Multi-Performance Optimization for MAS Based Grid Computing

SHENGGI YU, YANPING XUANG*, ZONGYI XU, SA MENG, and QIANG LI
School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, P.R. CHINA
(Received on August 22, 2013; revised on September 13, 2013)

Abstract: The challenge of multi-performance optimization has been extensively addressed in the literature based on deterministic parameters. In Grid Computing platforms, since resources are geographically separated and heterogeneous, it is rather difficult to apply a uniform distribution algorithm for achieving various optimization goals. This paper proposes a multi-agent system (MAS) based approach for optimal network resource distribution to satisfy requirements of both users and service providers. Moreover, agents’ communication is discussed and simulation is described.

Keywords: multi performance optimization, multi-agent system, grid computing, global agent, scheduling agent

1. Introduction

Cloud Computing and Grid Computing are newly emerging technologies for solving large-scale resource sharing and global-area collaboration problems, and many important developments are being generated around the world. Due to the size and complexity of the system, its performance and reliability are difficult to model, analyze, and evaluate [1]. In the Grid Computing platform, resources are geographically separated and heterogeneous. Therefore it is difficult to apply a uniform distribution algorithm. In addition, the users’ requirements which include one or more of real-time guarantees, low costs, high reliability and security are increasingly diverse. From the perspective of the service providers, they want to maximize their profits by striking a balance between execution time, reliability, security and cost. Thus the current architectures and algorithms of traditional networks could not be well adapted for large scale complex networks.

In order to maximize the provider's profit on the basis of meeting the users’ different requirements, this paper presents a Multi-Agent Systems (MAS) approach for the optimal service task partition and distribution among the agents.

2. Proposed Method

In [2], networks with star or tree topologies and task sharing controlled by a Resource Management System (RMS) were considered. The automatic resource allocation can be implemented using MAS, which offers a natural extension to RMS as they allow multiple resource providers to inter-operate through negotiation.

2.1 Service Execution by the Grid System Based on MAS

To increase the platform’s fault tolerance, all agents are defined as intelligent control loops composed of one or more modules. Every control loop supports scheduling activities by implementing the foundational steps performed by an autonomic MAPE (Monitor-Analyze-Plan-Execute) loop [3]. Modules that define an agent correspond to the
monitor, analyzer, planner and executor (cf. Fig. 1). Environment data received through the monitor module is analyzed by the analyzer module. According to environment data, tasks are planned by planner module and allocated among available resources by the executor module. The monitoring of resources and task-related conditions feed back to ensure the rescheduling process.

A more detailed design of two types of agent is as follows.

![Figure 1: The MAPE (Monitor-Analyze-Plan-Execute) Loop](image)

### 2.2 Global Agents

The monitor module is responsible for receiving tasks and gathering status information on task execution. Each task carries the requirements information such as minimum execution time, lowest cost, highest reliability, and maximum expected profit.

The monitored information is then sent to the analyzer for formulating the selection strategies of the scheduling agents. For each task, according to the requirements, the analyzer quantifies its performance indexes, states the redundancy and then constructs optimization goal model that is the selection strategies of the scheduling agents.

The planner then broadcasts a message containing the information received from the analyzer to all/part of the available scheduling agents. After receiving the replies, the planner selects one or several appropriate scheduling agents that can meet the requirements and send notification(s) to activate it/them into the selected scheduling agent(s). The final execution results are reported to the executor of the global agent, which either outputs the results of the implementation to the customers or triggers a rescheduling process.

### 2.3 Scheduling Agents

The monitor modules are responsible for receiving tasks from global agents, gathering information regarding the current situation of the tasks and computational resources, and reporting the necessary information to the global agents. Whenever a new task is ready to be scheduled, the monitor gathers the relevant information of the task from the global agents and then sends this information to the analyzer module.

The analyzer module guides the intra-scheduling process by the effective optimization tools. The optimization goal of the task partition and distribution is provided by the received optimization goal model. Once the policy that provides the optimal solution is selected by the global agent, the selected policy is sent to the planner module by the analyzer.

