The Pegasus Portal: Web Based Grid Computing

Gurmeet Singh, Ewa Deelman, Gaurang Mehta, Karan Vahi, Mei-Hui Su
Information Sciences Institute, University of Southern California
G. Bruce Berriman, John Good
Infrared Processing and Analysis Center, California Institute of Technology
Joseph C. Jacob, Daniel S. Katz
Jet Propulsion Laboratory, California Institute of Technology
Albert Lazzarini, Kent Blackburn
California Institute of Technology
Scott Koranda
University of Wisconsin-Milwaukee

ABSTRACT
Pegasus is a planning framework for mapping abstract workflows for execution on the Grid. This paper presents the implementation of a web-based portal for submitting workflows to the Grid using Pegasus. The portal also includes components for generating abstract workflows based on a metadata description of the desired data products and application-specific services. We describe our experiences in using this portal for two Grid applications. A major contribution of our work is in introducing several components that can be useful for Grid portals and hence should be included in Grid portal development toolkits.

Keywords
Grid computing, Workflow management, Web based computing, Scheduling, Resource allocation, Portals.

1. INTRODUCTION
Computational Grids provide access to software and hardware resources geographically distributed and maintained by different institutions. Large-scale collaborations [11] enable scientists to share their resources, data and software using grid technologies to pursue common goals. Such collaborations are visible in gravitational-wave science [1], astronomy [10] and in other fields. The computations and analysis being done on the Grid by these scientists consist of a set of tasks with data and/or control dependency between them [4, 10]. This set of tasks is defined as the application workflow [8].

The workflows can be generated manually by the scientists. However, the workflows can be very large, consisting of hundreds or thousands of nodes, making it impossible to generate them manually. Tools such as Chimera [12] can be used to automate the process of workflow generation. Chimera creates a workflow for generating a data product by querying the virtual data catalog [12]. The workflow generated by Chimera is abstract since it does not specify the grid resources needed to execute the tasks of the workflow. Pegasus is a planning framework [8] that maps abstract workflows for execution on the Grid. It transforms the abstract workflow into a concrete workflow by associating an execute resource with each task in the workflow and adding tasks for transferring input data and output data. The concrete workflow generated by Pegasus can be executed using Condor-G and DAGMan [13].

Chimera, Pegasus, Condor-G and DAGMan are useful components for generating, scheduling and executing complex application workflows. Still, it can be very difficult for the user to install these components and their pre-requisite software, be familiar with their usage, generate workflows, submit the workflow and monitor its execution. The Pegasus Portal is a web-based interface that simplifies these tasks. It is a generic portal where any user with the appropriate credentials can generate and submit workflows for the applications supported by the portal. The portal can be extended to include new applications for which an abstract workflow generator or a Chimera Virtual Data Language Generator (VDL) is available.

Up to date, the portal has been used with the Montage [4] and the LIGO [1] applications. In this paper, we present the design and usage of the Pegasus Portal and the technologies required to implement it. We discuss our experiences in using the portal for the above-mentioned applications and provide recommendations for the Grid portal toolkits.

2. RELATED WORK

There have been several efforts in the Grid community to build web-based portals. BioSimGrid [3] has developed a web-based portal that provides access to simulation datasets. The BioSimGrid portal is not a computational portal but provides distributed query facility using OGSA-DAI. The user authentication and authorization information is stored in an accounts database. The Pegasus Portal on the other hand is a computational portal and does not require any account set up for accessing the portal. WebCom [14] is a computational grid portal. It is a web-based portal that provides access to the computational grid testbed of Xian Jiaotong University. In order to be able to use the portal, the administrator needs to add the user to the access control database. In addition, the user submits jobs in the form of program files. The Pegasus Portal does not require that the Grid resources to be accessed be predefined. Instead, the user has the flexibility to specify the resources to be used. The job submission in the Pegasus Portal is in the form of metadata attributes instead of program files.

The NPACI Hotpage portal [21] provides a generic web-based access to the NPACI resources. It provides a set of interactive services including command execution, account management and file transfer etc. The Pegasus Portal is specialized for executing applications on Grid resources and does not provide generic interactive access to these resources. The user can however find the status of the resources by authenticating with them using his Grid credentials. The ASC portal [18] is a web-based portal that provides an interface for accessing the functionality of the Cactus computational toolkit. The authentication mechanism is similar to the Pegasus Portal. The portal queries a centralized Grid Information Index Server (GIIS) [6] for resource discovery. Additional resources are made available by having the Grid Resource Information Servers (GRIS) [6] responsible for those resources report to the centralized GIIS. The Pegasus Portal has a more flexible mechanism where the users can specify the GIIS servers to be queried for resource discovery.

