Chapter 104
Efficient Service Task Assignment in Grid Computing Environments

Angelos Michalas
Technological Educational Institute of Western Macedonia, Greece

Malamati Louta
Harokopio University of Athens, Greece

INTRODUCTION

The availability of powerful computers and high-speed network technologies is changing the way computers are used. These technology enhancements led to the possibility of using distributed computers as a single, unified computing resource, leading to what is popularly known as Grid computing (Foster, 2001).

The term Grid is adopted from the power Grid which supplies transparent access to electric power regardless of its source. Cloud computing, scalable computing, global computing, internet computing, and more recently peer-to-peer computing are well known names describing the Grid technology in distributed systems.

Grids facilitate the employment of various nodes comprising supercomputers, storage elements and databases that are distributed for resolving computational demanding problems in many disciplines of science and commerce (Foster, 2001). To utilize Grids effectively, an efficient allocation algorithm is needed to assign service tasks to Grid resources. Thus, assuming that a user wishes to perform a specific service task, which can be served by various candidate Grid nodes (CGNs), a problem that should be addressed is the assignment of the requested service task to the most appropriate Grid node. In this paper, the pertinent problem is called Service Task Allocation (STA).

This study is related to the pertinent previous work in the literature, since efficient resource utilization, load balancing and job scheduling are topics that attract the attention of the researchers, as computational Grids have become an emerging trend on high performance computing. Most studies in the field of resource allocation schemes aim at

DOI: 10.4018/978-1-61520-611-7.ch104

efficiently utilizing the otherwise unutilized computing power spread throughout a network. Different global objectives could be considered, such as minimization of mean service task completion time, maximization of resources utilization (e.g., CPU time), and minimization of mean response ratio, while in most cases load balancing among nodes is considered.

A high level problem statement addressed in the current version of this study may be the following. Given the set of candidate Grid nodes and their layout, the set of service tasks constituting the required services, the resource requirement of each service task in terms of CPU utilization, the characteristics of each Grid node, the current load conditions of each Grid node and of the network links, find the best assignment pattern of service tasks to Grid nodes subject to a set of constraints, associated with the capabilities of the Grid nodes. The proposed service task allocation scheme handles complex services composed by tasks requiring communication (i.e., message exchange) with other service components (e.g., databases). Care is also taken in case there is no resource with available spare capacity so as to accommodate a new service task on a congested system.

Our approach uses an Ant Colony Optimization algorithm (ACO) for service task allocation in Grid computing environments. ACO actions follow the behavioural pattern of real ants in nature, which travel across various paths marking them with pheromone while seeking for food. ACO is used to solve many NP-hard problems including routing, assignment, and scheduling. We assume each service task is an ant and the algorithm sends the ants to search for Grid nodes.

BACKGROUND

Most studies in the field of resource allocation schemes aim at efficiently utillising the resources spread throughout a network. In most cases the problem is reduced to load balancing among specific nodes. Basic service task assignment strategies comprise the following (Balasubramanian, 2004): First, Round Robin, according to which the tasks are allocated to nodes by simply iterating through the nodes list. Second, Random, where the nodes to be assigned with the tasks are selected randomly. Third, Least Loaded in accordance with which the tasks are assigned to a specific node until a pre-specified threshold is reached. Thereafter, all subsequent requests are transferred to the node with the lowest load and the aforementioned steps are repeated. Fourth, Load Minimum, where the average load of the system is calculated. In case the load of a node is higher than the average node and of the least loaded node by a certain amount, all subsequent requests are transferred to the least loaded location.

According to the task farming paradigm (Andrews, 1991), a pool of tasks and one worker on each node of the system is considered. Each worker repeatedly claims a task from the pool, executes it and claims the next task. This way, the system load is efficiently distributed to the available resources. Considering dynamic, distributed controlled resource allocation, schemes in most cases follow three basic types (Agrawal, 1987): Sender-Initiated, where congested nodes (nodes where the load reaches a predefined threshold) take the initiative and probe other nodes in order to determine the most suitable node (e.g., least loaded node) for remote task execution, Receiver-Initiated, where lightly-loaded nodes search for work in a similar manner (probe other nodes in order to determine the node(s) that should be relieved from tasks e.g., the most loaded node), Symmetrically-Initiated, according to which both congested and lightly loaded nodes take the initiative. In (Lazowska 1986, Krueger 1988) the performance of these schemes is evaluated. The sender-initiated scheme is shown to perform better in light or moderate loaded systems, while the receiver-initiated paradigm is preferable at higher load conditions, under the assumption that the cost of transferring a task between the