A Survey On Grid Computing Approach

[1]Komal Vashisht, [2]Shefali, [3]Karishma Shukla
[1,2,3]CSE Dept. , Apeejay College of Engg Vill.-Silani,Sohna
[1]komal.vashisht@gmail.com, [2]shefalimadan@gmail.com, [3]karishmashukla2@gmail.com

Abstract
Grid computing enables virtual organizations to share geographically distributed resources as they pursue common goals, assuming the absence of central location, central control, omniscience, and an existing trust relationship. A recent survey ranked grid computing as sixth on priority lists for IT spending among industry professionals. This paper details the grid protocol architecture. Workflow is the orchestration of a set of activities to accomplish a larger and sophisticated goal. In this paper, we review various workflows and the management of workflow in grids and a data transfer scheme of grid workflow based on weighted directed graph. The scheme selects data transfer model according to the job's data processing rate. In a grid workflow, data transfer scheme plays a key role on its performance. This paper has also provided an insight classification of the computational and desktop grid systems on the basis of infrastructure, models, and software applications.

Keywords
Workflows, VO, GARA, gLite, MiG.

I. Introduction

Modern applications are demanding more and more computational power, which cannot be satisfied by the existing individual computers. In order to meet these computational challenges, it is necessary to have a standardized means of connecting disparate resources over high-speed networks to build high power virtual supercomputers. Grids are an emerging architecture for the execution of resource intensive programs by integrating large-scale, distributed and heterogeneous computing or storage resources [2]. For this reason the grid concept is of high interest to scientific communities with high requirements on computing or storage resources like high-energy physics, geophysics or astronomy. Moreover, scientific applications tend to combine grid computing with service orientation [3, 1]. Grid computing provides enormous opportunities in terms of resource sharing, maximization of resource utilization and virtualization of resources. A computational Grid consists of a set of resources, such as computers, networks, on-line instruments, data servers or sensors that are tied together by a set of common services which allow the users of the resources to view the collection as a seamless computing or information environment [4]. The standard Grid services include

- Security services which support user authentication, authorization and privacy
- Information services, which allow users to see what resources (machines, software, other services) are available for use,
- Job submission services, which allow a user to submit a job to any compute resource that the user is authorized to use,
- Co-scheduling services, which allow multiple resources to be scheduled concurrently,
- User support services, which provide users access to "trouble ticket" systems that span the resources of an entire grid.

II. Grid Protocol Architecture

Grids started off in the mid-90s to address large-scale computation problems using a network of resource-sharing commodity machines that deliver the computation power affordable only by supercomputers and large dedicated clusters at that time. The major motivation was that these high performance-computing resources were expensive and hard to get access. Grids focused on integrating existing resources with their hardware, operating systems, local resource management, and security infrastructure. In order to support the creation of the so called "Virtual Organizations"—a logical entity within which distributed resources can be discovered and shared as if they were from the same organization, Grids define and provide a set of standard protocols, middleware, toolkits, and services built on top of these protocols. Interoperability and security are the primary concerns for the Grid infrastructure as resources may come from different administrative domains, which have both global and local resource usage policies, different hardware and software configurations and platforms, and vary in availability and capacity. Grids provide protocols and services at five different layers as identified in the Grid protocol architecture (see Fig.1) [6].

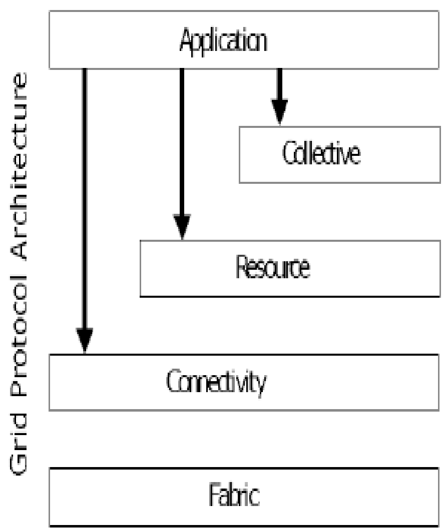

Fig. 1: Grid Protocol Architecture

At the fabric layer, Grids provide access to different resource types such as compute, storage and network resource, code repository, etc. Grids usually rely on existing fabric components, for instance, local resource managers, General-purpose components such as GARA (general architecture for advanced reservation) [5], and specialized resource management services. The connectivity layer defines core communication and authentication protocols for easy and secure network transactions. The resource layer defines protocols for the publication, discovery, negotiation, monitoring, accounting and payment of sharing operations on individual resources. The collective layer captures interactions across collections of resources, directory services. The application layer comprises whatever user applications built on top of the above protocols and APIs and operate in VO environments.

