The Collaborative P-GRADE Grid Portal

Gareth J. Lewis¹, Gergely Sipos², Florian Urmetzer¹,
Vassil N. Alexandrov¹, and Peter Kacsuk²

¹ Advanced Computing and Emerging Technologies Centre,
School of Systems Engineering, University of Reading,
WhiteKnights, P.O. Box 225, Reading, RG6 6AY, United Kingdom
`{g.j.lewis, f.urmetzer, v.n.alexandrov}@rdg.ac.uk`
<http://www.acet.rdg.ac.uk>

² MTA SZTAKI, Laboratory of Parallel and Distributed Systems,
H-1518 Budapest, Hungary
`{sipos, kacsuk}@sztaki.hu`
<http://www.lpds.sztaki.hu>

Abstract. Grid portals are increasingly used to provide uniform access to the grid infrastructure. This paper describes how the P-GRADE Grid Portal could be used in a collaborative manner to facilitate group work and support the notion of Virtual Organisations. We describe the development issues involved in the construction of a collaborative portal, including ensuring a consistent view between participants of a collaborative workflow and management of proxy credentials to allow separate nodes of the workflow to be submitted to different grids.

1 Introduction

The Grid Infrastructure is essential in supporting the development of Virtual Organisations. The Grid enables the sharing of resources dynamically and in a secure manner. Grid Portals are increasingly used as a convenient interface to the Grid by providing uniform access to grid resources.

P-GRADE [1] is a graphical programming environment, used in the development of parallel applications. P-GRADE incorporates GRAPNEL (Graphical Process Net Language), which is a graphical parallel programming language and GRED (Graphical Editor), which can be used to write parallel applications. Along with several other components, P-GRADE provides an abstraction from the low level details associated with the message passing. P-Grade has several layers which are used in the development and execution of parallel programs. The layer of particular interest, for the purposes of this paper, is the Workflow layer. A Workflow is a set of consecutive and parallel jobs, which are cooperating in the execution of a parallel program [2]. Different nodes of a certain workflow can be executed on different resources within a grid. The Workflow layer allows users to design and execute workflows, specifying properties of individual nodes, and specifying whether the output data of one node is used as the input of another node in the workflow.

The P-GRADE Grid Portal [2] provides uniform access to underlying grid resources. It has two main components, the Portal Server, and the Workflow Editor (A Java Webstart application). The Portal Server was developed using Gridsphere, a grid portal development framework. It is responsible for managing several aspects, such as, security, monitoring and execution visualisation. The Workflow editor allows a user to design a workflow and provides graphical feedback on its progress during the execution.

The objective of the project outlined in this paper, was the development of a collaborative P-GRADE Portal, which could be used by several participants in "real-time". The motivation for this collaborative ability was two-fold; firstly, it would allow different users to collaboratively construct workflows and secondly, participants could use their personal certificates to submit nodes of the workflow to different grids.

2 The P-GRADE Portal

As previously mentioned, the P-GRADE Portal consists of two major components, the Portal Server and the Workflow editor. Together these components facilitate the development of workflows and their execution within a grid. The Workflow Editor allows the end-user to construct workflow graphs whilst, the Portal Server is responsible for managing the security as well as the visualisation and monitoring of the execution. The Workflow Editor is separate from the Portal and runs on the end users local machine. It communicates with the Portal Server to provide information about the current workflow. The user of the workflow editor can create a workflow graph consisting multiple jobs, which are either sequential, PVM or MPI programs.

The Grid Security Infrastructure (GSI) [4] (based on X.509 certificates) is used to provide secure authentication and access to the grid resources. Portals are becoming popular for providing a convenient interface to these computational grids. The grid portals rely on the MyProxy [5] repository to enable the integration of the portal with the GSI. The MyProxy server allows the grid portal to use the GSI to interact with the grid resources. The P-GRADE portal is responsible for managing the downloading of proxy credentials from the MyProxy Server, and allowing users to view the lifetime of the proxies. Several different proxies can be downloaded but only a single proxy can be active, as each workflow utilises the resources from a single grid.