The GridPort [20] and the Grid Portal Development toolkit (GPDK) [16] provide APIs for developing Grid portals. However, they do not provide a functionality for managing complex workflows. We have used the basic architecture of GPDK for implementing the Pegasus Portal. Both the GridPort and GPDK provide Globus Resource Allocation Manger (GRAM) [7] interfaces for resource management and job submission. However, the GRAM interface can only be used for submitting a single task at a time. This is insufficient for submitting and managing large workflows.

3. COMPONENTS AND SERVICES USED IN THE PEGASUS GRID PORTAL

The portal uses a set of components and services in order to generate and execute application workflows. The components used by the portal are described below

3.1 MyProxy

The portal uses MyProxy [17] for obtaining user credentials. These credentials are in the form of a X509 proxy certificate that is used for authentication with the Grid resources, accessing the Replica Location Service [5] and executing workflows using Condor-G and DAGMan [13]. MyProxy is an online credential repository for the grid. Users can store their Grid credentials in a MyProxy server using a username and passphrase. Any entity can retrieve a limited time proxy credential from the MyProxy server using that username and passphrase.

3.2 Chimera

Chimera [12] is a virtual data system that combines a virtual data catalog (VDC) with a virtual data language (VDL) interpreter. The virtual data catalog is used to store partial workflows specified using the virtual data language. Chimera can use these partial descriptions to generate an abstract workflow for producing a particular data item.

3.3 Pegasus

Pegasus [8] is a planning framework for mapping abstract workflows for execution on the Grid. It can be configured to use different resource discovery mechanisms and perform workflow reduction based on available intermediate data products. Pegasus does resource allocation for the tasks in the abstract workflow. The concrete workflow generated by Pegasus also includes tasks for transferring input and output data and for data registration. The registration tasks are added to register the intermediate and final data products in the Replica Location Service (RLS) [5]. Pegasus can reduce the abstract workflow based on data items that are already generated and registered in the RLS.

3.4 Replica Location Service

The Replica Location Service (RLS) [5] maintains and provides access to mappings between logical names of data items and their target names. The target names may specify the physical location from where the data items can be retrieved or they can refer to another level of logical naming for the data item. The target name can be a FTP, HTTP, file or GridFTP [20] URL. RLS uses Grid Security Infrastructure (GSI) for authentication and authorization.

3.5 Metadata Catalog Service

The Metadata Catalog Service (MCS) [19] maintains and provides access to mappings between logical identifiers of data objects and their metadata attributes. MCS can be used to store metadata attributes of data items and to query for data items based on a set of specified metadata attributes. It provides mechanisms for aggregation of items. MCS can be used along with RLS for data publication and discovery.

3.6 Condor-G and DAGMan

Condor-G [13] is a computation management system that can be used for job submission across administrative domains. It can be used along with Condor DAGMan for workflow submission and management. Condor-G can submit jobs specified as Condor submit files to GRAM gatekeepers [7] running on the grid resources. Condor DAGMan can be used to manage a set of jobs specified as a directed acyclic graph. It releases jobs for execution only when the parent jobs have successfully completed.

4. APPLICATION WORKFLOWS

4.1 Creation and Mapping

The process of generating an application workflow can be divided into two parts

- generating an abstract workflow and
- generating a concrete workflow from the abstract one.

The abstract workflow is a directed acyclic graph where the vertices in the graph represent the compute tasks and the edges represent data dependency between tasks. Figure 1 and 2 show examples of the Montage and the LIGO abstract workflows.

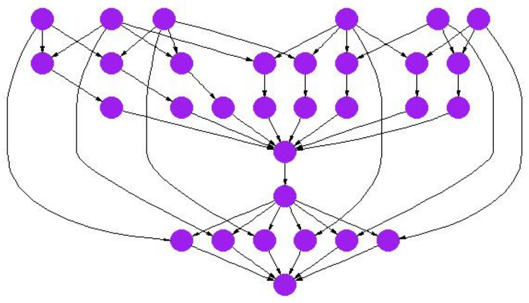


Figure 1. A Montage abstract workflow

Each task in the workflow is a transformation that receives some input data and produces some output data. The task is completely specified by the transformation name, its arguments and the logical names of the input and output data files.

