University of Michigan (GRIS)	MDS	History	28 days, 10 hours, 11 mins	92%	4	0.24, 0.06, 0.10
Oklahoma University (GIIS)	MDS	History	1 days, 6 hours, 16 mins	96%	1	0.64, 0.41, 0.26
Boston University (GRIS)	MDS	History	11 days, 10 hours, 24 mins	61%	1	0.08, 0.10, 0.06
Charles University, Prague (GIIS)	MDS	History	11 days, 10 hours, 5 mins	88%	2	0.07, 0.08, 0.04

Return to UTA GRID Computing page

by Kaushik De, email: kaushik@uta.edu.
Last modified on: Tuesday, December 18, 2001 19:55:48.

Figure 6-1 GridView display of U.S. ATLAS Grid Testbed

6.2 Local Resource Monitoring

There are different monitoring needs on a wide-area (grid-level) system than on a local system. Primarily in this is the need for local data to be summarized up to the grid level system for scalability purposes. Dantong Yu at Brookhaven has been leading the effort in local resource monitoring for the U.S. ATLAS testbed.

The different resources used to execute ATLAS applications will be monitored to aid in accessing different options for the virtual data. Initially, the following resources will be monitored with different tools: system configuration, network, host information and important processes:

- System Configuration: Monitoring systems should perform a software and hardware configuration survey periodically and obtain the information on what software (version, producer) are installed on this system, what hardware is available. This data is then collected together for an entire set of local resources and advertised to the grid level to be used, for example, but a Grid scheduler to choose the right system environment for the system-depend ATLAS applications.

- Host/Device Monitoring: host information includes CPU load, Memory load, available memory, available disk space, and average disk I/O time. Once summarized for an entire farm of machines or advertised out as is for single resource, this information will help Grid scheduler and Grid user to choose computing resource to run ATLAS applications intelligently. In addition, an ATLAS facility manager can use this information for site management. The necessary information for Grid computing will be identified and deployed at ATLAS testbed.

- Process Monitoring: Process sensors monitor the running status of a process, such as number of this type of processes, number of users, queue lengths, etc.. one use of these sensors to have a threshold set up to trigger an alarm when the threshold is reached to prevent overloading system resources or help recover the system from failure.

The local resource monitoring effort is currently being coordinated with PPDG, GriPhyN, iVDGL, EU DataGrid and other HENP experiments to ensure that the local resource monitoring infrastructures satisfy the needs of Grid users and Grid applications.

6.3 Application Monitoring

The ATLAS applications will be instrumented at various levels to obtain performance information on how much time is spent with between accesses to data and used with different data.

First, some of the Athena libraries will be instrumented so to get detailed performance information about file access and file usage. For the case when the instrumentation overhead is small, the libraries can be automatically used when specified in a user's job script. For the case when the instrumentation overhead is large, the instrumented libraries must be specified by the user; such libraries will not be used by default.

Second, the Athena auditors will be used to obtain performance information. The auditors provide high-level information about execution of different Athena algorithms. Auditors are executed before and after the call to each algorithm, thereby providing performance information at the level of algorithm execution. Currently, Athena includes auditors to monitor the CPU usage, memory usage and number of events for each Athena algorithm. Athena also includes a Chrono & Stat service to profile the code (Chrono) and perform statistical monitoring (Stat). Hence, Athena will be instrumented at both the algorithm and libraries levels to obtain detailed performance data.

6.4 Performance Models

The trace data found in log files and performance databases (such as MDS and Prophesy) will be used to develop analytical performance models that can be used to evaluate different options related to access to virtual data. In particular, various techniques will be used such as curve fitting and detailed analysis and modeling of the core ATLAS algorithms. The models will be refined as more performance data is obtained. The models can be used to evaluate options such as is it better to obtain data from a local site for which it is necessary to perform some transformations to get the data in the desire format or access the data from remote sites for which one needs to consider the performance of given resources such as networks and the remote storage devices. The analytical models would be used to evaluate the time needed for the transformation based upon the system used for execution.

6.5 Grid Telemetry

A distinction is made between grid *instrumentation* and grid *telemetry*. At the fabric level, instrumented devices such as network components (data switches and routers) produce data for status and monitoring purposes. For example, data flowing from these devices is captured and used in problem management situations by Network Operation Centers (NOC) and for on-line and archival monitoring. Such data be of a temporal nature and may signal critical events such as equipment failure, bottlenecks and congestion, or the data may report performance measures such as bandwidth utilization along a given network link.

Extending the concept, telemetry data can be captured and sent to/from various sources for monitoring and input to resource allocation algorithms. At the application level, "counters" which record, for example, numbers of events in a production system for a particle physics simulation can be collected from distributed sources to be used for high level tracking and monitoring. At the middleware level, workflow managers and distributed batch systems such as Condor[11] may require (or provide) telemetry data to improve efficiency of operation or to take advantage of new resources as they become available. At lower levels, services indicating CPU utilization, the status of authentication services, host monitoring, data transfer (I/O load indicators), cache and archive storage utilization need to be collected to provide an information basis for job planning and resource estimation.

Several groups have developed toolkits which either produce or provide instrumentation hooks for grid telemetry data. The Internet End-to-end Performance Monitoring (IEPM) Group[12] has developed a set of tools to monitor data collection, site connectivity, and tools for monitoring packet loss and response time of registered sites within the network. The Indiana University Network Administration Suite is a collection of programs developed for the maintenance and management of IU campus networks as well as the Abilene, TransPAC, and STAR TAP networks[13]. The Netlogger[14] toolkit developed at NERSC provides a message passing library that enables real-time diagnosis of problems in complex high-performance distributed systems. The tool has been successfully used to debug low throughput or high latency problems in distributed applications. The system includes tools for generating precision event logs that can be used to provide detailed end-to-end application and system level monitoring, and tools for visualizing log data to view the state of the distributed system in real time. Open source Linux cluster management toolkits such as NPACI Rocks[15] provide monitoring data which can be useful if sourced to remote monitoring systems. What is missing from these toolkits is the surrounding infrastructure to collect, archive, and manage the information in formats suitable for high level problem management, diagnosis, and resource information systems.

A separate ITR proposal was funded to develop a grid telemetry data acquisition system which intends to build on these advances by providing an integrated collection system for monitoring, problem diagnosis and management, and resource decision making algorithms operating in a grid environment. Since telemetry data acquisition systems will be linked closely with their sources, it is important that development of such a system be made in close communication with application, middleware and fabric developers and engineers. Development of such a system within the context of particle physics data grids and research projects such as petabyte-scale virtual data grid research by GriPhyN provides this opportunity. This work, which will be coordinated closely with the monitoring activities of GriPhyN, PPDG, and others participating in the joint monitoring group, will thus support GriPhyN and iVDGL laboratory operations.

