SURVEY OF WORKFLOW MANAGEMENT TOOLS FOR GRID COMPUTING ENVIRONMENT

Suresh Kumar N*
School of Computing Science and Engineering, VIT University, Vellore, Tamilnadu, India.
Email:sureshkumar.n@vit.ac.in

Received on 15-05-2016
Accepted on 09-06-2016

Abstract

Commonly Grid Computing is the group of simple computer resources from various admin domains, which is constructed to reach a common objective or goal. Similarly Grid is same as distributed system network with different tasks. In general Grid can have number of slaves to give high performance for achieving targets of large computing. It has a rich value by having focus on large scale resource sharing, modern applications, and very good target point of reference. Here, proposing a classification that distinctiveness and ordering the variety of approaches for structuring and executing workflows on Grid Computing, and also survey on some Grid Workflow systems developed by different grid based projects to display the depth of the classifications. This survey not only highlighting the design and working similarities, this will showing us and identifies the need of further improvements on Workflow management of Grid computing system.

Keywords: Grid Computing, Distributed Computing, Grid Ant, GWorkflow DL, Pegasus, Dagman.

Introduction

Grid computing combines computers from many managerial domains to achieve a common goal, to solve a single job. Use a middleware methodology to split and assign parts of a program among numerous computers is one of the main significant strategies of Grid Computing system.

Commonly the size of a grid may vary from small to high with respect to the job, small grid network is limited to network computers within a organization, large grid network is public collaborations with lot of organizations and networks. Workflow management system is used to characterize the management and execution of workflow processes, and it is used to carry out and achieve some specified job. For that it is undergo with many different execution of
computerized definitions. This workflow management tools helps to lead the way to destiny for execution of defined job.

**Stages of Grid Workflow Management tools**

Grid computing is a unique kind of parallel computing that relying on total computers and its physical resources. Every physical resources of different various computers connected to a network by enabling connection between them. This connected network has many processors which are basically connected by very high speed computer buses \[^1\]. To manage a software complex issue, connected systems are frequently separated into a number of dependent components. This issues and services show the challenges in software engineering.

Workflow management is used for important and large computing business applications such as computational sciences \[^2\]. The important usage of Distributed computing is each and every node can be bought as product hardware to create a computing resource as super-computer. But it should be at a very lower cost, because of the economical of produce the product hardware. It is also similar as custom supercomputers \[^16\].

The main disadvantage in performance is that there is no quick connection between various processors and its respective local storage areas.

So these resources can easily make a many parallel computations without any need of intermediates \[^1\]. Physically detached grids with high end scalability are generally positive, because of the very low necessity for connection between the computing nodes.

In between the Hardware and the Software the middleware layer is placed. Total technical areas situated on top of the middleware. More important care is given to the technical areas for a profitable solution \[^6\]. In Grid computing, Middleware layers are used to communicate and control a variety of hardware and its data sets.

Basically workflow management system is for manages and defines different types of jobs within a network to give a better output. The workflow management system always relies on software which controls within it; here every group is answerable for their specified task.

The workflow software allocate those are responsible for the next job, and receive the output data that is require for execute the whole process \[^3\].
A workflow management system controls the assigned automated process to achieve a computed output and also for replace a common job order transfers. Above the figure 1 diagram shows us a architecture of Grid Workflow Management system. This architecture describes a various items between the Grid users and Grid resources. Grid Users directly contacted to the grid workflow application, it gives a grid workflow specification. Grid workflow enactment service has three items within it, which are workflow scheduling system, Data movement and Fault management. This will be directly communicated with Grid Middleware. Grid information Services communicated with two workflows, which are grid workflow application modeling and definition tools and grid workflow enactment service, this communication is also known as interaction with information services [9].

