Meta Services: Abstract a Workflow in Computational Grid Environments

Sangkeon Lee and Jaeyoung Choi

School of Computing, Soongsil University,
1-1 Sangdo-dong, Dongjak-gu, Seoul 156-743, Korea
sglee@ss.ssu.ac.kr
choi@ssu.ac.kr

Abstract. We define a concept of Meta services which describes a way of abstracting and mapping a workflow to a service in computational Grid environments. By using Meta services, a workflow in a Grid environment could adapt various service concepts such as Grid services, Web services, and portal services without modification to the workflow. And the converted Meta services could be shared and reused by users. Furthermore, historical performance data could be included in the Meta service, making effective execution of the workflow possible.

1 Introduction

Grid computing is a new distributed computing infrastructure which supports the sharing of large-scale resources that are geographically distributed. GRID systems provide higher performance and a wider range of resource sharing than traditional distributed systems and they allow advanced application functionalities [1]. Nowadays, research for applying workflows to a Grid environment is actively being studied.

GridFlow [2] is a workflow management system, which uses agent-based resource management and a local resource scheduling system, Titan. It focuses on the scheduling of time-critical grid applications in a cross-domain and highly dynamic grid environment by using a fuzzy timing technique and performance prediction. MyGrid [3] provides ontology-based services such as resource discovery, workflow enactment, and distributed query processing for integration. It is a middleware research to support biological environments on Grids. The GridLab project [4] is developing application tools and middleware services for Grid environments. Services provided by GridLab include resource brokering, data management, and adaptive services. And Condor [5] provides a workload management system for computation-intensive jobs and a scheduling of dependencies using DAGMan. Pegasus [6] is a workflow mapping and planning system that can map abstract application specific workflows to an executable form for DAGMan. MSF (Meta Scheduling Framework) [7] provides a graphical workflow composition service and workflow management service for computational Grid.

From our experience of constructing a Grid portal with MSF (Meta Scheduling Framework) [7], we need to define a new concept of services to integrate the

V.S. Sunderam et al. (Eds.): ICCS 2005, LNCS 3516, pp. 916–919, 2005.

workflow described with JCML (Job Control Markup Language) to the portal without modification of the workflow. The JCML is a workflow description language used in MSF. We also found that sharing workflows is essential for the Grid portal users. Because most particular Grid portals are designed to support specific scientific research, sharing workflows among the Grid users is a very useful feature. In this paper, we present Meta services in order to map existing Grid workflows to Meta services without modification of description of the workflows. And the Meta services provide service specific information that allows the sharing of workflows among users and resource information for effective scheduling of Grid resources. And historical performance data is required for effective scheduling of Grid resources. However all prior research does not provide such functionalities. Other Grid research such as Smith's work [8], Pegasus, and Gridflow shows that historical performance data can be used for effective scheduling of Grid resources. Therefore we define a concept of Meta services to meet these requirements.

2 Meta Services

Meta services map an existing Grid workflow to a service by overriding attributes of the Grid workflow. Most of workflows are described by XML or they can be converted to XML documents. And Meta services describe how to override attributes in XML documents with service parameter and service specific information. The mapping is shown in Figure 1.

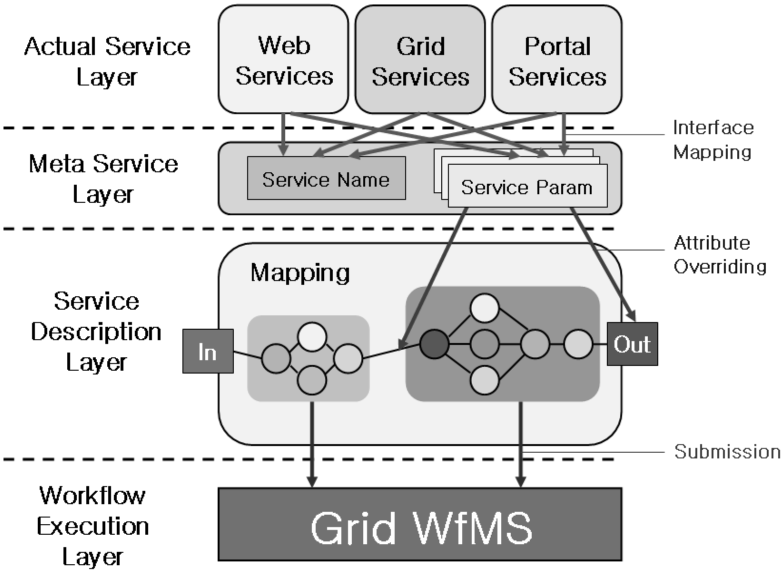

Fig. 1. Meta Services: Mapping workflows to services

Because a Meta service is accessed by it's name and parameters, the interface of the Meta service is easily mapped to the corresponding Web service or Grid service, which contains one operation within one porttype. Moreover it is also easily mapped to a Portal service written in java servlet or CGI. When a Meta service is called, the service name is used as an identifier of the service, the service parameters override attributes of the workflow in the service, and the service description includes the service logic composed by the workflow. Service description of a Meta service contains the followings:

- Workflow description
  - Declaration of workflow units
  - A service logic to connect the workflow units
- Information for sharing workflows
  - User's permission
- Resource information for effective scheduling
  - Resource restriction
  - Historical performance data

The declaration of workflow units in Meta services includes one or more workflows which a user wants to use, and declares them as workflow units. The units are linked to each other, and then executed within a loop or a condition of the declaration. In an extreme case, a Meta service may include only one service. When Meta services are called, attributes of the workflows are overridden with service parameters or values defined by the user. In executing the services, instances of the workflows are generated with parameters by overriding attributes of the workflows. Because workflows are generated by calling the corresponding services with parameters, users could easily generate and reuse workflows.

A portal provides common services to its users, and the users may define frequently used services as Meta services in the portal. For this purpose, Meta services might be declared as public or other users might be permitted to use them. As a result, permitted users could read and execute the common Meta services, but their instances are generated in the users' repositories. Therefore, if a user wants to modify the service, he could modify it with a copy of the service.

And a user may want to avoid executing the workflow on faulty resources. If historical performance data is collected and recorded on the Meta services, a scheduler could save the time required to find appropriate resources. Meta services maintains lists of MER (most effective resources) and MFR (most faulty resources). Moreover, though the same services are executed on several resources with the same performance and specifications, it may be difficult to expect the same performance on Grid environments. And after a user has executed the same Meta service many times, he may find that the service is executed efficiently on specific resources or it is executed unsuccessfully on other resources. That is, Meta services are designed so that the users may execute workflows on their preferred resources by using resource profile information.

3 Conclusion

In this paper, we presented a concept of Meta services. The concept of Meta services is combined with the MSF system in order to provide more reusable and adaptable workflow management environments. Users of the combined system could browse, search, and call the Meta services with the name and parameters of a service. A Meta service is mapped to the corresponding JCML workflows, and successfully executed in MSF.

The aim of this paper is to introduce a concept of Meta services which enables the followings: Meta services are ported to various service environments such as Grid services, Web services, and portal services. Meta Services could provide user's permission information to share workflows among portal users. Meta services also provide a resource profiling information, which could be used for efficient scheduling of resources. We applied the concept of Meta Services to extend our previous work, MSF, but the concept can be applied to other WfMS. In the near future, we will integrate heterogeneous Grid workflows with Meta services.

Acknowledgement

This work was supported by Korea Research Foundation under contract KRF-2004-005-D00172.

References

1. I. Foster and C. Kesselman, ed., The Grid: Blueprint for a New Computing Infrastructure, Morgan Kaufmann, (1998)
2. J. Cao, S. A. Jarvis, S. Saini and G. R. Nudd, GridFlow: Workflow Management for Grid Computing, 3rd International Symposium on Cluster Computing and the Grid, (2003) 12-15
3. R. Stevens, A. Robinson and C. Goble, myGrid: Personalized bioinformatics on the information grid, Bioinformatics, 19(1), (2003) 302-304
4. G. Allen et al., Enabling applications on the Grid - a GridLab overview. Intl. Journal on High Performance Computing Applications, 17(4), (2003) 449-466
5. M. Litzkow, M. Livny and M. Mutka, Condor - A Hunter of Idle Workstations, 8th International Conference of Distributed Computing Systems, (1998) 13-17
6. Y. Gil, C. Kesselman, G. Mehta, S. Patil, M. Su, K. Cahi, M. Livny, Pegasus Mapping Scientific Workflows onto the Grid. E. Deelman, J. Blythe, Across Grids Conference (2004)
7. Seogchan Hwang, Jaeyoung Choi: MSF: A Workflow Service Infrastructure for Computational Grid Environments, LNCS 3292, 445-448 (2004)
8. W. Smith, I. Foster, V. Taylor, Predicting Application Run Times Using Historical Information, Proc. IPPS/SPDP '98 Workshop on Job Scheduling Strategies for Parallel Processing (1998)