6.5.1 Prototype Grid Telemetry System

A prototype grid telemetry acquisition system is shown schematically in the Figure 6-2. Telemetry data is collected and organized into "pools", which could be distributed to provide redundancy and scalability. The pools consist of database servers and storage area networks connected to the external network over fast links, and some may have access to archival (tape) storage systems. The data residing in these pools would be accessible with a variety of tools, including web based applications for visualization and API's written in Java, Perl or Python. Netlogger may be used as a message passing service for the system.

Each layer in the distributed grid may be instrumented to source telemetry data. Instrumented applications in particle physics may report event statistics, error conditions, and performance data. Data recorded by the server can be queried by the planning, estimation, and execution layer to optimize throughout performance. Archival, transport and data caches can report status and other performance data to the servers, again to be used by the upper two layers. In addition, security services can be queried, and policy decisions for specific applications or grid users using information logged by the server. The fabric can be continuously monitored and both real-time and historical data for host status/performance, network performance, data cache capacities, for example, can be archived by the system.

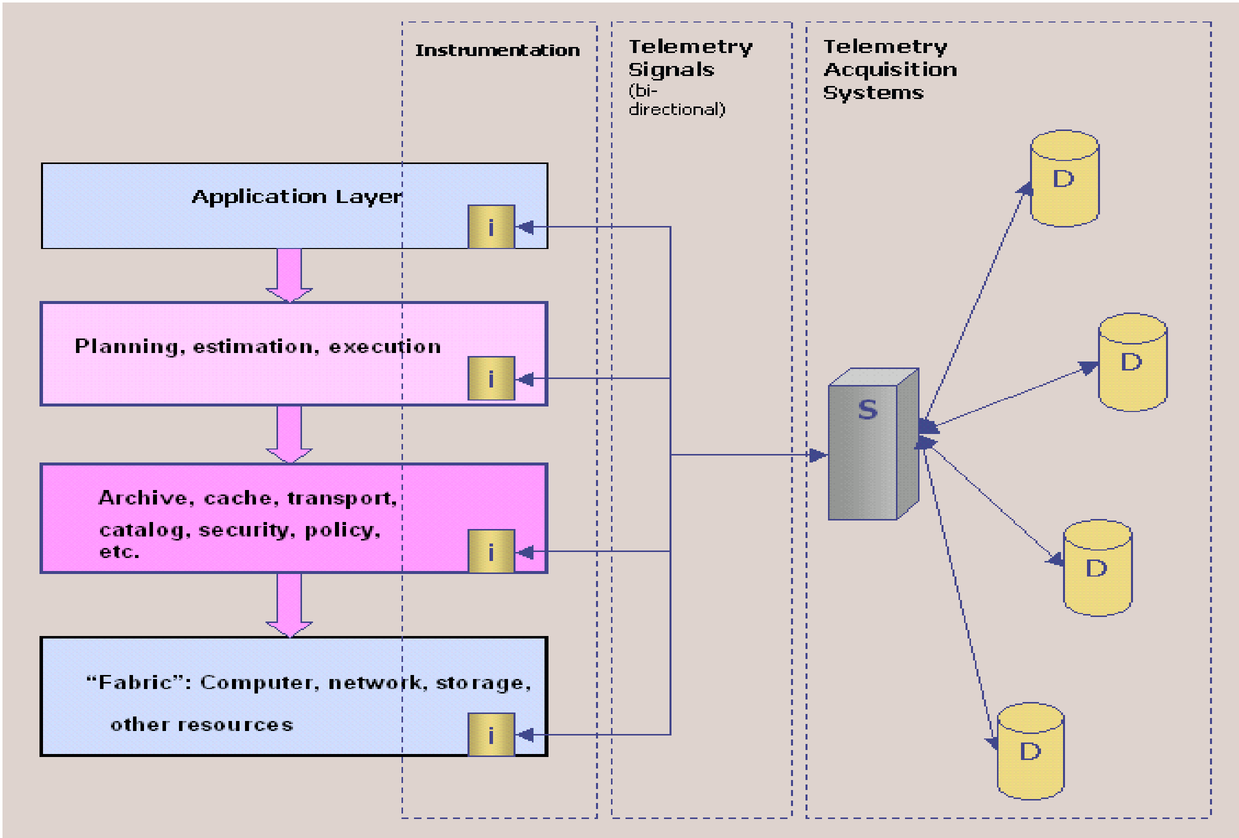


Figure 6-2 Components of a grid telemetry acquisition system. Instrumented modules within the grid application, middleware, and fabric levels (denoted by modules "i") transmit and receive telemetry signals from a server "S" providing access databases "D".

6.5.2 Telemetry Program of Work

The main outline of work is the following:

1. Collect monitoring and resource decision making requirements from core application physicists and grid middleware developers.
2. Identify and evaluate existing toolkits which source grid telemetry data.
3. Design the high level architecture for the grid telemetry data acquisition system and create technical design specification.
4. Prototype the design.
5. Implement grid telemetry data acquisition system. Facilities at Indiana University will be used. The hardware requirements are for dedicated database servers and storage area networks to provide high performance access to telemetry databases.
6. Instrument application level monitors for ATLAS core software.
7. Provide a resource service for grid and application developers.

The work will be carried out within the context of the U.S. ATLAS, GriPhyN, iVDGL, and international CERN testbeds.

6.6 Monitoring Schedule

Table 8 Monitoring work items and milestones

GriPhyN Code	ATLAS Grid WBS	Name	Description	Start	End
GG-M1		Evaluation	Initial evaluation of Grid monitoring services and requirements	Y2-Q1	Y2-Q4
GG-M1.1		Requirements analysis and specification	Requirements will be built from use case scenarios		
GG-M1.2		Evaluation of existing monitoring tools	For each type of local monitoring information, we will evaluate 2~3 monitoring tools.	Y2-Q1	Y2-Q1
GG-M1.3		Identify and select necessary mentoring tools.	Some tools may be modified and developed as part of this effort if they are not addressed by other work groups and not available commercially.	Y2-Q2	Y3-Q3
GG-M2		Tools	Development of tools, integration of identified monitoring services into Grid information service, etc.	Y3-Q1	Y4-Q4
GG-M2.1		Tool deployment Phase I	Deploy the required tools for testing at single sites	Y2-Q1	Y2-Q1
GG-M2.2		Tool Deployment – Phase II	Refine tool suite as needed, deploy on two sites with some feedback used to make some decisions; incorporate tools with information databases	Y2-Q1	Y2-Q2
GG-M2.3		Tool Deployment – Phase III	Incorporate tools into for inter-site monitoring	Y2-Q2	Y2-Q3
GG-M2.4		Tool Deployment – Phase IV	Incorporate tools into for inter-site monitoring	Y2-Q3	Y2-Q4
GG-M3		GridView	Grid information views	Y2-Q1	Y2-Q4
GG-M3.1		GridView – Phase I	Setup hierarchical GIIS server based on Globus 2.0	Y2-Q1	Y2-Q2
GG-M3.2		GridView – Phase II	Develop graphical tools for better organization of monitored information.	Y2-Q3	Y2-Q4
GG-M4		Grid Telemetry		Y2-Q2	Y3-Q4
GG-M4.1		Requirements gathering	Collect monitoring and resource decision making requirements from core application physicists and grid middleware developers.	Y2-Q2	Y2-Q3
GG-M4.2		Evaluation	Identify and evaluate existing toolkits which source grid telemetry data.	Y2-Q2	Y2-Q3
GG-M4.3		Design	Design the high level architecture for the grid telemetry data acquisition system and create	Y2-Q4	Y2-Q4