**Figure 1: Architecture of Grid Workflow Management System.**

**Figure 2: Essential parts of Grid Workflow Management System.**

**GridAnt: A Grid Workflow System**

GridAnt gives a framework for instantly prototyping grid based applications on XML specifications. GridAnt pre-define basic jobs, which are file transfer, authentication and job execution. GridAnt is based on the product tool for ant. So it
offers the ability to use the features and requirements from ant. It gives an important definition of basic tasks performed in Grids, which is achievable to use the similar application specification on a variety of Grid hosting environments\[^4\]. The reputation of UNIX and DOS was initiated through the accessibility of sophisticated shells to state command line sequences\[^15\]. Make files have a necessary part in automating the difficult build process leading to an uprising in software engineering. It permitted users to make up the code and to construct refined libraries with ease. For the Java Platform a similar tool available and comes under the Apache Project with the name of Ant. Ant is used by a many number of people, it has been transitioned to a business tool and is the regular for controlling the build the processes in Java. Grid workflows, proposed to reuse the Ant framework to develop a simple so far powerful client side workflow system for Grids, which named as GridAnt\[^10\]. GridAnt proved to be an excellent tool, which is not only to map critical client-side workflows, but also as a simple client to measurement the functionality of different Grid services. GridAnt is not claimed as a substitute for more sophisticated and powerful workflow engines such as BPEL4WS, XLANG and WSFL. The basic idea of GridAnt project is to use workflow engine, which exists with Apache Ant and develop a Grid workflow dictionary on top of the workflow engine\[^8\]. The main job of Apache Ant is to make process described in make files and it reliant upon each other. Usually the job of Ant is building the Java applications. It has inbuilt tasks such as compile, assemble, test and run Java applications.

**GWorkflow DL**

Full form of GWorkflowDL is Generic Workflow Description Language. It is developed in background of the K-Wf Grid and the CoreGRID projects. This is XML-based language, which is in the form of Petri nets to explain the characteristic of distributed Grid Jobs. The Grid Job Description Language (GJobDL) was developed for the Resource Grid\[^11\].

The Grid Workflow Description Language has many features like the systems of categorization of Petri nets. It has many operations defined itself such as WS Class Operation. In G Workflow DL has good transition within a sub net with a hierarchical approach. Edges of G Work flow DL link the transition to Input places and Output places. Its expressions are used to allocate input places to a specific parameter of WS operation. These transitions may be noted with a number of conditions. These conditions could be in the form of XPath queries; every user defined modules of
web service return a Boolean value. True or False tokens are positioned on high-level information data or the web service operations. This token (true or false) is used to stores the status of the workflow.

**Condor DAGMan**

DAGMan is the abbreviation of Directed Acyclic Graph Manager. This is meta-scheduler for the Condor. The main job of DAGMan is supervising the dependency between condor scheduler and higher level jobs. It is also used to define a set of programs such as where the input, where the output or execution of programs. Usually it is dependent on other programs. In the graph, the nodes are noted as vertices, edges are noted as arcs. Condor also finds networks for the execution of programs, but it cannot schedule the programs based on the dependencies. This is also known as meta-scheduler for jobs. The main job of DAG Man is submits the jobs to Condor by a Directed Acyclic Graph and processes the results. Every node in the DAG wants its own Condor, because to submit a description file. DAG uses a particular Condor log file to submit the jobs and enforce the commands. From this we came to know that DAG is a input file of DAG Man, and Scheduling, recovery and reporting to the Condor these are works under the in charge of DAG Man.

**Pegasus**

Pegasus having a important place on workflow description, which is in the form of Directed Acyclic Graph in XML format. Programmers, no one have a willing to write workflows in the forms of XML based, and users also wants to use a very simple language that they very much interested and mostly they expected that should be in form of Graphical User Interface, that is for just click and point it. These Pegasus incorporates into different types of workflow instances generation systems like Virtual Data Language, Wings, Triana, and they recently developed new prototype integration with Kepler.

Wings use a comfortable semantic descriptions method and workflow templates in terms of domain ontology’s and constraints. Template editor is used to edit composed components and its data flows. Wings are used to assist the user by enforcing the constraints especially specific workflow components.