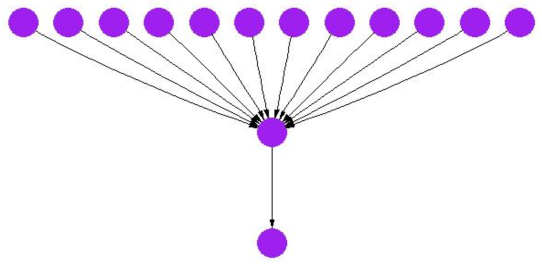


Figure 2. A LIGO abstract workflow

The process of creating an abstract workflow is highly dependent on the application and the purpose of the workflow. This process can only be designed by application experts. Chimera [12] can be used to automate the process but still the virtual data catalog in Chimera has to be populated by the expert. We have used Chimera for generating the abstract workflow in case of LIGO. In the case of Montage, the application experts have implemented a web service that can produce the abstract workflow. The web service is invoked by the portal with the sky location of the mosaic to be created and the size of the mosaic [4]. The service returns a zipped file that contains the abstract workflow and other files required for the execution of the workflow.

Pegasus can be used to map the abstract workflow onto Grid resources thereby creating a concrete workflow. This step is relatively independent of the application but the Transformation Catalog [9] used by Pegasus should specify the location of the application executables on the Grid resources

4.2 Execution

The concrete workflow generated by Pegasus is also a directed acyclic graph where the vertices are the compute or data transfer tasks and the edges represent control dependency between the tasks. Pegasus creates a Condor submit file for each task in the concrete workflow and a main file which lists the dependencies between the tasks. Condor DAGMan is used to execute this concrete workflow. Condor-G is used for submitting individual tasks to the GRAM gatekeepers on the remote Grid resources. A Grid resource can be a single machine or a pool of machines having a local scheduler such as Condor or PBS.

5. THE PEGASUS GRID PORTAL

The Pegasus Portal provides an HTTP(S)-based interface that can be accessed using a standard web browser. The portal architecture is composed of three layers as shown in Figure 3. The top layer consists of the user machines and web browsers.

The second layer consists of the web application server hosting the portal. The server is multithreaded and can handle multiple user requests at the same time. The portal content is dynamic and is generated using java server pages and java servlets. The web server uses a shared file system for storing user profiles and configuration information. Thus, multiple portal instances can be set up on different servers that provide the same view of resources and jobs to the users, thereby increasing availability. The third layer consists of the grid components and services used by the portal described in Section 3. Next, we describe how these components and services are used by the Portal.

5.1 Authentication

In order to use the Pegasus Grid Portal the user needs to have a valid Grid credential in a MyProxy server. The username and the passphrase used for storing the credential allow access to the portal. No other account setup procedure is required to use the portal. The portal uses Java CoG Kit to retrieve a proxy for the user from the MyProxy server. This proxy is used by the portal for authenticating with the GRAM gatekeepers on the grid resources, to access the RLS and to execute the workflow using Condor-G. The lifetime of the delegated proxy is 2 days that is sufficient for many workflows.

5.2 Configuration

Once logged in, the user can create his/her profile by uploading various configuration files and updating email address etc. The configuration specified below can be done by a system administrator for the Virtual Organization (VO) [11] that is using the portal. The configuration files can be shared by all the VO members.