			technical design specification		
GG-M4.4		Prototype	Prototype the design.	Y2-Q4	Y3-Q1
GG-M4.5		Implement	Implement grid telemetry data acquisition system. Facilities at Indiana University will be used. The hardware requirements are for dedicated database servers and storage area networks to provide high performance access to telemetry databases.	Y3-Q1	Y3-Q2
GG-M4.6		Instrument	Instrument application level monitors for ATLAS core software	Y2-Q2	Y2-Q3
GG-M4.7		Production service	Provide a resource service for grid and application developers.	Y2-Q2	Y2-Q3
Milestones					
GM-M1		Monitoring Tool X evaluation	Evaluation of tool X completed	End Y2-Q4	
GM-M2		Deploy monitoring tools	For each type of local monitoring (network, host, configuration, important service), at least one tool should be identified and deployed at each individual ATLAS testbeds.	End Y2-Q4	
GM-M3		Construct Performance databases	Importance performance trace data should be archived in databases. Integrate the database into Grid information services.	End Y2-Q4	
GM-M4		First integration into Athena Auditor package	First test integration of GriPhyN monitor tools with Athena Auditor services	End Y2-Q3	

7 Grid Package Management – Pacman

If ATLAS software is to be smoothly and transparently used across a shifting Grid environment, we must also gain the ability to reliably define, create and maintain standard software environments that can be easily moved from machine to machine. Such environments must not only include standard ATLAS software via CMT and CVS, must able also include a large and growing number of "external" software packages as well as Grid software coming from GriPhyN itself. It is critical to have a systematic and automated solution to this problem. Otherwise, it will be very difficult to know with confidence that two working environments on the Grid are really equivalent. Experience has shown that the installation and maintenance of such environments is not only labor intensive and full of potential for errors and inconsistencies, but also requires substantial expertise to install and configure correctly.

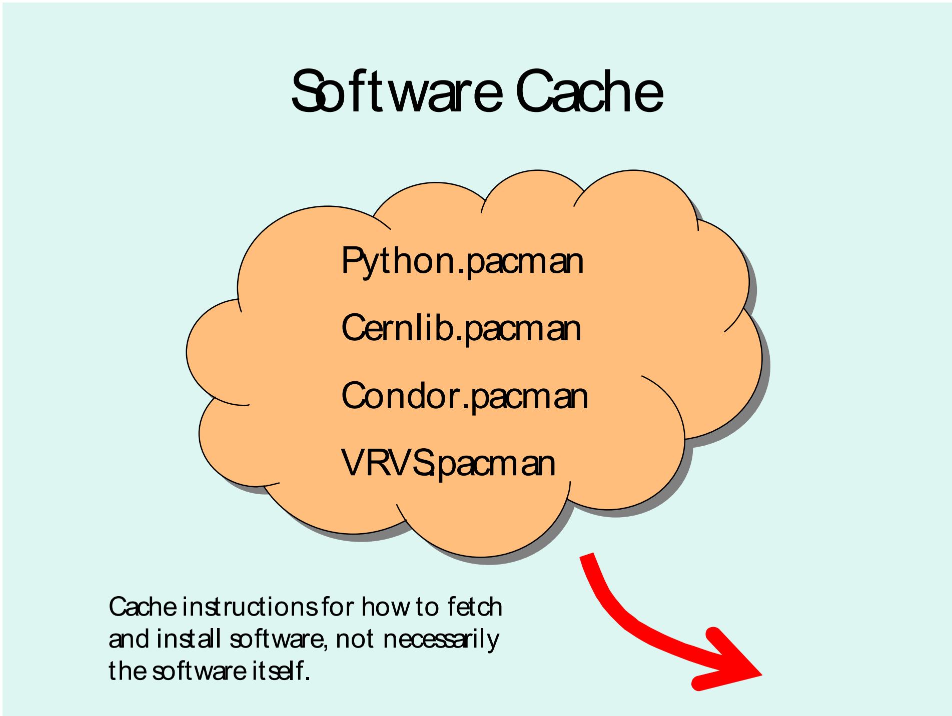
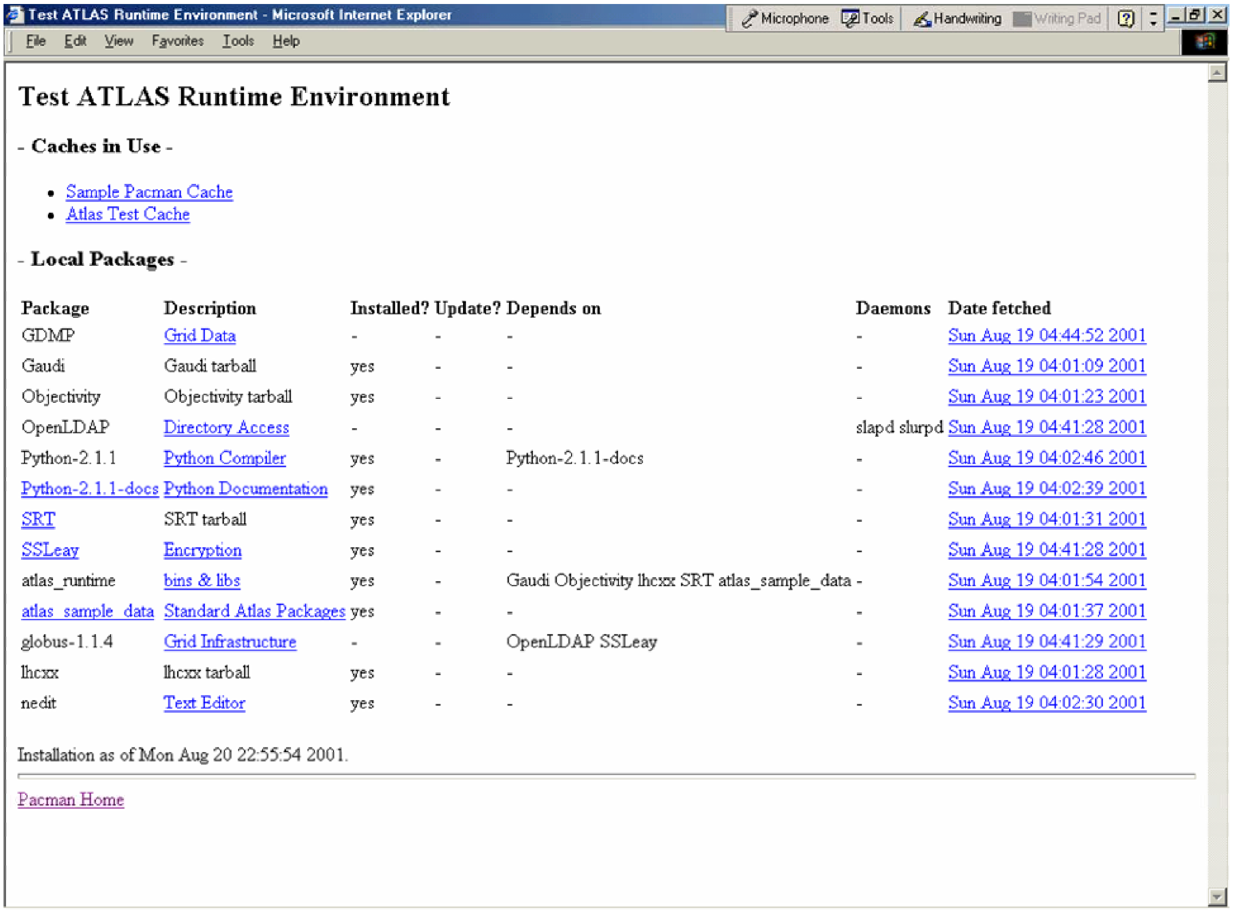