The Portal must be able to monitor the execution of the workflow and provide visual feedback to the end-user. The P-GRADE portal allows the user to monitor the workflow in real-time, using the Mercury monitoring tool [2]. It represents the progress of the execution, both in the workflow editor and online within the portal. The user can monitor the entire workflow and the individual jobs. The portal shows the communication between the workflow jobs and the processes within each job.

3 Collaborative Version of the P-GRADE Portal

A driving force in the development of the Grid Infrastructure is the collaboration of distributed partners (forming Virtual Organisations) interacting over the grid. The collaborative P-GRADE portal supports the idea of distributed groups working toward a common goal using the underlying grid infrastructure. The main motivations for the development of a collaborative P-Grade Portal, were to allow workflows to be constructed collaboratively, and facilitate submission of different parts of the workflow to several grids. There are two approaches to implementing a collaborative application such as the workflow editor. The first involves sharing of the application in a transparent manner, and does not require it to be adapted to be aware of the collaboration. Collaborative-unaware sharing would be achieved by sharing a visual representation of the same workflow editor, without sharing any application specific data structures. The second approach depends upon the application being aware of its collaborative functionality. In the case of the P-GRADE Portal it is sensible to adapt the workflow editor and the portal server to work in a collaborative manner. The portal server provides a central point to which each workflow editor can connect and be synchronised. The server also provides the solution to transferring data between different grids (as described in the following sections).

The collaborative version of the workflow editor must allow workflows to be constructed in a collaborative manner. The end-users must be able to add and remove nodes of the workflow with these changes being visible to each participant. For successful collaboration, between the users cooperating in the construction of a collaborative workflow, the portal must ensure that each participant has the same view. Simultaneous editing of the workflow could lead to inconsistencies, where participants have a different view of the same data. The client/server network topology assists the implementation of floor control. A single copy of the workflow data can be contained at the portal server and its consistency assured by guaranteeing that access to the central workflow data is mutually exclusive. The Workflow editor must also provide a visual representation of the workflows execution - currently this functionality is provided by representing different states of execution in different colours. The execution functionality of the Workflow editor must evolve to work within a collaborative environment.

The Portal Server must also be adapted to include the necessary collaborative functionality. An important issue in the development of the collaborative P-GRADE portal, is management of the proxy credentials, which are downloaded from the MyProxy server. The current P-GRADE Portal allows different proxy credentials to be downloaded, but as a workflow can only be submitted to a single grid, only one of the proxies is active. A major aim of the shared P-GRADE portal was to allow separate parts of collaborative workflows to be submitted to different grids. To facilitate this objective The Credential Manager [2], must be adapted to be able to use multiple proxy credentials concurrently. The visualisation and monitoring of an executing workflow is an important consideration within the collaborative setting. The portal server must ensure that all participants of a collaborative workflow receive the same visual information,

and that the central workflow object is not modified by any workflow editor during the execution period.

3.1 Collaborative P-GRADE Portal Design

This section expands upon the issues described above and gives details on specific design decisions. There are three main stages in the development and execution of parallel applications using the P-GRADE portal. The initial step is the construction of the Workflow graph, this involves constructing how the consecutive and parallel jobs cooperate, and where output from one job can be used as input to another. Before the workflow can be submitted to the grid the portal server must ensure that the necessary proxy credentials are available. This step is managed by the Credential Manager, which downloads the proxy credentials from the MyProxy server. The final stage is the execution of the workflow including its monitoring and visualisation. The current P-GRADE portal will have to be adapted at each of these stages to be utilised successfully within a collaborative setting.

3.2 Collaborative Workflow Construction

The construction of the workflow graph is achieved in the workflow editor which, runs on the users local machine. Much of the development associated with the construction of the workflow graph will be within the workflow editor. However, the portal server is important in this step as it contains the stored workflows and will be responsible for ensuring a consistent view of the workflow between the distributed participants. In the current version of the P-GRADE portal, the workflow editor runs on the users local machine and connects to the central P-GRADE portal server to exchange information related to the workflow graph. This graph is then used by the server to submit the workflow to the grid infrastructure. The workflow editors have no reason to send information between themselves as a workflow is “owned” by a single user. In the collaborative version, the editors working on the same workflow will have to communicate with each other to ensure the workflows consistency. Having the workflow editors connect directly to each other would not be sensible and would lead to problems seen in many peer-to-peer applications. The central portal server allows the editors to indirectly communicate with each other. By containing a central workflow object at the server, and ensuring mutual exclusive access to all or part of this object, we can achieve a consistent view between the distributed editors.