The planner module assigns tasks to specific resources for execution according to the selected scheduling policy. The executor module performs the execution plan, and reports the final execution results to the monitor module. As soon as the execution is completed, the results are collected and sent back to both the monitor and the global agents. Since a completed execution does not necessarily mean the service is successful, the monitor...
module queries the specific status from the executor to trigger a rescheduling process in case the execution has failed.

3. Agents Communication and Simulation

Each scheduling agent in charge of a set of resources can be considered as a small RMS to execute the real and specific task. When the RMS receives a service request from the global agents, the task is then divided into a set of EBs that can be executed in parallel. The RMS assigns its EBs to the resources available for execution. After the resources finished the assigned jobs, they return the results to the RMS, and then the RMS integrates the received results into the entire task output which is requested by the global agents. Algorithm that the global agent assigns a task to specific scheduling agent and achieves the goal of profit/reliability/security maximization or cost minimization has been proposed. Here, we only summarize the communication process briefly.

Global agent:
Global agent assigns a task to specific scheduling agent and achieves the goal of profit maximization.

Step 1: Remove one task from the ready queue.
Step 2: Broadcast messages containing detailed parameters and optimization goal of this task to all/part-available scheduling agents according to the quantified performance indexes.
Step 3: Get replies from scheduling agents:
For each scheduling agent, assign the task to the scheduling agent whose profit obtains the maximum value while meeting the requirements on expected reliability, security and service price.
Step 4: If the task is successfully executed, return to step 1.
Else insert the task into the ready queue, then return to step 1.

Scheduling agent:
The universal generating function technique and genetic algorithm presented in [2] allow one scheduling agent to determine the provider's profit for arbitrary task partition and distribution. By using the distribution algorithm [2], the expected provider's profit is estimated. Global agent achieves the expected provider's profit and selects an optimal scheduling agent. The selected scheduling agent communicates with the global agent and executes on the actual resources. Once it is done, the final execution results will be reported to the global agents.

We complete the simulation by running tasks on 50 groups of resources that are simultaneously managed by different scheduling agents. The attributes of these resources and their corresponding communication channels, such as processing or transmission speeds, failure rates and per time unit costs, are different. We put the scheduling agents in competition such that they formulate scheduling strategies based on the current situations. The simulation results demonstrate the effectiveness of the proposed approach.

4. Conclusions
This paper proposes an MAS-based approach for optimal network resource distribution according to users’ different requirements. To increase fault tolerance, all agents are defined as intelligent MAPE control loops composed of 4 modules. Agent’s communication is described and a simulation platform is built. Finally, the results demonstrate the effectiveness of the proposed approach.
Acknowledgment

This paper was supported by the National Natural Science Foundation of China (No. 61170042), Key Technology R&D Program of Jiangsu Province (No. BE2012029) and Key Technology R&D Program of Sichuan Province (No. 2012GZ0001).

References


Shengji Yu received his B.S. and M.S. degrees from the University of Electronic Science and Technology of China (UESTC) in 2001 and 2004 respectively. Now he is a lecturer at the School of Computer Science & Engineering at UESTC, and research associate at the Collaborative Autonomic Computing (CAC) laboratory at UESTC. His current research interests include cloud computing, cloud storage and operating system.

Yanping Xiang received her B.S. and M.S. degrees from Shanghai Jiao Tong University in 1993 and 1996, respectively; and Ph.D. degree from National University of Singapore in 2003. In 2008, Dr. Xiang joined the School of Computer Science and Engineering, University of Electronic Science and Technology of China (UESTC). Now she is an Associate Professor at the Collaborative Autonomic Computing (CAC) laboratory at UESTC. Her current research interests include cloud computing and decision making.

Zongyi Xu received her Bachelor's degree from University of Electronic Science and Technology of China (UESTC) in 2012. Now she is a Master student at the Collaborative Autonomic Computing (CAC) laboratory in UESTC. Her research interests focus on cloud computing and large-scale data analysis.

Sa Meng is a Master student at the Collaborative Autonomic Computing (CAC) laboratory in University of Electronic Science and Technology of China. Her research interests focus on cloud computing and multi-agent systems.

Qiang Li is a Master student at the Collaborative Autonomic Computing (CAC) laboratory in University of Electronic Science and Technology of China. His research interests focus on cloud computing.