Figure 7-1 Pacman -- Package Management System components

To solve this problem we propose to effectively raise the problem from the individual machine or cluster level to the Grid level. Rather than having individual ATLAS sites work through the various installation and update procedures, we can have individual experts define how software is fetched, configured and updated and publish these instructions via "trusted caches." By including dependencies, we can define complete named environments which can be automatically fetched and installed with one command and which will result in a unified installation with common setup script, pointers to local and remote documentation and various such conveniences. Since a single site can use any number of caches together, we can distribute the expertise and responsibility for defining and maintaining these installation procedures across the collaboration. This also implies a shift in the part of Unix culture where individual sites are expected to work through any installation problems that come up in installing third party software. The responsibility for an installation working must, we feel, be shifted to the "cache manager" who defined the installation procedure to begin with. In this way, problems can be fixed once by an expert and exported to the whole collaboration automatically.

Over the next year or so, and particularly in order to prepare for Data Challenge 1, we will use an implementation of the above ideas called "Pacman" to define standard ATLAS environments which can be installed via caches. This will include run-time ATLAS environments, full development environments and project specific user defined environments. In parallel, we will work with the VDT distribution team and with Globus to develop a second-generation solution to this problem that can be more easily integrated with the rest of the GriPhyN Grid tools.


Test ATLAS Runtime Environment - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Test ATLAS Runtime Environment

- Caches in Use -

- Sample Pacman Cache
- Atlas Test Cache

- Local Packages -

Package	Description	Installed?	Update?	Depends on	Daemons	Date fetched
GDMP	Grid Data	-	-	-	-	Sun Aug 19 04:44:52 2001
Gaudi	Gaudi tarball	yes	-	-	-	Sun Aug 19 04:01:09 2001
Objectivity	Objectivity tarball	yes	-	-	-	Sun Aug 19 04:01:23 2001
OpenLDAP	Directory Access	-	-	-	slapd slurpd	Sun Aug 19 04:41:28 2001
Python-2.1.1	Python Compiler	yes	-	Python-2.1.1-docs	-	Sun Aug 19 04:02:46 2001
Python-2.1.1-docs	Python Documentation	yes	-	-	-	Sun Aug 19 04:02:39 2001
SRT	SRT tarball	yes	-	-	-	Sun Aug 19 04:01:31 2001
SSLeay	Encryption	yes	-	-	-	Sun Aug 19 04:41:28 2001
atlas_runtime	bins & libs	yes	-	Gaudi Objectivity lhcxx SRT atlas_sample_data	-	Sun Aug 19 04:01:54 2001
atlas_sample_data	Standard Atlas Packages	yes	-	-	-	Sun Aug 19 04:01:37 2001
globus-1.1.4	Grid Infrastructure	-	-	OpenLDAP SSLeay	-	Sun Aug 19 04:41:29 2001
lhcxx	lhcxx tarball	yes	-	-	-	Sun Aug 19 04:01:28 2001
nedit	Text Editor	yes	-	-	-	Sun Aug 19 04:02:30 2001

Installation as of Mon Aug 20 22:55:54 2001.

Pacman Home

Figure 7-2 Package cache display in Pacman

7.1 Pacman Schedule

Table 9 Pacman work items and milestones

GriPhyN Code	ATLAS Grid WBS	Name	Description	Start	End
GG-P1		Pacman distribution of Globus 2	Configured as needed for DC1	Y2-Q1	Y2-Q2
GG-P2		Pacman distribution of VDT 1.0	Working with Miron Livny's team	Y2-Q1	Y2-Q3
GG-P3		Feedback Pacman experience to GriPhyN CS teams	Work with GriPhyN CS teams to develop second generation solutions to grid package management	Y2-Q1	Y3-Q4
GG-P4		All 3d party software needed by Atlas distributed with Pacman		Y2-Q1	Y2-Q2
GG-P5		Pacman integrated with CMT		Y2-Q2	Y2-Q3

GriPhyN Code	ATLAS Grid WBS	Name	Description	Start	End
GG-P6		Pacman more general dependences implemented		Y2-Q2	Y2-Q2
GG-P7		Caches setup at BNL, BU, Indiana, UT Arlington, LBNL, Michigan		Y2-Q1	Y2-Q1
Milestones					
GM-P1		ATLAS AFS, runtime and stand along development environments delivered with Pacman.	Single operation of full installation of ATLAS environments on Linux and Sun Solaris.	Y2-Q2	

8 Security and Accounting Issues

We will work with the existing GSI security infrastructure to help the Testbed groups deploy a secure framework for distributed computations. The GSI infrastructure is based on the Public Key Infrastructure (PKI) and uses public/private key pairs to establish and validate the identity of Grid users and services. The system uses X.509 certificates signed by a trusted Certificate Authority (CA). Presently U.S. ATLAS Testbed sites use the Argonne/Globus CA, but will begin accepting ESNet CA certificates. By using the GSI security infrastructure we will be compatible with other Globus-based projects, as well as adhering to a de-facto standard in Grid computing. We will work in close collaboration with ESNet and PPDG groups working on CA issues to establish and maintain Grid certificates throughout the testbeds. We will support and help develop a Registration Authority for ATLAS – GriPhyN users.

A related issue is the development of an authorization service for resources on the testbed. Within Globus, there is much research on-going effort[16] which we will closely follow and support when these services become available.