The collaborative P-GRADE portal will be implemented with several workflow objects. The workflow editor will contain workflow objects called Local Workflows Objects (LWO), these will contain the data present within the stored workflows and additional information concerning the locking and un-saved changes. The portal server will have two types of workflow objects, the first will be known as the Dynamic Global Workflow Object (DGWO) and will contain the same information as local workflow objects. The DGWO objects will be updated by each of the workflow editors when they perform a local-to-global update. The LWO for each of the workflow editors will be updated by a

global-to-local update. The second workflow object at the portal server will be the Static Global Workflow Object (SGWO). This object will contain the data stored in the saved collaborative workflow and the locking information, the SGWO will have no data associated with components that are not currently saved (Fig. 1).

The major interfaces for the project lay between these different workflow objects. The first interface is between the SGWO and the DGWO - this interface represents the point between the collaborative workflow object, where the changes to local workflows are stored, and the workflow that mirrors the data contained on disk. There are several functions that must be present in this interface to enable interactions between these two objects. The SGWO contains the workflow data from disk and the locking information for the entire collaborative workflow. The interface must allow the stored collaborative workflows to be retrieved from the portal server, the locking information to be synchronised between the two global workflow objects and the unsaved changes from the DGWO to be incorporated into the SGWO during the saving process. The SGWO can be used specifically by the portal server for submitting the workflow to the grid resources. Another advantage of the SGWO is that updates to the object are more efficient than saving changes to disk when saving or when updating locking information.

The second major interface lays between the portal server and the workflow editors. The workflow editors must be able to communicate with the portal server to maintain the consistency of the collaborative workflow between the participants. There is some additional functionality that must be available to ensure consistency, including: local-to-global updates, global-to-local updates, locking nodes and unlocking nodes. The workflow editor communicates with the portal server via HTTP, and so cannot receive messages asynchronously from the server. The workflow editor must be responsible for polling the server to receive the latest global view. When the workflow editor polls the server it will be involved in the local-to-global synchronisation. The server can respond to this request by issuing a response in the form of a global-to-local update. Obviously the polling interval is crucial, the interval should be small to ensure a more consistent workflow between the participants, however, if the interval is too small, then it could lead to an undue load on the network.

The locking of the workflow is also an important consideration at this interface. The DGWO contains data that is shared between several different editors concurrently. As with all shared data, there is a distinct possibility of race conditions due to concurrent access to the data. To overcome this problem the portal server must ensure that access to the DGWO is mutually exclusive. The portal must provide a locking mechanism by which participants are able to lock a part of the workflow to be modified. In the interface there must be two functions available to the workflow editor. The first is the lock function, which will allow participants to lock a node and associated nodes. If participants had to acquire a lock for the entire workflow, it would place a large constraint on effective collaboration. The alternative is that the participants are able to lock

Workflow
Editors

Portal Server

Fig. 1. Overview of the Collaborative P-GRADE Portal showing workflow objects

single jobs and associated jobs of a workflow. The portal server must manage the locking mechanism to make sure that multiple editors cannot lock the same nodes. The management of the locking will take place at the SGWO level. The unlock function must also be available to the workflow editors. Once a participant has completed the modification to part of the workflow, these nodes should be released. There must also be a set time interval for how long a participant can hold a lock before it is forcibly removed by the system. A single participant must not be able to monopolise a particular part of the workflow and any failing workflow editors should not be able to hold locks indefinitely.

3.3 Execution and Management of a Collaborative Workflow

Once the collaborative workflow has been constructed it is ready to be executed on a grid or over several grids. There are several issues that have to be considered when submitting the collaborative workflow. The aim is to have parts of the collaborative workflow submitted to different grids. To achieve this, the Portal must be responsible for effectively managing the proxy credentials, ensuring locking of the workflow during the execution period and facilitating the transfer of intermediate data between the different grids.

