Grid Workflow Scheduling Based on Task and Data Locations

Yujin Wu\textsuperscript{1}, Ning Gu\textsuperscript{1}, Yuwei Zong\textsuperscript{2}, Zhigang Ding\textsuperscript{2}, 
and Shaohua Zhang\textsuperscript{2}, Quan Zhang\textsuperscript{2}

\textsuperscript{1} Department of Computing and Information Technology, Fudan University, 200433, Shanghai, China. \{yujinwu, ninggu\}@fudan.edu.cn 
\textsuperscript{2} Shanghai Development Center of Computer Software Technology, 201114, Shanghai, China.

Abstract. Grid workflow systems provide mechanisms to execute complex tasks which consist of related sub tasks. Due to the intensive computing and data transferring in Grid workflows, the locations of tasks and data have great impact to the execution performance of Grid workflows. In this paper, we present a novel approach to search for optimal Grid workflow scheduling effectively. We model workflow execution with fetching input data and running tasks, and present a optimized scheduling searching algorithm based on simulated annealing, which can find neighborhood scheduling fast. The experimental results show that our approach is effective and scalable.

1 Introduction

With the development of Grid computing technologies, it is possible to share resources and solve complex problems among dynamic, multi-institutional virtual organizations coordinately \cite{1}. Grid workflow systems are Grid services responsible for constructing, verifying, execution and monitoring distributed workflows in Grid. Due to the large quantity of Grid resources with different speed, memory and network bandwidth, it is very difficult to schedule workflow tasks on Grid. Two factors must be considered in Grid workflow scheduling:

\textbf{Task Location}: workflow tasks can be run on different resources. Generally resources with higher performance run tasks faster. However, there may be more tasks assigned on fast resources, so later assigned tasks need to wait until previous tasks complete. A “fast and idle” resource may be more suitable for the task.

\textbf{Data location}: workflow tasks usually need some input data. If a task need to use data in other Grid resources, the data must be transferred to where task will be executed in. Consequently, if the location of data is “closer” to the task, the task can be started and completed earlier. In data-intensive workflows, data location has great effect on workflow’s execution performance.

\textsuperscript{*} Project Supported by the National Natural Science Foundation of China (Grant No.: 60473124), 863 Hi-Tech Research and Development Program of China (NO: 2004AA1Z2390) and SEC E-Institute: Shanghai High Institutions Grid.
In this paper, we present a novel approach to address these two factors in Grid workflow scheduling problem. We model a Grid task with two phases: fetching input data and executing the task. Then WETC algorithm is presented to compute a workflow’s execution time under a given scheduling. To reach the short execution time, SASS and EWETC algorithms are presented to search for optimal scheduling in Grid with simulated annealing approach. Finally, we experimentally evaluate the model and the algorithms presented.

The rest of this paper is organized as follows. In Section 2, related work is discussed. The workflow scheduling model is presented in Section 3, and scheduling search process is described in Section 4. Experiment evaluations are presented in Section 5. Finally Section 6 contains our conclusions and directions for future work.

2 Related Work

Scheduling mechanisms are quite different in various Grid workflow systems. ICENI[4] predicts each task’s running time on a specific resource, computes total cost of workflow, and searches for scheduling with lowest cost. However, it does not consider the data location and frequent data transfer within tasks. GridFlow[2] model workflow scheduling in two levels, and simulates workflow execution like a “just-in-time” way, so it can not provide the global optimal execution performance. Pegasus[3] map workflow tasks to resources before execution, but performance issue is not considered. In some other systems, tasks are scheduled to “best idle” resources just-in-time, which tries to reach the single task’s best performance. This scheduling mechanism is simple and fast, but in most cases it is not optimal for the global workflow.

3 Grid Workflow Scheduling Model

In this section, we first present the model of Grid and workflow. Based on the model the WETC algorithm is presented to compute workflow’s execution time under a specific scheduling.

3.1 Grid Workflow Model

We use DAG as our process model, which is used in most Grid workflow systems. First we define files used in workflows. We use F to denote all files in a workflow. Let F = \{f_1, f_2, \ldots, f_m\} and let |f_i| denote the size of file f_i. Next we define the Grid task.

Definition 1. A Grid task T_i is a tuple \((I_i, C_i, O_i)\) where I_i is the set of input files, O_i is the set of output files, and C_i is the computing task T_i need to perform. We use T to denote all tasks in Grid. Not losing generality, we assume tasks in T are ordered.

Definition 2. A Grid workflow is a directed acyclic graph \(W = (V, E)\). V is the set of vertexes and E is the set of edges.

Each vertex in V is a Grid task T_i, so V is also the set of tasks in workflow W. The directed edge \((T_i, T_k) \in E\) means the start time of T_k must be equal to or later than the end time of T_i. We say T_i is a previous task of T_k and T_k is a subsequence task of