III. Workflows in the Grid

A workflow describes the order of a set of tasks performed by various agents to complete a given procedure within an organization. Whereas, workflow management generally aims at modeling and controlling the execution of complex application processes in a variety of domains, including the traditional business domain, governmental applications, the wide field of electronic teaching and learning, or more recently the natural sciences. As grid is an emerging architecture for the execution of resource intensive programs by integrating large-scale, distributed and heterogeneous computing or storage resources [7], it makes sense to verify whether the workflow concepts can be successfully used in or adapted to the context of the grid In essence, workflows are executable models of processes, and as such can distinguish a schema from an instance. The development process of workflow applications in general goes through various phases:

Information gathering involves empirical studies such as informal interviews and available documentation. The activities of this phase centers around the application.

Process modeling In this phase, the information previously gathered is used to specify process models, i.e., deal with the processes, the main purpose is to provide a general and easy-to-read notation, which enables information system experts and domain experts to validate and optimize the resulting process models.

Workflow modeling is to enhance the process model with information needed for the controlled execution of workflows by a workflow management system or a workflow engine. In this phase workflow languages are used. Workflows can be classified as shown below :

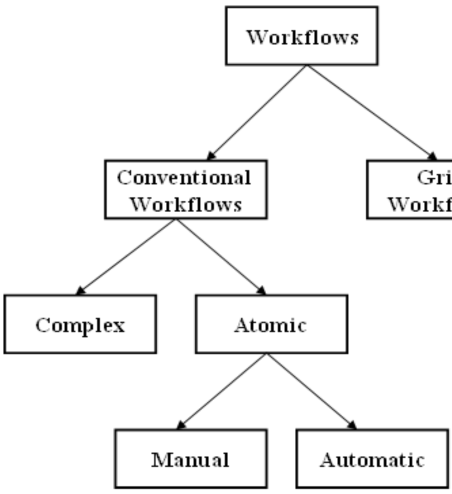

Fig. 2: Classification of Workflows

Complex workflows consist of several sub-workflows and a workflow schema defining the functional order of the workflow execution. Atomic workflows are "black boxes" for the workflow management system and their inner assembly is not known to it. Automatic workflows represent simple activities, which can be done without user interaction. The executing resources of manual tasks are humans. For each Manual workflow [10] a set of requirements regarding the roles of the executing person can be defined. Grid workflows can be represented by conventional workflow concepts. Since automatic workflows are playing a supporting role in conventional workflow scenarios. In grid scenarios also, automatic workflows play a central role, so they need to be modeled in more detail. One promising approach is to extend role concepts used for manual workflows also to automatic workflows. This extension means that additionally to humans, machines will be treated as resources too. Consequently, roles have to be assigned now to machines. By this way it is possible to define non-functional and technical properties of the different machines (resources) in the grid. Fig 3 shows a model how workflow management systems can be linked with the Grid. Grid resources register their capabilities in the grid information service, which is accessed by workflow management systems to carry out role resolution. Workflow management systems may belong to different organizations and are responsible for task scheduling and monitoring on a global scale regarding their workflow or organizational context.

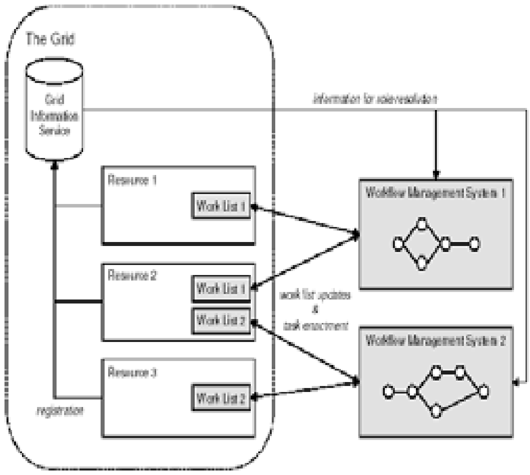

Fig. 3: Workflow Management and the Grid

IV. Data Transfer in Grid Workflow

- Data transfer scheme plays an important role in the management of grid workflows. Earlier, the data was transferred between source node and destination node directly by GridFtp [8] if needed, which resulted in the low efficiency of data transfer. The efficiency of data transfer is improved by using a data transfer scheme for grid workflow management based on weighted directed graph [9]. This scheme mainly optimizes the data transfer in two aspects.
- The flexible means of data transfer. There are many protocols and tools for data transfer between source node and destination node. It can obviously improve the efficiency of data transfer to choose a suitable protocol and tool according to different situations. This scheme flexibly chooses the efficient means of data transfer in terms of network status.
- The optimized route of data transfer. It will take a long time to transfer data directly between source and destination, if the two nodes have a connection with very low bandwidth or even no connection. This scheme finds an optimized route of data transfer by choosing some nodes with higher bandwidth as the proxies.