Credential Management. In the current version of the P-GRADE portal, the Credential Manager is responsible for managing the proxy credentials. This involves allowing users to download an appropriate proxy certificate from the MyProxy server, and providing information such as its lifetime. The current version allows multiple proxies to be downloaded, but allows only one of these

to be actively used for submission of the workflow. A non-collaborative workflow is submitted to a single grid and so it is sensible to be able to use only one proxy. An aim of the collaborative portal is to allow parts of a single workflow to be submitted on different grids using different participants proxy credentials. To achieve this aim, the Credential Manager must be adapted to be capable of managing several active proxies for a single workflow.

To manage several active proxies, the portal server must be capable of associating active proxies with different nodes of the workflow graph. The user must be able to specify within the editor which proxy will be used for which node. The current version of the portal, is not capable of automatically downloading proxies - this would be necessary in the collaborative version. The portal must keep track of the relation between the users, the MyProxy server and the available grid resources.

Workflow Locking. Once a collaborative workflow has been submitted by one of the collaborative users, the portal must ensure that all editors, are locked and set to an execution state. This feature is currently available within the portal, but consideration must be given to ensuring that all workflow editors are within this locked state before submission to the grid can proceed.

Transfer of Intermediate Data. Different nodes of the workflow could be submitted to different grids within the proposed collaborative P-GRADE portal. The major issue with this design is that there is generally no facility available for the data to be transferred between the different grids. Thus in the workflow, outputs from some jobs will be needed as inputs to other jobs on different grids. The central portal server can facilitate the indirect transfer of the output data (from one node) for use as input data to another node (submitted on a different grid). The portal server must be able to handle the transfer of this data identifying from which data the node has been received and to which node the data must be delivered. The management of the Proxy Credentials is essential in facilitating the indirect transfer of the intermediate data. Once execution completes, the output data from the collaborative workflow must be accessible to each participant of the collaborative session.

4 Conclusion

In this paper, we have discussed the issues and motivations involved in the development of a collaborative P-GRADE grid portal. The goal of this collaborative portal was to allow distributed participants to work on “collaborative workflows”, run workflows, and monitor the progress of the nodes’ execution across different grids. The paper describes how the workflow editor and portal server must be adapted to be used in a collaborative manner, ensuring a consistent view of the workflow by providing synchronisation and locking mechanisms. Management of the proxy credentials by the portal server has been discussed to allow the automatic download of proxies by monitoring the relationship between users, and available grid resources.

References

1. Peter Kacsuk: Visual Parallel Programming on SGI Machines. The SGI Users' Conference, Krakow, Poland, pp. 37-56, 2000
2. Csaba Nemeth, Gabor Dozsa, Robert Lovas and Peter Kacsuk: The P-GRADE Grid Portal. Computational Science and Its Applications - ICCSA 2004: International Conference Assisi, Italy, LNCS 3044, pp. 10-19
3. Joshy Joseph, and Craig Fellenstein: Grid Computing. On-Demand Series, Prentice Hall, 2004
4. R. Butler, D. Engert, I. Foster, C. Kesselman, S. Tuecke, J. Volmer and V. Welch: Design and Deployment of a National-Scale Authentication Infrastructure. IEEE Computer, 33(12):60-66, 2000
5. J. Novotny, S. Tuecke, and V. Welch: An Online Credential Repository for the Grid: MyProxy. In Symposium on High Performance Distributed Computing, San Francisco, Aug. 2001.
6. J. Chris Lauwers and Keith A. Lantz: Collaboration awareness in support of collaboration transparency: Requirements for the next generation of shared window systems. In Proceedings of ACM CHI90 Conference on Human Factors in Computing Systems, pages 303-311, 1990.
7. G J. Lewis, S. Mehmood Hasan, Vassil N. Alexandrov: Building Collaborative Environments for Advanced Computing. In the proceedings of the 17th International Conference on Parallel and Distributed Systems (ISCA), pp. 497-502, San Francisco, 2004.