International Journal of Web-Based Learning and Teaching Technologies
Volume 15 • Issue 1 • January-March 2020

# High Performance Fault Tolerant Resource Scheduling in Computational Grid Environment

Sukalyan Goswami, Institute of Engineering & Management, Kolkata, India
Kuntal Mukherjee, Birla Institute of Technology, Mesra, Lalpur Campus, Ranchi, India

## ABSTRACT

Virtual resources team up to create a computational grid, which is used in computation-intensive problem solving. A majority of these problems require high performance resources to compute and generate results, making grid computation another type of high performance computing. The optimization in computational grids relates to resource utilization which in turn is achieved by the proper distribution of loads among participating resources. This research takes up an adaptive resource ranking approach, and improves the effectiveness of NDFS algorithm by scheduling jobs in those ranked resources, thereby increasing the number of job deadlines met and service quality agreements met. Moreover, resource failure is taken care of by introducing a partial backup approach. The benchmark codes of Fast Fourier Transform and Matrix Multiplication are executed in a real test bed of a computational grid, set up by Globus Toolkit 5.2 for the justification of propositions made in this article.

## KEYWORDS

Computational Grid, Fault Tolerance, Resource Failure, Resource Management, Task Scheduling

## 1. INTRODUCTION

Task scheduling in appropriate resources and achieving balanced load in computational grid environment (Foster, Kesselman & Tuccke, 2001) are the two major research areas which need to be explored more. This is mainly because of heterogeneous nature of grid and technological requirement of computation-data separation. Computational capability of grid can be enormous if participating resources remain well coordinated by broker and effective task scheduling is highest prioritized work for achieving balanced load across grid.

Nearest Deadline First Scheduled (NDFS) algorithm (Goswami & Das, nee De Sarkar, 2014) solves the above mentioned problems after the efficiency of the algorithm has been improved by considering average load of each resource along with its current load for a pre-defined interval. Saaty's Analytic Hierarchy Process (AHP) (Saaty, 2008) has been adapted and improvised to rank the participating resources. Subsequent job scheduling and load balancing in grid are completed by NDFS, thus meeting the Service Quality Agreement (SQA) (Goswami & Das, nee De Sarkar, 2014). Another objective of this research work is to make NDFS more robust by taking care of sudden possible occurrences of resource failure in grid. Periodical runtime backup to next adaptively ranked resource ensures compliance of approved SQA, which was signed between broker and client. Even if resource fails, job need not be resubmitted in fault tolerant NDFS, submitted job resumes execution

DOI: 10.4018/IJWLTT.2020010104

from last backed up point, into next ranked resource. Hence, job is submitted by client only once, thereby reducing total execution time. This paper puts thrust on single and multiple job execution in real grid environment through execution of benchmark codes, namely, Fast Fourier Transform and Matrix Multiplication. The experimental results are obtained for demonstration of balanced workload in computational grid along with optimal solution for resource failure by creating test bed of computational grid through Globus Toolkit, and comparing performance with other algorithms in GridSim (Buyya & Murshed, 2002).

The organization of this paper is as follows: after discussing relevant works by different researchers on grid in section 2, improvised NDFS algorithm is presented in section 3. Section 4 depicts results of execution of benchmark codes in grid and discussion on the obtained results. The paper is concluded in section 5.

## 2. RELATED WORKS

There are many challenges associated with computational grid, like, scheduling of jobs in appropriate resources for subsequent execution, increasing the number of deadline meets for submitted jobs, balancing of workload in grid and fault tolerance. Important related works in this arena of research are summarized in this section.

The load balancing process must take into consideration the dynamic loads of participating resources of the grid (Abo Rizka & Rekaby, 2012). Job scheduling has to ensure higher number of deadlines meets, thereby improving the performance of the grid. In (Goswami & Das, 2016), an adaptive execution scheme has been proposed which ensures guaranteed performance with respect to service quality agreements.

The properties of general distributed system is quite different from that of computational grid environment. Moreover, the client-server framework of grid proposed by Michael Stal (Stal, 1995) does not solve the problem of load balancing among participating resources in grid. Various job scheduling schemes in grid have been described in (Kant Soni, Sharma & Kumar Mishra, 2010). X-Dimension binary tree data model (Abo Rizka & Rekaby, 2012) have limited success in workload balancing among grid resources.

This research work documents an approach to improve the performance of computational grid by reducing the the number of job deadline misses and overall makespan in grid. Job scheduling in appropriate resources, being a multi-criteria-decision-making (MCDM) process, AHP model (Saaty, 2008) of Saaty, supporting MCDM, is augmented in this research. The broker schedules submitted jobs in adaptively ranked resources, ranking being done with help of AHP model. This adaptive scheduling ensures performance improvement of grid by subsequently reducing number of deadline misses and workload balancing among resources. Deadline of individual job is chosen as the prioritization parameter for job scheduling in Nearest Deadline First Scheduled (NDFS) (Goswami & Das, 2015) algorithm, and GridSim (Buyya & Murshed, 2002) simulation results show that NDFS improves grid performance. Another milestone achieved is the number of submission by single client, one job needs to be submitted once only by client, need of re-submission by client is done away with, broker takes care of the possible re-submission in different resource in grid and monitors previously signed SQA between client and broker. Resource ranking of NDFS has been refined in this research by incorporating average load of the resource.

Resource failure creates possibility of deadline misses in grid. Multiple fault tolerant approaches in grid have been explored in (Goswami & Das, nee De Sarkar, 2014). Author's earlier research in (Goswami & Das, nee De Sarkar, 2014) dealt with resource failure by introducing resubmission approach which had incurred overhead. This research work improves the performance of NDFS by incorporating partial backup approach without diluting performance of the grid.