9 Site Management Software

The LHC computing model implies a tree of computing centers where "Tier X" indicates depth X in the tree. For example, Tier 0 is CERN, Tier 1 is Brookhaven National Laboratory, and Boston University and Indiana University are "Tier 2" centers, etc. University groups are at the Tier 3 level and Tier 4 is meant to be individual machines. While the top of this tree is fairly stable, we must be able to add Tier 3 and Tier 4 nodes coherently with respect to common software environment, job scheduling, virtual data, security, monitoring and web pages while guaranteeing that there is no disruption of the rest of the tree as nodes are added and removed. To solve this problem we propose to define what a Tier X node consists of in terms of installed ATLAS and Grid software and to define how the Grid tools are connected to the existing tree. Once this is done, we propose to construct a nearly automatic procedure (in the spirit of Pacman or successors) for adding and removing nodes from the tree. Over the next year, we will gain enough experience with the top nodes of tree of Tiers to understand how this must be done in detail. In 2002, we propose to construct the software that nearly automatically adds Tiers to the tree.

10 Testbed Development

10.1 U.S. ATLAS Testbed

The U.S. ATLAS Grid Testbed is a collaboration of ATLAS U.S. institutions that have agreed to provide hardware, software, installation support and management of collection of Linux based servers interconnected by the various U.S. production networks. The motivation was to provide a realistic model of a Grid distributed system suitable for evaluation, design, development and testing of both Grid software and ATLAS applications to run in a Grid distributed environment. The participants include designers and developers from the ATLAS core computing groups and collaborators on the PPDG and GriPhyN projects. The original (and current) members are the U.S. ATLAS Tier 1 computing facility at Brookhaven Laboratory, Boston University and Indiana University (the two prototype Tier 2 centers), Argonne National Laboratory HEP division, LBNL (PDSF at NERSC), the University of Michigan, Oklahoma University and the University of Texas at Arlington. Each site agreed to provide at least one Linux server based on Intel X86 running Red Hat version 6.x OS and Globus 1.1.x gatekeeper software. Each site agreed to host user accounts and access based on the Globus GSI x509 certificate mechanisms. Each site agreed to provide a native or AFS based access to the ATLAS offline computing environment, sufficient CPU and Disk resources to test Grid developmental software with ATLAS codes. Each site volunteers technical resource people to install and maintain a considerable variety of infrastructure for the Grid environment and developed software by the participants. In addition, some of the sites choose to make the Grid gatekeepers as gateway to substantial local computing resources via Globus job manager access to LSF batch queues or Condor pools. This has been facilitated and managed by bi-weekly teleconference meetings over the past 18 months. The project began with a workshop[17] on developing a GriPhyN – ATLAS testbed at Indiana University in May 2000. A second testbed workshop[18] was held at the University of Michigan in February 2001.

The work of the first year included installation and operation of an eight node Globus 1.1.x Grid; installation and testing of components of the U.S. ATLAS distributed computing environment, development and testing of PDSF developed tools. These included Magda, GDMP, and alpha versions of the Globus DataGrid Tool sets. Testing and evaluation of the GRIPE account manager[19], the development and testing of network performance measurement and monitoring tools. The development, installation and routine use of Grid resource tools e.g. GridView. The development and testing of new tool for distribution, configuration and installation of software: Pacman. The testing of the ATLAS Athena code ATLFast writing and reading to Objectivity databases on the testbed gatekeepers; testing and preparations for installation of Globus 2.0 and associated DataGrid tools to be packaged in the GriPhyN VDT1.0; preparations and coordination with the European DataGrid testbed, and coordination with the International ATLAS Grid project. The primary focus has been on developing infrastructure and tools.

The goals of the second year will include: Continuing the work on infrastructure and tools installation and testing. A coordinated move to a Globus 2.0 based Grid. Providing a reliable test environment for PPDG, GriPhyN and ATLAS core developers. The adoption and support of a focus on ATLAS application codes designed to exploit the Grid environment and this testbed in particular. A principal mechanism will be the full participation in the ATLAS Data Challenge 1 (DC1) exercise. This will require the integration of this testbed into the EU DataGrid and CERN Grid testbeds. During the second half we expect to provide a prototype Grid based production data access environment to the simulation data generated as part of DC1, thus a first instance of the U.S. based distributed computing plan for U.S. offline analysis of ATLAS data. To achieve these goals we will evolve the US testbed into two pieces: an eight site prototype-production grid (stable, user-friendly, production and services oriented) and (4-8 site) test-bed grid (with traditional test-bed properties for developmental software and quick turn-around reconfiguration etc).

10.2 iVDGL

The iVDGL project will provide the computing platform upon which to evaluate and develop distributed Grid services and analysis tools developed by GriPhyN. Two ATLAS – GriPhyN institutions will develop prototype Tier 2 centers as part of this project, Indiana University and Boston University. Resources at those facilities will not only support ATLAS specific applications, but also the iVDGL/GriPhyN collaboration at large, both physics applications and CS demonstration/evaluation challenges. In addition, other sites within the iVDGL, domestic and international, will be exploited were possible for wide area job execution using GriPhyN developed technology.

10.3 Infrastructure Development and Deployment

Below are some specific software components which need to be configured on the Testbed.

Testbed configuration during Year 2:

- VDT1.0 (Globus 2.0Beta, Condor 6.3.1, GDMP 2.0)
- ATLAS Software releases 2.0.0 and greater
- Magda
- Objectivity 6.1
- Pacman
- Test suite for checking proper installation
- Documentation, web-based, for the Testbed configuration at each site
- The above packaged with Pacman

We will begin by deploying VDT services, ATLAS software, and ATLAS required external packages on a small number of machines at 4 to 8 sites. At each site, skilled personnel are identified as points of contact. These are: ANL (May), BU (Youssef), BNL (Yu), IU (Gardner) in first 3 months (required), with UTA, NERSC, U of Michigan, and OU following as their effort allows. Additional parts of the work plan include:

1. Identify a node at CERN to be included in early testbed development. This will include resolution of CA issues, and accounts. May leads.
2. Define simple ATLAS application install procedure, neatly package up a simple example using Pacman, including documentation, with simple run instructions and a readme file. Sample data file and a working Athena jobs are needed. Ideally, several applications will be included. Shank, Youssef, and May lead.
3. Provide an easy setup for large scale batch processing. This will include easy account and certificate setup, disk space allocations, and access to other site specific resources. Ideally this will be done with a submission tool, possibly based on Grappa or included within Magda. Wenaus, Smallen lead.

