Computers and Electrical Engineering 37 (2011) 300–310

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Virtual workflow system for distributed collaborative scientific applications on Grids ☆

Lizhe Wang [a,*], Dan Chen [b,c], Fang Huang [d]

[a] Pervasive Institute of Technology, Indiana University, United States
[b] School of Computer Science, China University of Geosciences, China
[c] School of Computer Science, University of Birmingham, UK
[d] Institute of Geo-Spatial Information Technology, College of Automation, University of Electronic Science and Technology of China, Chengdu 611731, China

ARTICLE INFO

Article history:
Received 20 August 2009
Received in revised form 11 January 2011
Accepted 12 January 2011
Available online 18 February 2011

ABSTRACT

Grid computing has become an effective computing technique in recent years. This paper develops a virtual workflow system to construct distributed collaborative applications for Grid users. The virtual workflow system consists three levels: abstract workflow system, translator and concrete workflow system. The research highlight of the implementation is that this workflow system is developed based on CORBA and Unicore Grid middleware. Furthermore, this implementation can support legacy application developed with Parco and C++ codes. This virtual workflow system can provide efficient GUI for users to organize distributed scientific collaborative applications and execute them on Grid resources. We present the design, implementation, and evaluation of this virtual workflow system in the paper.

1. Introduction

Grid computing [1] is one of the most influential aspects of computing techniques in recent years. Distinguished from conventional parallel and distributed computing, Grid computing mainly focuses on resource sharing among geographically distributed sites and the development of innovative, high performance oriented applications. Computational Grids can present Grid users with pervasive and inexpensive access to a wide variety of resources.

Most of current Grid users are scientists and engineers from research fields other than computer science, e.g., bio-informatics, particle physics, and aerodynamics. They are not familiar with Grid techniques, such as resource selection, task preparation and submission. Thus, a friendly user layer which hides a Grid's technical details for end-users is required. Furthermore, Grid infrastructures are required to support heterogeneous middleware and application environments, for example, legacy applications and systems.

The collaborative applications are one of most important applications from scientific and engineering fields. Many tasks running in Grid environments will be executed on multiple sites by several collaborators [2].

This paper is devoted to the new modeling and executing environments of distributed collaborative applications. A virtual workflow system is designed and implemented for the purpose. Users can construct the collaborative application in the virtual workflow system without detailed Grid knowledge. To justify our work, we evaluate a use case from the aerodynamics research field. The contribution of this paper includes:

☆ Reviews processed and proposed for publication by Editor-in-Chief Dr. M. Malek.
* Corresponding author. Tel.: +1 812 856 5446; fax: +1 812 856 1537.
E-mail address: lizhe.wang@gmail.com (L. Wang).

0045-7906/$ - see front matter
doi:10.1016/j.compeleceng.2011.01.004

- We develop flexible and collaborative workflow models that support multiple workflow implementations and Grid middleware.
- We develop a virtual workflow system that can decouple the workflow collaboration processes, definition and execution.
- We demonstrate our work with a CORBA based legacy application on Unicore Grids.

The rest of this paper is organized as follows: Section 2 introduces the background and related work of Grid computing, Grid workflow and distributed collaborative applications. Section 3 defines research issues for developing collaborative applications on Grids and our proposed solutions. Sections 4 and 5 presents our collaborative workflow model and the implementation of virtual workflow system. A use scenario is selected and examined in Section 6 to justify the system and Section 7 concludes the paper.

2. Background and related work

This section introduces related work of Grid computing and its middleware, Grid workflows and distributed collaborative applications.

2.1. Grid computing and middleware

Grid computing has been one of the most active research fields in resent years. While Grid technologies have made great progress recently, there still remain technical problems towards meet various requirements of QoS (Qualities of Service) when applications are executed on different types of platforms. Unicore (Uniform Interface to Computing Resources) [3] is a reference implementation of middleware for Grid computing. The development of the Unicore system is conceived in 1997 by German Ministry for Education and Research (BMBF). Its goal is to facilitate users, who have large problems in computational science, to utilize various computing resources in different locations. Currently, Unicore software is an open source project and has been integrated into German national Grid initiative – D-Grid [4]. Our implementation is adapted to the Unicore Grid middleware.

Software component model, which emphasizes the development of an application by assembling existing pieces of software, has been accepted widely. Component model can bring some benefits, for instance, development time is reduced, codes can be reused and the resulting application presents more modularity. CORBA (Common Object Request Broker Architecture) provides a standard mechanism for defining the interfaces between components as well as some tools to facilitate the implementation of those interfaces using the developer's choice of languages. In addition, the Object Management Group (OMG) specifies a wealth of standard services, such as directory and naming services, persistent object services, and transaction services. Each of these services is defined in a CORBA-compliant manner, so they are available to all CORBA applications. In our work, the Ag2D application is implementation with parallel CORBA.

2.2. Scientific workflow on computational Grids: state of the art

Our work develops scientific workflows for warping legacy application with parallel CORBA. This section introduces the related work for Grid workflows. A workflow is concerned with the automation of procedures where documents, information or tasks are passed between participants according to a defined set of rules to achieve, or contribute to, an overall business goal [5]. A WorkFlow Management System (WFMS) is the one which provides procedural automation of business process by managing the sequence of work activities and the invocation of appropriate human and/or IT resources associated with the various activity steps [6].

A Grid workflow system is defined as: "with support from Grid middleware and Grid infrastructures, Grid workflow system defines, specifies and manages workflow on computational Grids" [7].

This section gives a summary on following important research aspects for Grid workflows:

- Control flow model of Grid workflow.
  A Grid workflow includes multiple tasks and is executed according dependencies among the tasks. Control flow, which is referred as workflow pattern in Section 2.3, models the execution order of a workflow. Some Grid workflow systems, for example DAGMan [8], Unicore [3] and GriPhyN [9] adopt Directed Acyclic Graphs (DAG) as control flow model. In DAG based control flow, workflow can be executed in patterns of sequence, parallelism and condition. Sequence means that tasks runs in serial order; parallelism represents tasks are executed concurrently; in condition pattern, the "if – then – else" pattern is employed to specify branches of control flow. In other systems, e.g., Triana [10], ASKALON [11], rich semantics are used for control flow and support iteration pattern, which allows executing tasks repeatedly.
- Data flow model of Grid workflow.
  To run a task of Grid workflow, input data should be staged to the hosts where task is executed and output data are required to be transferred to users or other tasks. To model a data flow, there are two methods: explicit method and implicit method. With explicit method, data flow is clearly specified with the data source, data destination and transfer protocols. This transfer process can be modeled as a normal task in control flow. With implicit method, tasks only define