It also helps to data selection to make sure the datasets selected for the workflow template. Computations and net data products are specified by the workflow instances, which are generated by the Wings. These new data products generate a metadata from the defined input all through the constraints and descriptions, which is specified for the each working components.
In general scientific applications usually want to offer users with an easy interface, which is in the form of metadata query. It is incorporated into a portal environment, because the users are ready to fill the metadata attributes in the available web form. Based on the users input, the workflow instance in generated and directly given to the Pegasus for the purpose of mapping and it is given to DAGMan for the purpose of implementation and execution. Workflows are directly sent to DAGMan by the Pegasus for the purpose of execution of the workflow components, and it includes data registration, data movement, and remote job execution. GSI gives the full authentication to the remote resources. We can get provenance data about the execution tasks during the workflow execution, these provenance consists various information such as environment variables, runtime, and the host information.

**Pegasus Workflow Management System**

PWMS includes mapper and workflow executor for the DAGMan. The Pegasus mapper takes tough applications assigned as workflows and maps them to cyber infrastructure resources and submits it to the DAGMan for the execution purpose between cluster and campus grids. For example with the help of the Pegasus earthquake scientist can easily generate the affected area maps, it can be used by the civil engineers to design and create a new construction, they can use it as a earthquake-prone areas. Similarly Astronomers use the Pegasus to generate a large scale, science grade mosaics of the sky, that allow them to see structures not captured before. With the help of the Pegasus Physicists can easily analyze the gravitational waves.

Pegasus is composed by three components namely Mapper (Pegasus Mapper), Execution Engine (DAGMan) and Task Manager (Condor Schedd). It maintains the perfect workflow management system to achieve the goal of the job.

![Figure 4: Layered Architecture of the Pegasus Workflow Management System.](image-url)
The workflow map engine is a compiler, which is used to translate in between advanced terms and primary execution part of the system. The mapping process consist a task of find the suitable software and hardware resources, same as execution takes place to find copies of information data indicating in the workflow instance \[^{12}\]. This process also involves in workflow improvement towards optimizing the performance of overall workflow. The result of mapping process is in the form of executable workflow, which is executed by a workflow engine. This engines follow the workflow dependencies and executes the activities as per defined in the multiple nodes. The performance of workflow mapping and workflow optimizations will be same, it enables us to run a large scale scientific workflows with powerfully and reliably.

We describe some of these key features is given below.

- Locating Physical locations for both data and workflow components by automatically.
- It engages to find resources to perform the computations.
- Allow the users to publish their data into some data registry.
- To avoid replications of data, Pegasus uses some valid names with referenced in the workflow to the data registry \[^{7}\].
- Users or any community can easily discover the new data products.
- Performance of Static linked executables supported by Pegasus. These executables can be treated as input data and transferred to the remote grid sites \[^{13}\].

Comparison of Grid Workflow Management Tools

Table 1 represents the list of available workflow management tools and its features.

Table-1: Workflow Management Tools and it salient features.

<table>
<thead>
<tr>
<th>Workflow Management Tools</th>
<th>Important Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid Ant</td>
<td>It is product tool for Ant. It permits users to make their own code and to construct refined libraries easily. It is simple tool to measure the functionality of different Grid services. It has inbuilt tasks such as compile, assemble, test and run Java applications.</td>
</tr>
<tr>
<td>GWorkflow DL</td>
<td>This is XML-based language; it explains the characteristic of distributed Grid Jobs. It has the features of categorization of Petrinets and it has many operations defined itself such as WS Class Operation</td>
</tr>
</tbody>
</table>
Condor DAGMan | This is meta-scheduler for the Condor. Its main job is supervising the dependency between condor scheduler and high level jobs and submit the jobs to Condor by a Directed Acyclic Graph and processes the results

Pegasus | GUI based workflow description. It has different types of workflow instances like VDL, Wings, Triana and Kepler. It maintains the perfect workflow management system to achieve the goal of the job.

Conclusion

In this paper, we analyzed the workflow management system systematically and found Grid Computing Workflow Management System has great potential to build the useful computational grids. And we analyzed the Grid Workflow processes and its issues. We investigated mapping validations and verifications. The described techniques provide a solution to solving problems of workflow management issues. It indicates grid workflow management is used to optimize the workflow execution time and solve other conflicts. It may be useful in very high dynamic grid environments. So this paper tells the conclusion, that cooperation is needed among these explained grid services and components to challenge the security and performance issues.

References:


**Corresponding Author:**

Sureshkumar N*,

**Email:** sureshkumar.n@vit.ac.in