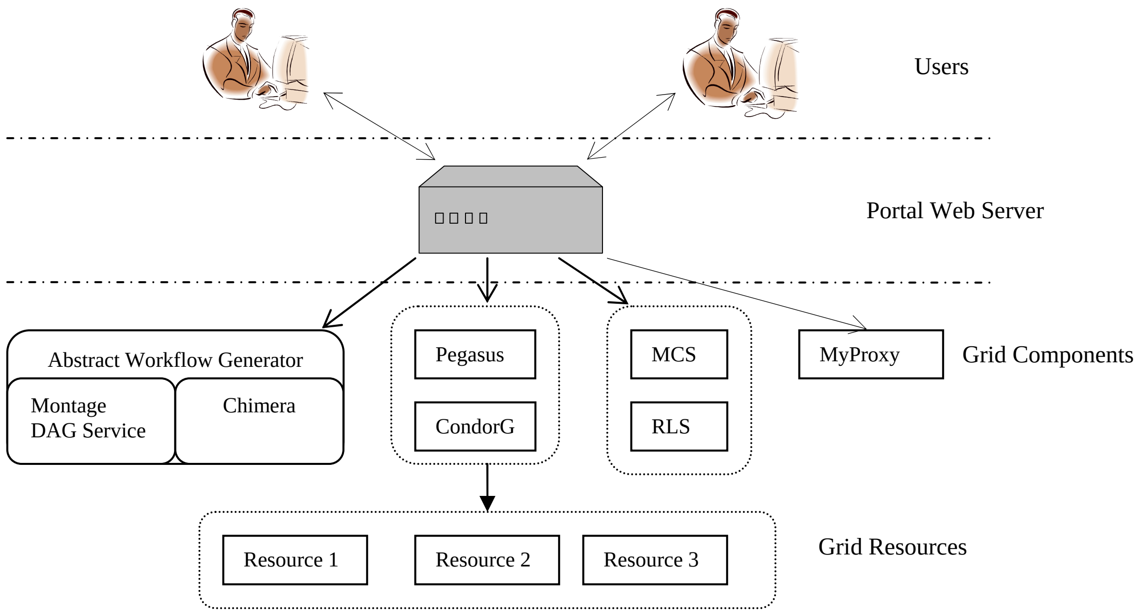


Figure 3. The Portal architecture

The following configuration files are required: i) a resource configuration file called pool.config, ii) a Transformation Catalog that contains information about application executable, iii) a properties file that modifies the behavior of Pegasus.

The portal does not provide access to a predetermined set of resources. Instead, the user can specify the resources to be used. The resource configuration file provides information about the GRAM gatekeepers [7] and the GridFTP servers [2] running on the resources. This resource configuration file is called pool.config and has a syntax as specified in Figure 4. It is used by Pegasus for generating the Condor submit files. Each pool element identifies a Grid resource. The *gridftp* and *jobmanager* elements provide the location of the GridFTP server and the GRAM gatekeeper on the resource. Instead of the user specifying the resources, the Pegasus portal can also query the Globus Monitoring and Discovery Service (MDS) [6] to find the available resources. The resource discovery mechanism to be used can be specified in the properties file (a combination of hand-encoded and MDS information is also feasible).

The abstract workflow only specifies the logical names of transformation. These need to be resolved against the physical location of these transformations on the grid resources. This is done using the Transformation Catalog [9]. Each line of this file consists of <resource name, logical transformation name, physical path of the transformation, environment variables required by the transformation>:

```
grid_rsc1  mProject  /home/nfs/foo/bin/mProject
                    MONTAGE_BIN=/bin
```

Pegasus can also use a database-based implementation of the Transformation Catalog. The particular Transformation Catalog implementation is to be used can be specified in the properties file.

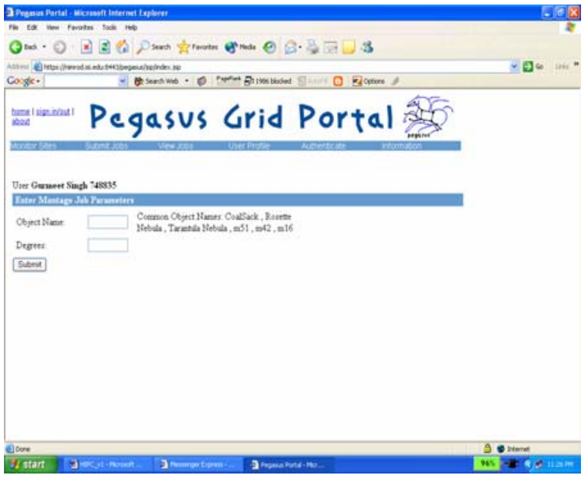
The VO system administrator can setup a RLS server for registering the input and output data products. The location of this RLS server can be included in the user profile. The profile also allows the user to specify the email address to be used for notification purposes.

```
    <pool handle="grid_rsc1" gridlaunch="/home/nfs/foo/bin/kickstart">
    <lrc url="rls://rsc.domain.edu" />
    <gridftp url="gsiftp://rsc.domain.edu/home/nfs/foo/storage_dir" storage="/home/nfs/foo/storage_dir"
major="2" minor="4" patch="3" />
    <jobmanager universe="vanilla" url="rsc.domain.edu/jobmanager-con" major="2" minor="4"
patch="3" />
    <jobmanager universe="transfer" url="rsc.domain.edu/jobmanager-util" major="2" minor="4"
    patch="3" />
    <workdirectory>/home/nfs/foo/exec_dir</workdirectory>
```

