Research on an MOM-Based Service Flow Management System*

Pingpeng Yuan, Hai Jin, Li Qi, and Shicai Li

Huazhong University of Science and Technology, Wuhan 430074, China
yuanpingpeng@mail.hust.edu.cn, hjin@hust.edu.cn

Abstract. Workflow is an important approach for the specification and management of complex processing tasks. This approach is especially powerful for utilizing distributed service and processing resources in grid systems. Some researches on the combination of workflow and grid exist, leading to many approaches to grid workflow support. We describe our *MOM-based Service Flow Management System* composing complex processing chains. We describe the technologies to enable the execution of these processing chains across wide-area computing systems.

1 Introduction

Grid computing aims to utilize distributed services owned or provided by different organizations as a single, unified service. For example, many computing and data intensive scientific applications require a series of tasks to solve sophisticated problem collaboratively. Thus, how to glue services distributed over Internet and across organizations is the key problem for grid computing. Workflow technique can satisfy these requirements. Workflow is a technique by which a complex process is expressed as an interconnected series of smaller, less complicated tasks. The concept of workflow has successfully been used in many areas.

Due to the great capabilities of workflow to interconnect tasks, grid workflow has emerged. Some goals of grid workflow include: description of the interactions of grid services, composition of grid services and ability to trigger services if necessary. Currently, workflow systems for grid services are evoking a high degree of interest. In this paper we describe our approach to grid workflow.

The remainder of this paper is organized as follows: First, we review some related works. Then, we present our workflow definition language. In section 4, we describe our grid workflow architecture. Finally, we end this paper with conclusion.

2 Related Works

The goal of grid computing is to provide a service-oriented infrastructure that leverages standardized protocols and services to enable pervasive access to, and coordinated sharing of geographically distributed services and resources. New higher-level services and applications can be constructed from the available services through workflow.

* This paper is supported by Nature Science Foundation of China under grant 60273076, ChinaGrid project from Ministry of Education, and National 973 Basic Research program under grant 2003CB316903.

H. Jin, Y. Pan, N. Xiao, and J. Sun (Eds.): GCC 2004, LNCS 3251, pp. 783–786, 2004.

There is a limited amount of work related to the issues of the combination of grid and workflow in the grid computing community. Research efforts on grid workflow include Webflow [1], Chimera Virtual Data System (GriPhyN) [2], Symphony [3], DAGMan [4], UNICORE [5], and XCAT [6]. Webflow, one of the earlier workflow systems, supportS application composition in Grid environments. The Chimera Virtual Data System considers grid workflows as static graphs of services. Symphony is a framework for combining existing codes to meta-programs. DAGMan manages the dependencies between jobs and schedule jobs according to their dependency relation. XCAT Application Factories address workflow related issues for grid-based components within the *Common Component Architecture* (CCA) framework [7].

The key issue that differentiates our work from these is that we focus more on service-level support, workflow definition language, distributed workflow management and scheduling, as opposed to some researches at the programming level. The goals are to define a workflow definition language and provide the scheduling solution for services in distributed environment. One advantage of our approach is that services can be selected and scheduled dynamically. This is especially suitable for dynamical grid environments.

3 Workflow Definition Language

In a distributed system, users' requests can be represented as flows of services linked together. An important portion of service flow (workflow) is the definition of the process logic, which is expressed by the usage of a workflow specification language. The *Event-based Service Flow Definition Language* (ESFL) is a workflow specification language defined in our workflow approach.

In ESFL, the definition of a workflow or a process is specified in process element. The process element has a reference attribute to identify the process uniquely. The process elements are important building blocks of ESFL. The process elements include three kinds of sub-elements: *basic element*, *construction element* and *association element*.

The basic element can be any one of the following: *activity* (A step or task to be performed); *event* (occurrence of a particular situation or condition).

The *activity* types and *event* types can be constructed from construction element. The construction element is as follows: *choice* (exactly one activity (event) out of a set of activities (events)); *and* (all activities (events) out of a set of activities (events)); *or* (any one activity (event) out of two activities (events)).

The association element includes only one element: *OnEvent*. The *OnEvent* element indicates which activity should be executed when an event occurs and which event will occurs after the completion of the activity.

4 The MOM-Based Grid Workflow Management System

Users can describe the interactions between grid services by using of ESFL. Workflow system can orchestrate or control the interactions between services, which are used to perform tasks, such as computing task and storage task.

Our approach to workflow management in grid environments is *MOM-based Service Flow Management Systems* (MSFMS). MSFMS provides high-level middleware to enables transparent access to services and resources distributed over a wide-area

network. The fundamental ideas behind the MSFMS are very simple: it consists of a collection of federated servers with hosting MSFMS engine or brokers to applications, or services. The federations of processing resources, which host the MSFMS environment, all make their own placement and scheduling decisions independent of each other. The MSFMS environment provides the necessary framework for the seamless scheduling and execution of the component parts of the users' requests across the distributed system to ensure the request is fulfilled.