10.4 Testbed Schedule

Table 10 Testbed work items and milestones

GriPhyN Code	ATLAS Grid WBS	Name	Description	Start	End
GG-T1		GT1 testbed	Establish a 4-8 site testbed in parallel	Y2-Q1	Y2-Q1
GG-T2		Migrate 8 site testbed (GT1.1.x) to GT2	Establish proto-production US ATLAS Grid of 8 sites; uniform installation of VDT 1.0 and other Grid tools.	Y2-Q2	Y2-Q3
GG-T3		CA migration and global integration	Migrate both testbed and proto-production sites to ESnet CA and integration with EU DG, CERN, and other ATLAS Grids	Y2-Q2	Y2-Q3
GG-T4		DC1 participation	Integration with backend compute and data services; execute DC1 tests		
GG-T5		DC1 data services	Establish and execute production services for DC1 data analysis on the proto-production Grid		
GG-T6		SC demo preparations	Configuration and preparations for SC 2002 demonstrations	Y2-Q4	Y3-Q1
Milestones					
GM-T1		GT1 testbed	Demonstration GT1 testbed to be operational DC1 development	10/01/01	
GM-T2		VDT 1.0	VDT 1.0 deployed on all sites	1/1/02	
GM-T3		CERN Testbed node	Installation, configuration of a dedicated Grid Testbed node at CERN	1/1/02	
GM-T4		GT2 testbed	Demonstration GT2 testbed to be operational for DC1 analysis	7/1/01	
GM-T5		SC demo	SC demo preparations complete	11/1/01	

11 ATLAS – GriPhyN Outreach Activities

We plan to join GriPhyN and iVDGL outreach efforts with a number of on-going efforts in high energy physics, including, the ATLAS Outreach committee and QuarkNet.

Within GriPhyN – iVDGL, Hampton University will be building a Tier 3 Linux cluster as part of the iVDGL Outreach effort. HU is also a member of ATLAS and is heavily involved in construction of the Transition Radiation Tracking detector for ATLAS.

Specific outreach work items:

1. Provide ATLAS liaison and support for the GriPhyN Outreach Center[20].
2. Provide support and consultation for installation of GriPhyN VDT and ATLAS software at HU.
3. Interact with Hampton University students and faculty running ATLAS applications on the Grid.
4. Support HU and other iVDGL institutions in establishing a GriPhyN – iVDGL QuarkNet educational activities.

11.1 Outreach Schedule

Table 11 Outreach work items and milestones

GriPhyN Code	ATLAS Grid WBS	Name	Description	Start	End
GG-O1	NA	Web support	Provide web based information for ATLAS – GriPhyN activities for education and outreach purposes	Y2-Q1	Y3-Q4
GG-O2	NA	VDT support	Provide support for VDT installation, guidance at Hampton University	Y2-Q3	Y3-Q4
GG-O3	NA	QuarkNet	Interact with EO outreach faculty developing GriPhyN QuarkNet programs for high school teachers and students	Y2-Q3	Y3-Q4
GG-O4	NA	ATLAS Software	Provide support to HU and other outreach institutions requiring assistance with ATLAS software installation and support.	Y2-Q3	Y3-Q4
Milestones					
GM-O1	NA	Web page	GriPhyN – ATLAS outreach webpage complete	Y2-Q4	
GM-O2	NA	ATLAS job submission	Demonstration of ATLAS Monte Carlo generation, reconstruction, analysis codes executing by students and faculty located at outreach institutions using Grappa interface	Y2-Q4	

12 Summary, Challenge Problems, Demonstrations

Below we give a summary and schedule of GriPhyN – ATLAS goals, challenge problems, and demonstrations.

12.1 ATLAS Year 2 (October 01 – September 02)

12.1.1 Goals Summary

As discussed previously, during Year 2 ATLAS Data Challenges 0 and 1 occur and ATLAS will build up a large volume of data based on the most current detector simulation model and processed with newly developed reconstruction and analysis codes. This is an important opportunity for GriPhyN as there will be a demand throughout the collaboration for distributed access to the resulting data collections, particularly the reconstruction and analysis products which contain file sets suitable for analysis. These will occur during the second phase of DC1, ending sometime in July 2002. In close collaboration with PPDG we will integrate VDT data transport and replication tools, with particular focus on reliable file transfer tools, into a distributed data access system serving the DC data sets to ATLAS users. We will also use on-demand regeneration of DC reconstruction and analysis products as a test case for virtual data by materialization. These exercises will test and validate the utility of Grid tools for distributed analysis in a real environment delivering valued services to end-users.

Collaboration with the International ATLAS Collaboration, and the LHC experiments overall is an important component of the subproject. In particular, developing and testing models of the ways ATLAS software integrates with Grid middleware is a critical issue. The international ATLAS collaboration, with significant U.S. involvement, is responsible for developing core software and algorithms for data simulation and reconstruction. The goal is the successfully integrate Grid middleware with the ATLAS computing environment in a way that provides a seamless Grid-based environment used by the entire collaboration.

12.1.2 Challenge Problem I: DC Data Analysis

Data Challenge 1 (February 2002 - July 2002) involves producing 1% of the full-scale solution using existing core ATLAS software. The execution will run in a traditional, linear fashion without Grid interactions. Event generation (using the PYTHIA generator package from the Lund group) will be invoked from the Athena framework, while the Geant3-based detector simulation will use the Fortran-based program. The result will be data sets that are of interest to users in general, generating 10^7 events using O(1000) PC's, with a total data size of 25-50 TB.

Planning and execution of CP I will involve:

1. Tagging the data sample with physics generator metadata tags, and storage in a metadata file system for subsequent collection browsing. The Grenoble group is leading this effort; the interface with Magda is being done by PPDG.

2. Serving the data (and metadata) using Grid infrastructure file access. At a bare minimum, a well organized website will be used to supply first time ATLAS users a portal to the DC collections. A more advanced solution, similar to present Magda functionality,

which incorporates physics metadata on a file-by-file basis will be used, optionally fitted with a command line interface to provision files.

3. Providing first time users with instructions and guidance for using Grid based analysis tools. Working examples of Grid analysis sessions, complete with scripts and user algorithms, will need to be supplied.

4. Data storage capacities at each site will need to be clearly defined, with specifications regarding data types and access policy made clearly available to users and production managers.

5. Job submission tools with minimal smarts will be developed, using highly extensible frameworks. This will at first be Grappa (or its equivalent), used as remote job submission interface. Minimal scheduling smarts will be added once the basic submission infrastructure is in place. For example, components to identify where the reconstruction input file collections are located, and components which co-allocate CPU resources. A possible solution involves layering on top of DAGman.

6. Coherent monitoring for the system as a whole will need to be developed:
   - Components which gather and parse Condor log files
   - GridView
   - Real-time network monitoring with graphical display

Metrics for success will include working demonstrations of analysis tasks which produce physics histograms from large collections of DC data, providing much feedback about event throughput, performance, and status throughout the process. There should be opportunity for much user feedback, as users coming into the system with the motivation to extract physics plots will likely be quite vocal (and helpful) as they experience the Grid for the first time.