Figure 4 Pool Configuration

5.3 Workflow Submission

The main goal of the Pegasus Grid Portal is to simplify the process of workflow submission and execution. For the applications supported by the portal (LIGO and Montage), the user specifies the metadata parameters of the desired data to be generated or analysis to be done. For the LIGO application [1], the user specifies the sky location, the center and the frequency band for which the gravitational wave is to be searched. For the Montage application [4], the user specifies the sky location and the size of the desired mosaic (further described in section 6). There is no generic schema for the metadata parameters as they are specific to the application. The user interface for specifying these parameters has to be developed in consultation with the application experts. Figure 5 shows the user interface for specifying the metadata parameters for Montage.



User Gurmeet Singh 748835

Enter Montage Job Parameters

Object Name: Common Object Names: CoalSack , Rosette Nebula , Tarantula Nebula , m51 , m42 , m16

Degrees:

Figure 5 Montage workflow submission page

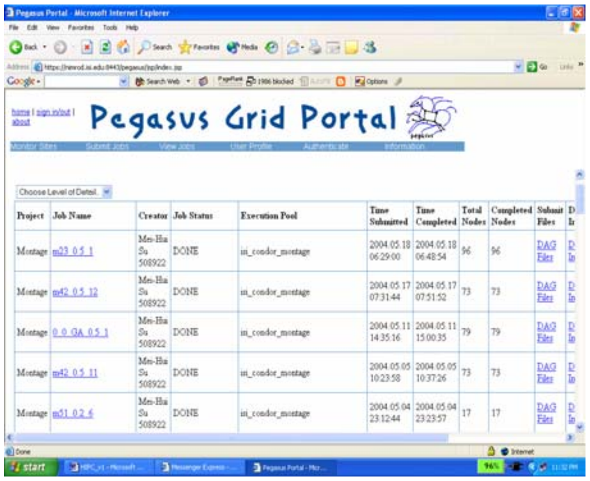
The users of these applications are most familiar with these metadata parameters and their significance. Once the parameters are specified, the abstract workflow is generated using an application specific interface. For the LIGO application, a Perl script was used to generate a VDL file that was stored into a Virtual Data Catalog. Then Chimera was used to generate the abstract workflow. For the Montage application, a web service is used to generate the abstract workflow using the specified parameters. The web service has been implemented by the Montage project team at IPAC and JPL. In general the steps required for submitting a workflow are:

1. Specify the metadata parameters for the desired data product.
2. The portal queries the Metadata Catalog Service to find already instantiated data products that have the specified attributes or are close enough.
3. The list of products found in step 2 is shown to the user. The user can use these instantiated data products or has the option to go ahead and create a new workflow.
4. The abstract workflow is generated using the application specific interface.
5. In the case of Montage, the image files required to execute the workflow are downloaded using the 2MASS image service and registered in the RLS. For LIGO all the required files are already registered in the RLS.
6. Pegasus does the mapping of tasks in the workflow to resources specified in the resource configuration. The user can specify a particular set of resources to be considered for the mapping. Pegasus can be configured to use a particular resource selection strategy. In the portal we just use a random strategy for allocating tasks to resources.
7. The concrete workflow is submitted to Condor DAGMan for execution. A logical identifier is created for the workflow from the parameters specified during its creation and it is registered in MCS. The metadata parameters of the workflow are also stored as user defined attributes in MCS. The portal also adds attributes specifying the workflow submission time, number of tasks in the workflow, status, the application it belongs to, the location of log files, user's email address, etc.

The submitted workflow may take a long time to complete. The user may logout from the portal and login later to check its status.

5.4 Workflow Monitoring and Notification

The portal allows users to view the status of the workflow (submitted, active, done, failed), the number of tasks already completed, the tasks currently executing, and other information (Figure 6). DAGMan creates a log file recording the execution history. The portal does job monitoring by periodically querying the MCS for the workflows that are currently active. It then finds the location of the log file for each workflow and parses it. If there is any change in the status (e.g. number of tasks executed, etc), it is updated in the MCS.