A workflow is composed of many tasks. Among these tasks, there are certain sequence relation and data relation. The output data of a task is probably used as the input data of the next task. The execution time of each task includes data transfer time and calculation time. The weighted graph scheme assumes the optimization method for each task is the same. So, this scheme took only one task as the target to describe the method. This scheme uses Tt and Tc to denote the data transfer time and

 ISSN : 2229-4333(Print) | ISSN : 0976-8491(Online)

the computational execution time of a task respectively. As, Tc, is determined by the performance of computing node. The resource allocation of the computing node is considered to be the responsibility of resource scheduling strategies, which are beyond this scheme; it only focuses on minimizing the value of Tt. On one hand, it chooses the correct data transfer protocol and tool. On the other hand, it chooses the optimized route for data transfer by constructing and analyzing the weighted directed graph. In general, the basic procedure of our scheme is as follows. First, the scheme filters the unqualified nodes according to the status of their hardware and the requirement of the grid workflow. Then, it chooses the data transfer method between every two nodes according to the information service of grid system. And, it builds a weighted directed graph in terms of the weight calculated by the bandwidth of every two nodes. Finally, the scheme forms the best route to transfer data based on the graph through the improved
Dijkstra algorithm.

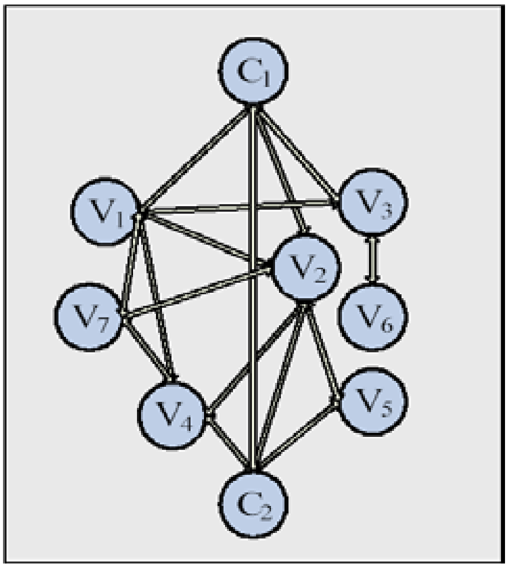

Fig. 4: The Weighted Directed Graph G

The scheme chooses the best transfer means and calculates out the transfer cost as its weight. Then, the scheme compares many transfer routes between C1 and C2 in G', and chooses the most effective path. It delete some useless vertices and edges to get a new weighted directed graph G'.

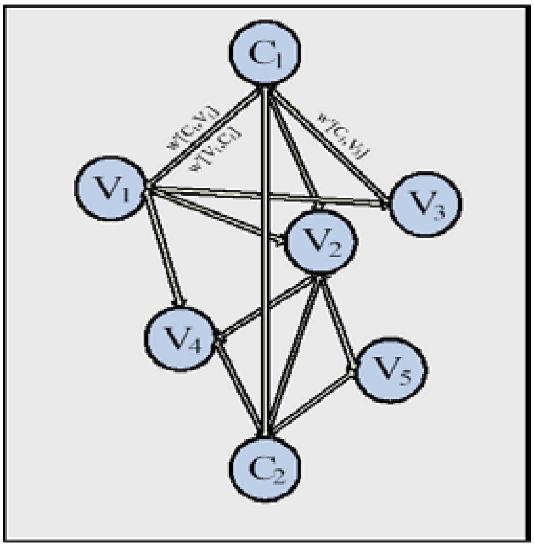

Fig. 4: The Reconstructed Weighted Directed Graph G'

V. Computational and Desktop Grids

A Computational grid states that it is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities. Whereas, [11] Desktop grid refers to a grid infrastructure that is confined to an institutional boundary, where the spare processing capacity of an enterprise's desktop PCs are used to support the execution of the enterprise's applications. In Computational Grids Infrastructure is needed because the computational grid is concerned with large-scale pooling of resources, whether compute cycles, data, sensors and people. Dependable service is required because users need assurances that they will receive predictable, sustained, and high levels of performance from the various components that constitute the grid. Consistency of service is to encapsulate heterogeneity without compromising high performance execution. Pervasive access provides services that are always available, within, almost every environment we expect to move.

Table 1: Key components of Computational and Desktop grids

KEY COMPONENTS	
Computational Grids	Desktop Grids
Infrastructure	Physical node management
Dependable service	Resource scheduling
Consistency of service	
Pervasive access	Job scheduling

In Desktop Grids the Physical Node Management is expected to provide naming, communication, resource-management, application control, security, process-management including file staging, application initiation and termination, error reporting, node recovery, terminating runaway and poorly behaving applications. Resource-scheduling layer must adapt to all kind of changes in resource status and availability, and also to high failure rates. The Job management provides simple abstractions for users, delivering a high degree of usability in an environment where it is easy to drown in the data, computation, and the vast number of activities. Computational and Desktop Grids can be classified on the basis on Infrastructure, Models and Software applications and platforms etc. some of them are given in Table 2.