12.1.3 Challenge Problem II: Athena Virtual Data Demonstration

The following demonstration will be implemented using software developed within the Adagio effort. We will also attempt to employ existing virtual data infrastructure, such as VDL and VDC as developed by CMS and LIGO GriPhyN teams.

We define a query to be an Athena-based consumer of an ATLAS Monte Carlo data (such as from ATLFAST) along with a tag that identifies the input dataset needed. In an environment in which user Algorithms are already available in local shared libraries, this may simply be a JobOptions file, where one of the JobOptions (like event selection criteria) is allowed to vary.

Three possibilities will be supported by GriPhyN virtual data infrastructure as implemented with core Athena code supported by VDL, VDC and the Adagio set of extensions to Athena:

1. The dataset exists as a file or files in some place directly accessible to the site where the consuming program will run. In this case, the Athena service that is talking to GriPhyN components (e.g., an EventSelector) will be pointed to the appropriate file(s).

2. The data set exists in some place remote to the executable. The data will be transferred to a directly accessible site, after which processing will proceed as in 1. This is virtual data transparency with respect to location.

3. The data set must be generated. In this case, a recipe to produce the data is invoked. This may simply be a script that takes the dataset selection tag as input, sets JobOptions based on that tag, and runs an Athena-based ATLFAST simulation to produce the data. Once the dataset is produced, processing continues as in 1. This is virtual data with respect to materialization (existence).

Metrics for success will include demonstrated executions for each of the possibilities outlined above, with a verification/monitoring algorithms used to certify results based on pre-calculated sets of histograms.

12.1.4 Demonstrations for ATLAS Software Weeks

A preliminary demonstration of Grappa functionality is planned to be in place for the World-Wide Computing Session of the first ATLAS Software Week in March, 2002. This should include:

1. Authentication to a personal XCAT portal
2. Design of Athena Monte Carlo generation analysis session
3. Selection of several grid resources from the U.S. Testbed, including Condor resources accessed through the Globus Job Manager.
4. Automatic generation of random number seeds for individual jobs
5. Automatic physical file name generation
6. Display of execution monitoring data
7. Preliminary interface displays to Magda metadata and physical replicas for data stored in the Testbed Grid.
8. Demonstration of GridView description, monitoring of Testbed Grid resources

Metrics for success will include real-time demonstration of the Athena analysis chain for user analysis, resulting in displays of physics plots and event throughput monitoring / statistical information during the demonstration.

12.1.5 Demonstration for SC2002

SC2002 will be held in Baltimore, November 16-22, 2002. Demonstration of Grid-based data analysis using ATLAS software and a significant number of Grid sites, beginning first with Tiers 0-2, later expanding to ATLAS Tier 3 sites, and later to non-ATLAS sites such as other sites in the iVDGL. To include:

1. Full chain production and analysis of ATLAS Monte Carlo event data
2. Illustration of typical physicist analysis sessions
3. Graphical monitoring display of event throughput throughout the Grid
4. Live update display of distributed histogram population from Athena
5. Illustration of Challenge Problem I, analysis of DC1 data collections

6. Illustration of Challenge Problem II, virtual data re-materialization from Athena
7. Illustration of Grappa job submission and monitoring examples.

Metrics for success will include a working demonstration which meets the above listed functionality requirements.

12.2 ATLAS Year 3 (October 02 – September 03)

12.2.1 Goals Summary

The first major goal of ATLAS Data Challenge 2 (January 2003 to September 2003) is to evaluate variations to the LHC Computing Model, as currently be debated within the international ATLAS World Wide Computing Group, which is overseen by the National Computing Board (NCB).

During DC2, we will compare a “strict Tier" model in which full copies of ESD data (Event Summary Data) reside on massive tape storage systems and disk at each ATLAS Tier 1 site, to a "cloud" model where the full ESD is shared among multiple sites. The latter results in a complete sample of ESD data on disk at any time. The two models may imply vastly different analysis access patterns, and could result in significant re-direction of facilities resources from computing cycles to network bandwidth capacity, for example. MONARC studies of the new models will be helpful, but the DC will provide the empirical experience from which to complete the design of the LHC computing infrastructure, leading up to the LHC turn-on.

The second major goal of Year 3 is to push development of virtual data technologies for ATLAS, building on the early successes within GriPhyN research on VDL and VDC for CMS and LIGO. At this time the ATLAS core software will be better suited for this type of work. Also needed are tools to evaluate the virtual data reconstruction methods, and algorithms to evaluate their success.

12.2.2 Challenge Problem III: Grid Based Data Challenge

DC2 will use Grid middleware in a production exercise scaled at 10% of the final system. CP-III will require large scale, robust Grid production and analysis tools for data management, job management on distributed resources, security and monitoring.

12.2.3 Challenge Problem IV: Virtual Data Tracking and Recreation

The goal of CP-IV will be the development and demonstration of virtual data re-creation, that is, the ability to rematerialize data from a query using a virtual data language and catalog. Some issues to be resolved:

1. Identify which parameters need tracking to specify re-materialization (things making up the data signature such as code release, platform and compiler dependencies, external packages, input data files, user and/or production cuts).

2. Identify a metric for evaluating the success of re-materialization. For example, what

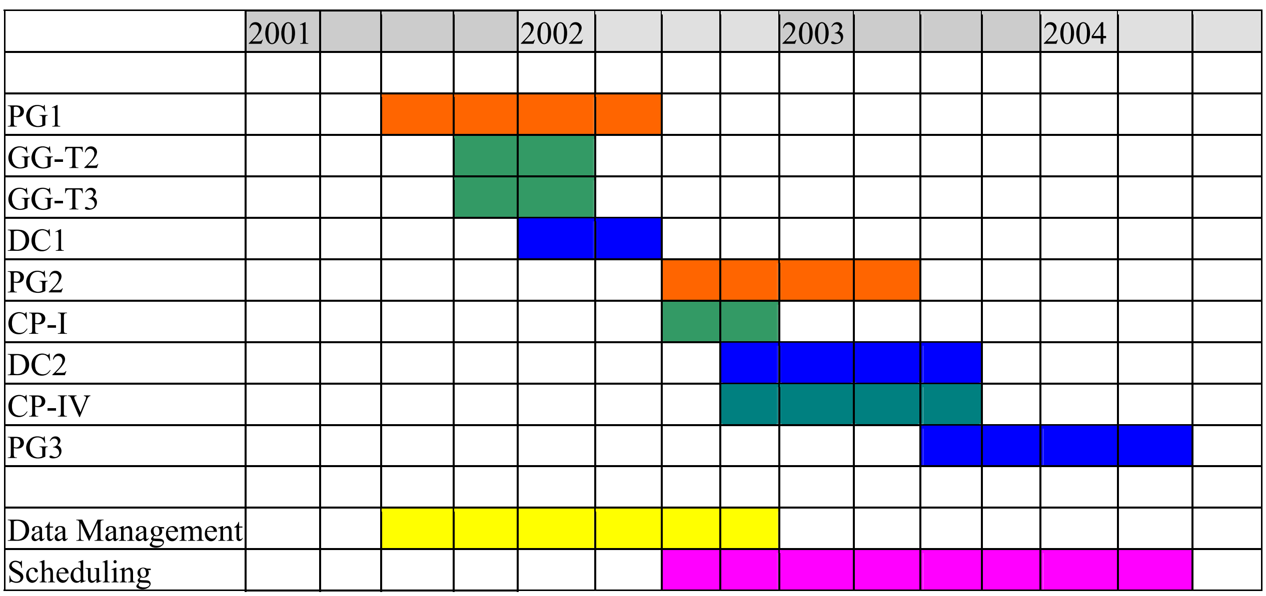
constitutes a successful reproduction of data products? Assuming bit-by-bit comparison of identical results is impractical, what other criteria can be identified which indicate "good enough" reconstruction? For example, statistical confidence levels obtained by comparison of materialized histograms with reference versions would provide statistics-based criteria for success.