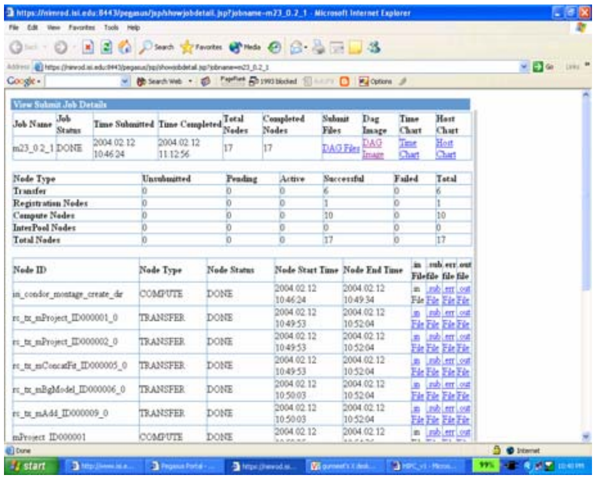


Project	Job Name	Creator	Job Status	Execution Pool	Time Submitted	Time Completed	Total Nodes	Completed Nodes	Submit Files	D In
Montage	m23_0.5_1	Mei-Hui Su 508922	DONE	isi_condor_montage	2004.05.18 06:29:00	2004.05.18 06:48:54	96	96	DAG Files	D In
Montage	m42_0.5_12	Mei-Hui Su 508922	DONE	isi_condor_montage	2004.05.17 07:31:44	2004.05.17 07:51:52	73	73	DAG Files	D In
Montage	0_0_GA_0.5_1	Mei-Hui Su 508922	DONE	isi_condor_montage	2004.05.11 14:35:16	2004.05.11 15:00:35	79	79	DAG Files	D In
Montage	m42_0.5_11	Mei-Hui Su 508922	DONE	isi_condor_montage	2004.05.05 10:23:58	2004.05.05 10:37:26	73	73	DAG Files	D In
Montage	m51_0.2_6	Mei-Hui Su 508922	DONE	isi_condor_montage	2004.05.04 23:12:44	2004.05.04 23:23:57	17	17	DAG Files	D In

Figure 6 Workflow status page

The workflow status page, shown in Figure 6, is dynamically generated by querying the MCS.

Users can also see the detailed status of a particular workflow that includes the status of all the tasks in the workflow (Figure 7) and their standard and error outputs. This is especially helpful in failure cases since the user can find out which task has failed and the output produced by the task.



View Submit Job Details

Job Name	Job Status	Time Submitted	Time Completed	Total Nodes	Completed Nodes	Submit Files	Dag Image	Time Chart	Host Chart
m23_0.2_1	DONE	2004.02.12 10:46:24	2004.02.12 11:12:56	17	17	DAG Files	DAG Image	Time Chart	Host Chart

Node Type	Unsubmitted	Pending	Active	Successful	Failed	Total
Transfer	0	0	0	6	0	6
Registration Nodes	0	0	0	1	0	1
Compute Nodes	0	0	0	10	0	10
InterPool Nodes	0	0	0	0	0	0
Total Nodes	0	0	0	17	0	17

Node ID	Node Type	Node Status	Node Start Time	Node End Time	in File	sub file	err file	out file
in_condor_montage_create_dir	COMPUTE	DONE	2004.02.12 10:46:24	2004.02.12 10:49:34	in File	sub File	err File	out File
rc_tx_mProject_ID000001_0	TRANSFER	DONE	2004.02.12 10:49:53	2004.02.12 10:52:04	in File	sub File	err File	out File
rc_tx_mProject_ID000002_0	TRANSFER	DONE	2004.02.12 10:49:53	2004.02.12 10:52:04	in File	sub File	err File	out File
rc_tx_mConcatFit_ID000005_0	TRANSFER	DONE	2004.02.12 10:49:53	2004.02.12 10:52:04	in File	sub File	err File	out File
rc_tx_mBgModel_ID000006_0	TRANSFER	DONE	2004.02.12 10:50:03	2004.02.12 10:52:04	in File	sub File	err File	out File
rc_tx_mAdd_ID000009_0	TRANSFER	DONE	2004.02.12 10:50:03	2004.02.12 10:52:04	in File	sub File	err File	out File
mProject_ID000001	COMPUTE	DONE	2004.02.12	2004.02.12	in	sub	err	out

Figure 7 details of a particular workflow

Upon completion of the workflow (either successfully or otherwise), an email notification is send to the user. The email address from the user profile at the time of workflow submission is used for notification. The final post-processing step for LIGO (the bottom node in Figure 2) stores the search results into a database. When the Montage workflow completes, the portal transfers the final generated mosaic to the portal web server and makes it available via HTTP. Thus, the Montage users can download the existing mosaics without having to generate them again.