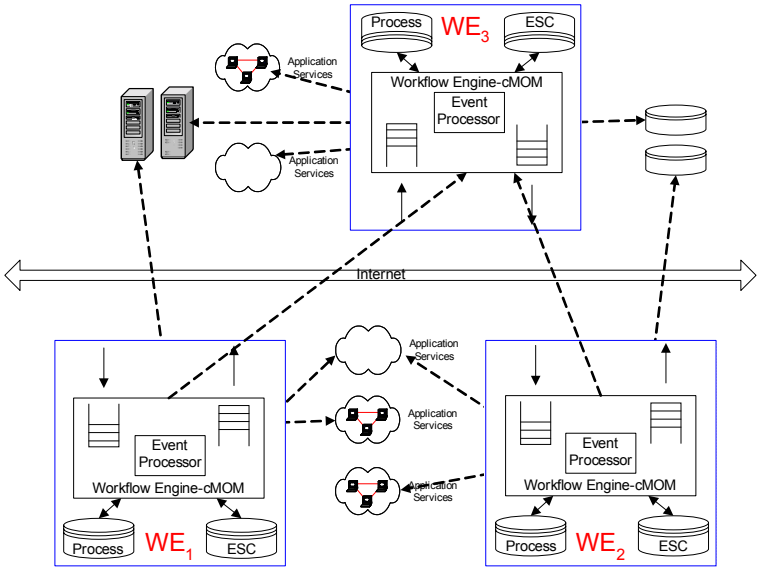

Fig. 1. The Architecture of MSFMS

The architecture of MSFMS adopts a service-oriented perspective, with a high degree of automation that supports flexible collaborations and computations on a large scale, as illustrated in Fig.1. In the architecture, workflow engines are central to the architecture of the MSFMS. Workflow engines are distributed across grid environment. We adopt the cMOM (*Composite-Event-Based Message-Oriented Middleware*) [8] as MSFMS engine. Services or resources can register themselves in one or more workflow engines. MSFMS engine can schedule those services and resources registered in the engine. An ESFL workflow definition can be seen as a template for creating grid services instances, performing a set of operations on the instances and finally destroying them. Therefore, MSFMS engines can also register themselves into other engines. By this means, MSFMS engines can be dynamically assembled into many architectures, such as Peer to Peer architecture, or layer architecture. Due to the dynamic nature of the grid environment, the MSFMS is suitable for grid environment.

5 Conclusion

In this paper, a *MOM-based Service Flow Management System* (MSFMS) is proposed. MSFMS enables the integration of services and resources within and across organizations. In MSFMS, we propose an XML-based workflow definition language,

namely *Event-based Service Flow Definition Language* (ESFL). Compared with other workflow definition language, ESFL is simple, however, it satisfies grid workflow requirements and users can compose complex service flow using ESFL. Furthermore we describe the architecture of MSFMS to demonstrate the great flexibility.

References

1. D. Bhatia, V. Burzevski, M. Camuseva, G. Fox, W. Furmanski, and G. PremChandran, "WebFlow: A Visual Programming Paradigm for Web/Java Based Coarse Grain Distributed Computing", *Concurrency: Practice and Experience*, 9(6):555–577, 1997.
2. M. Wilde, I. Foster, J. Vckler and Y. Zhao, "Chimera: A Virtual Data System for Representing, Querying, and Automating Data Derivation", *Proc. of SSDBM'02*, pp.37–46.
3. M. Lorch and D. Kafura, "Symphony: A Java-based Composition and Manipulation Framework for Computational Grids", *Proc. of 2nd IEEE/ACM Int. Symp. on Cluster Computing and the Grid*, pp.136– 143, Berlin, Germany, 2002.
4. J. Frey, T. Tannenbaum, I. Foster, M. Livny, and S. Tuecke, "Condor-G: A Computation Management Agent for Multi-Institutional Grids", *Proceedings of the 10th IEEE Symposium on High Performance Distributed Computing (HPDC10)*, pp.7–9, August 2001.
5. M. Romberg, "The UNICORE Grid Infrastructure", *Scientific Programming, Special Issue on Grid Computing*, 10(2):149–157, 2002.
6. M. Govindaraju, S. Krishnan, K. Chiu, A. Slominski, D. Gannon, and R. Bramley, "XCAT 2.0: A Component-Based Programming Model for Grid Web Services", *Technical report-tr562*, Dept. of C.S., Indiana Univ., June 2002.
7. M. Govindaraju, S. Krishnan, K. Chiu, A. Slominski, D. Gannon, and R. Bramley, "Merging the CCA Component Model with the OGSI Framework", *Proceedings of CCGrid2003, 3rd International Symposium on Cluster Computing and the Grid*, May 2003.
8. P. Yuan and H. Jin, "A Composite-Event-Based Message-Oriented Middleware", *Proceedings of the Second International Workshop on Grid and Cooperative Computing*, Sep. 2003, pp.491-498.