12.3 Overview of Major Grid Goals

Here we list schedules for some of the major GriPhyN goals (GG), challenge problems (CP), in relation to PPDG (PG) and to ATLAS data challenges (DC).

• June 01 - July 02	PG1	Development of Grid based data management with Magda
• Oct 01 – March 02	GG-T2	VDT 1.0 deployment and basic infrastructure
• Dec 01 – Feb 02	GG-T3	Integration of CERN testbed node into US ATLAS testbed
• Jan 02 – July 02	DC1	Data creation, use of Magda, Tier 0-2
• July 02 – June 03	PG2	Job management, Grid job submission
• July 02 – Dec 02	CP-I	Serving data from DC1 to universities, simple Grid job sub.
• Dec 02 – Sept 03	DC2	Grid resource mgmt, data usage, smarter scheduling
• Dec 02 – Sept 03	CP-IV	Dataset re-creation, metadata, advanced data Grid tools
• July 03 – June 04	PG3	Smart job submission, resource usage

Table 12 ATLAS - GriPhyN and PPDG Schedules



13 Project Management

GriPhyN software development activity, as it pertains to ATLAS, has components in both Software and Facilities subprojects within the U.S. ATLAS Software and Computing Project.

The Level 1 manager of the U.S. ATLAS S&C project is John Huth of Harvard University, also a member of the GriPhyN – ATLAS team. The Level 2 project manager for Software is Torre Wenaus of Brookhaven National Laboratory, who is the ATLAS PPDG project lead and is collaborating with GriPhyN. The Level 2 project manager for Facilities is Rich Baker of BNL who also collaborates with GriPhyN; Rich also supervises Dantong Yu, who coordinates monitoring activities within U.S. ATLAS. The Level 3 project manager for Distributed IT Infrastructure is Rob Gardner of Indiana University, also the project lead for GriPhyN – ATLAS. A Project Management Plan describes the organization of the U.S. ATLAS S&C project. Liaison personnel for GriPhyN have been named for Computer Science (Jennifer Schopf, Globus team) and Physics (Rob Gardner).

13.1 Liaison

Within the U.S. ATLAS S&C project, liaison duties are referenced in Grid WBS 1.3.2 (liaison between U.S. ATLAS software and external distributed computing software efforts). The work items entailed in these roles include:

1. Presentations at various computing reviews (EAC, DOE/NSF, etc.) by appropriate liaison
2. Coordination between U.S. ATLAS S&C personnel and others within the GriPhyN Collaboration
3. Planning, organization of GriPhyN project goals specific to ATLAS

13.2 Project Reporting

Monthly reports will be submitted to the GriPhyN project management. Periodic reviews will be made by the GriPhyN EAC and by the U.S. LHC Project Office. In addition, annual reports will be generated which will give an accounting of progress on project milestones and deliverables. Additional reports, such as conference proceedings and demonstration articles, will be filed with the GriPhyN document server.

14 References

[1] ATLAS: A Torroidal LHC Apparatus, homepage: http://atlas.web.cern.ch/Atlas/

[2] The Athena architecture homepage:
http://atlas.web.cern.ch/Atlas/GROUPS/SOFTWARE/OO/architecture/General/index.html

[3] The Gaudi Project: http://proj-gaudi.web.cern.ch/proj-gaudi/

[4] "ATLAS Detector and Physics Performance", Technical Design Report (ATLAS Collaboration), LHCC 99-14/15, May 1999.
http://atlasinfo.cern.ch/Atlas/GROUPS/PHYSICS/TDR/access.html

[5] U.S. ATLAS Grid Planning page:
http://ATLASsw1.phy.bnl.gov/Planning/usGridPlanning.html

[6] The XCAT Science Portal, Sriram Krishnan, et. al., in proceedings of Supercomputing 2001.

[7] Programming the Grid: Distributed Software Components, P2P and Grid Web Services for Scientific Applications, D. Gannon, et. al, to appear in the IEEE Journal on Cluster Computing, Special Issue on HPDC01.

[8]Grappa: Grid Access Portal for Physics Experiments:

- Homepage: http://lexus.physics.indiana.edu/griphyn/Grappa/index.html
- Scenario document http://lexus.physics.indiana.edu/~griphyn/Grappa/Scenario1.html

[9] CCA: Common Component Architecture forum: http://www.cca-forum.org/ ;
At Indiana University: http://www.extreme.indiana.edu/ccat/

[10]Algorithmic Virtual Data (NOVA project), at BNL:
http://ATLASsw1.phy.bnl.gov/cgi-bin/nova-ATLAS/clientJob.pl

[11] Condor Project: http://www.cs.wisc.edu/condor/

[12] IEPM Group at SLAC: http://www-iepm.slac.stanford.edu/

[13] Abilene Network Engineering team: http://www.abilene.iu.edu/index.cgi?page=engineering

[14] Netlogger: A Methodology for Monitoring and Analysis of Distributed Systems, National Energy Research Scientific Computing Center, Computing Sciences Division, Lawrence Berkeley National Laboratory.
http://www-didc.lbl.gov/NetLogger/

[15] NPACI Rocks Clusters homepage http://slic01.sdsc.edu/

[16] Community Authorization Service (CAS): http://www.globus.org/security/CAS/

[17] Workshop to develop an ATLAS – GriPhyN Testbed, Indiana University, June 2000:
http://lexus.physics.indiana.edu/~rwg/griphyn/june00_workshop.html

[18] U.S. ATLAS Testbed workshop, University of Michigan, June 2001: See links from
http://www.usatlas.bnl.gov/computing/grid/

[19] GRIPE: Grid Registration Infrastructure for Physics Experiments:
http://iuATLAS.physics.indiana.edu/griphyn/GRIPE.jsp

[20] GriPhyN Outreach Center: http://www.aei-potsdam.mpg.de/~manuela/GridWeb/main.html