Currently the status of a workflow submitted using the portal is visible to all the portal's users. However the status information can only be deleted by the administrator of the portal.

6. EXPERIENCES

We have used the Pegasus portal for two applications, the Laser Interferometer Gravitational Wave Observatory [1] and Montage [4].

LIGO is a Laser Interferometer Gravitational Wave Observatory project. The portal is used to search for pulsars at random points in the sky near the galactic core and for a range of frequencies between 150-350 Hz.

Montage is an application for generating custom astronomical image mosaics on demand. The user specifies the mosaic to be constructed in terms of its location on the sky and its size.

We have performed approximately 3000 searches for the 5x5 celestial grid in case of LIGO and built approximately 50 mosaics for Montage using the portal. The total numbers of tasks in the workflows were 94008. Out of these 20146 were data transfer tasks, 16152 were registration tasks and the rest were compute tasks. The portal has also proved to be a very useful information repository for gathering statistics, doing diagnostics and building application profiles. The portal has also been used for educational and demonstration purposes.

The components used by the portal (mentioned in section 3) are typically used from the command line. No programmatic access to these components is available except for MyProxy, RLS and MCS. We had faced several issues with making Condor-G and DAGMan work with different user proxies at the same time from the portal web server. We were able to resolve these issues with help from the Condor team. Still forking processes to invoke these components from the portal server is inefficient and we are hoping that a Java API interface will be available in new versions of these components.

7. CONCLUSION AND FUTURE DIRECTIONS

The Pegasus Portal provides a very simplified interface for workflow submission and management. It is not limited to a particular application. Any user having valid grid credentials can use the portal. We have found the portal very useful for the LIGO and the Montage applications.

The Pegasus Grid Portal is very useful in scenarios where a virtual organization (VO) [11] wants to provide easy to use application submission interface to its members. The VO system administrators can do the configuration mentioned in Section 5 and the metadata-based interface can be used by VO members for submitting workflows. Application experts can develop application-specific abstract workflow generators. Users are shielded from the complexity of installing and using the various components in order to access the Grid resources.

To the best of our knowledge, no Grid portal or toolkit provides such generic workflow management capabilities. As Grid applications become complex, it becomes important that the Grid portals be able to support submission and management of more complex workflows. Our main contribution is in providing

- A generic web based interface that does not require any special client side software.
- A simplified workflow submission interface. We anticipate that various other interfaces are also possible
- A planning framework (Pegasus) that can map workflows to Grid resources.
- A generic workflow management and notification interface

We are interested in the integration of such capabilities in Grid portal toolkits such as GridPort [20]. The next generation toolkits should provide components such as Condor DAGMan that can be used for workflow submission and management. The Globus GRAM interfaces in the toolkits should be extended to include support for Condor-G. The process of abstract workflow generation will remain to be application specific but Pegasus can be used as a planning component for mapping the abstract workflow to Grid resources. We plan to build the next version of the Pegasus portal using a portal development toolkit that provides generic workflow management capabilities.

8. ACKNOWLEDGMENTS

Pegasus is supported by NSF under grants ITR-0086044 (GriPhyN) and ITR AST0122449 (NVO). LIGO Laboratory operates under NSF cooperative agreement PHY-0107417. Montage is supported by the NASA Earth Sciences Technology Office Computing Technologies (ESTO-CT) program, under Cooperative Agreement Notice NCC 5-6261.