Table 2: Classifications of Computational and Desktop Grids

CLASSIFICATIONS	
Computational Grids	Desktop Grids
gLite	Distributed.net
NorduGrid/ARC	Entropia
UNICORE	SETI@home
MiG	Bayanihan
WebCom-G	Condor
Office Grid	BOINC

VI. Conclusion

In this paper we discuss the Grid protocol architecture, workflows in grid environments in general. Scientific applications impose new requirements to workflow support, namely flexibility and distributed logging. None of these are sufficiently addressed in current grid workflow proposals. In order to be useful for experiment management and data lineage, these types of requirements need to be supported adequately. The data transfer is very important in the whole workflow execution in grid environment. This paper reviews a data transfer scheme to economize the cost of data transfer and execution of workflow and improve the performance of the workflow. We have also drawn a brief comparison between computational and desktop grids.

References

[1] Dennis Gannon, Randall Bramley, Geoffrey Fox, Shava Smallen, Al Rossi, Rachana Ananthakrishnan, Felipe Bertrand, Ken Chiu, Matt Farrellee, Madhu Govindaraju, Sriram Krishnan, Lavanya Ramakrishnan, Yogesh Simmhan, Alek Slominski, Yu Ma, Caroline Olariu, Nicolas Rey-Cenvaz, "Programming the Grid: Distributed Software Components, P2P and Grid Web Services for Scientific Applications", Department of Computer Science, Indiana University.

[2] Dominik Kuropka1, Gottfried Vossen2, Mathias Weske1, "Workflows in Computation Grids", Hasso Plattner Institute at the University of Potsdam, Prof.-Dr.-Helmert-Str. 2-3, 14482 Potsdam, Germany, European Research Center for Information Systems at the University of Münster, Leonardo-Campus 3, 48149 Münster, Germany

[3] Foster, Ian; Kesselman, Carl, "The Grid: Blueprint for a New Computing Infrastructure", 2nd Edition, Elsevier, 2004.

[4] Foster, Ian; Kesselman, Carl, "The Grid: Blueprint for a New Computing Infrastructure", 2nd Edition, Elsevier, 2004.

[5] Huhns, M.N., Singh, M.P., "Service-Oriented Computing: Key Concepts and Principles", IEEE Internet Computing Jan./Feb. 2005, pp. 75 – 81.

[6] I.Foster, C. Kesselman, C. Lee, R. Lindell, K. Nahrstedt, A. Roy., "A Distributed Resource Management Architecture that Supports Advance Reservations and Co-Allocation", Intl Workshop on Quality of Service, 1999.

[7] Ian Foster, Yong Zhao, Ioan Raicu, Shiyong Lu, "Cloud Computing and Grid Computing 360-Degree Compared", Department of Computer Science, University of Chicago, Chicago, IL, USA.

[8] Ludäscher, B., I. Altintas, C. Berkley, D. Higgins, E. Jaeger-Frank, M. Jones, E. Lee, J. Tao, Y. Zhao, "Scientific Workflow Management and the Kepler System", Concurrency and Computation: Practice & Experience, Special Issue on Scientific Workflows, 2005

[9] Nicolae-Zoran Constantinescu-Fülöp, "A Desktop Grid Computing Approach for Scientific Computing and Visualization", Department of Computer and Information Science Faculty of Information Technology, Mathematics and Electrical Engineering Norwegian University of Science and Technology, May 2008

[10] Song Wu1 Hai Jin1 Kang He1 Zongwei Luo2, "A Data Transfer Scheme of Grid Workflow Based on Weighted Directed Graph*", 1Cluster and Grid Computing Lab, School of Computer, Huazhong University of Science & Technolgoy, Wuhan, 430074, China, 2E-Business Technology Institute, The University of Hong Kong, Hong Kong, China

[11] W. Allcock, J. Bester, J. Bresnahan, A. Chervenak, I. Foster, C. Kesselman, S. Meder, V. Nefedova, D. Quesnel and S. Tuecke, "Data Management and Transfer in High-Performance Computational Grid Environments," Parallel Computing Journal,2001.

Komal Vashisht
Lecturer CSE Department
Apeejay College of Engineering
Sohna, Gurgaon.

Shefali Madan
Lecturer CSE Department
Apeejay College of Engineering
Sohna, Gurgaon.

Karishma Shukla
Sr. Lecturer CSE Department
Apeejay College of Engineering
Sohna, Gurgaon.