9. REFERENCES

[1] Abramovici, A., Althouse, W.E. and et al. *LIGO: The Laser Interferometer Gravitational-Wave Observatory*, Science, vol. 256, pp. 325-333, 1992

[2] Allocock, W., Bester, J., et al. *Secure, Efficient Data Transport and Replica Management for High-Performance Data-Intensive Computing*, Proceedings of the IEEE Mass Storage Conference, pp. 13-28 April 2001

[3] Bing Wu; Dovey, M.; Muan Hong Ng; Kaihsu Tai; Murdock, S.; Jeffreys, P.; Cox, S.; Essex, J.; Sansom, M.S.P. *A web /grid portal implementation of biosimgrid: a biomolecular simulation database*, Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on, Vol.2, Iss., April 5-7, 2004 Pages: 50- 54

[4] Berriman, G. B. et al, *Montage A Grid Enabled Image Mosaic Service for the National Virtual Observatory* , ADASS XIII, ASP Conference Series Vol XXX, F Oshsenbein M Allen and D Egret eds, 2003

[5] Chervenak, A., Deelman, E., Foster, I., Guy, L., Hoschek, W., et al. *Giggle: A Framework for Constructing Scalable Replica Location Services*, Proceedings of SuperComputing 2002 (SC2002), November 2002

[6] Czajkowski, K., Fitzgerald, S., et al. *Grid Information Services for Distributed Resource Sharing* , Proc. of the 10th IEEE International Symposium on High Performance Distributed Computing. Pages 181-194, 2001

[7] Czajkowski, K., et al. *A Resource Management Architecture for Metacomputing Systems*, Proc. IPPS/SPDP'98 Workshop on Job Scheduling Strategies for Parallel Processing, 1998

[8] Deelman, E. et al. "Pegasus : Mapping Scientific Workflows onto the Grid", Across Grids Conference 2004, Nicosia, Cyprus, 2004

[9] Deelman, E., Kesselman, C., et al. *Transformation Catalog Design for GriPhyN*, Technical Report GriPhyN-2001-17, 2001

[10] Deelman, E. et al. "Grid-Based Galaxy Morphology Analysis for the National Virtual Observatory", Proceedings of the SuperComputing Conference 2003, Phoenix, Arizona

[11] Foster, I., Kesselman, C., and Tuecke, S., "The Anatomy of the Grid: Enabling Scalable Virtual Organizations" International Journal of High Perfomance Computing Applications, vol. 15, pp. 200-222, 2001

[12] Foster, I., Voeckler, J., Wilde, M., Zhao, Y *Chimera: A Virtual Data System for Representing, Querying, and Automating Data Derivation*, presented at Scientific and Statistical Database Management, 2002.

[13] Frey, J.; Tannenbaum, T.; Livny, M.; Foster, I.; Tuecke, S. *Condor-G: a computation management agent for multi-institutional grids* High Performance Distributed Computing, 2001. Proceedings. 10th IEEE International Symposium on, Vol., Iss., 2001 Pages:55-63

[14] Ge He; Zhiwei Xu *Design and implementation of a Web-based computational grid portal* Web Intelligence, 2003. WI 2003. Proceedings. IEEE/WIC International Conference on, Vol., Iss., 13-17 Oct. 2003 Pages: 478- 481

[15] Laszewski, G., Foster, I., Gawor, J., Lane, P., *A Java Commodity Grid Toolkit*, Concurreny: Practice and Experience, 13, 2001

[16] Novotny, Jason. *The Grid Portal Development Kit* National Laboratory for Applied Network Research (NLANR) http://doesciencegrid.org/projects/GPDK

[17] Novotny, J.; Tuecke, S.; Welch, V. *An online credential repository for the Grid: MyProxy* , High Performance Distributed Computing. Proceedings. 10th IEEE International Symposium on, Vol., Iss., 2001 Pages:104-111

[18] Russell, M.; Allen, G.; Daues, G.; Foster, I.; Seidel, E.; Novotny, J.; Shalf, J.; von Laszewski, G., *The Astrophysics Simulation Collaboratory: a science portal enabling community software development*, High Performance Distributed Computing, 2001. Proceedings. 10th IEEE International Symposium on, Vol., Iss., 2001, Pages:207-215.

[19] Singh, G., et al, *A Metadata Catalog Service for Data Intensive Applications* Proceedings of Supercomputing(SC), 2003

[20] Thomas, M.; Mock, S.; Dahan, M.; Mueller, K.; Sutton, D.; Boisseau, J.R. " The GridPort toolkit: a system for building Grid portals", High Performance Distributed Computing, 2001. Proceedings. 10th IEEE International Symposium on, Vol., Iss., 2001 Pages:216-227

[21] Thomas, M.; Mock, S.; Boisseau, J. "Development of Web toolkits for computational science portals: the NPACI HotPage", High-Performance Distributed Computing, 2000. Proceedings. The Ninth International Symposium on, Vol., Iss., 2000 Pages:308-309