Chapter 29
A Hybrid Batch Mode Fault Tolerance Strategy in Desktop Grids

Geeta Rani
Chitkara University, India

Jyoti Bansal
Baba Farid College of Enggineering & Technology, Bhatinda, India

ABSTRACT

Desktop grids make use of unused resources of personal computers provided by volunteers to work as a huge processor and make them available to users that need them. The rate of heterogeneity, volatility, and unreliability is higher in case of a desktop grid in comparison to conventional systems. Therefore, the application of fault tolerance strategies becomes an inevitable requirement. In this article, a hybrid fault tolerance strategy is proposed which works in three phases. First, two phases deal with the task and resource scheduling in which appropriate scheduling decisions are taken in order to select the most suitable resource for a task. Even if any failure occurs, it is then recovered in the third phase by using rescheduling and checkpointing. The proposed strategy is compared against existing hybrid fault tolerance scheduling strategies and ensures a 100% success rate and processor utilization and outperforms by a factor of 3.5%, 0.4%, and 0.1% when turnaround time, throughput, and makespan, respectively, are taken into account

1. INTRODUCTION

There is an exponential growth in internet-connected desktop computers and their computing power (Satapathy et al., 2017). The study shows they will add up to billions in 2017. Moreover, CPU idleness is about 95% (Domingues et al., 2005) and 98% idleness are among Windows-based machines. On the other hand, there are many computationally intensive applications (Bhateja et al., 2018; Bhalla et al., 2018) which require this computational power in order to complete their tasks. Multinational companies

DOI: 10.4018/978-1-7998-5339-8.ch029

have already spent loads of money on personal computers, so there is no efficient alternative except using already existing personal computers for supercomputing applications. Desktop Grids makes use of unused resources of personal computers provided by volunteers to work as a huge processor and make them available to users that need them. Desktop Grid is a runtime aggregation of volatile, heterogeneous, non-dedicated personal computers so as to work like a Virtual Organization (VO) (Berman et al., 2003; Zheng & Veeravalli, 2009; Qureshi et al., 2011). Due to volatile, heterogeneous and widely distributed nature of Desktop grid, it is better suited for the execution of Bag-of-Task (Anglano et al., 2006) of applications (BOT). Bag-of-Task applications are loosely coupled parallel applications which are completely independent of each other.

Desktop grid resources are owned by volunteers and communication system is inherently unreliable. Furthermore, grid users are not experts and they can quit or join anytime that effects the execution of tasks. Thus, the rate of heterogeneity, volatility and reliability is higher in case of the desktop grid in comparison to traditional systems (Kondo et al., 2004; Foster et al., 2001). Therefore, the occurrence of faults is obvious which adversely affects the performance of the desktop grid. So, it becomes the requirement to have fault tolerance strategies to deal with different kinds of failures effectively. Fault tolerance is the property that ensures the delivery of expected services even in the presence of failures (Qureshi et al., 2011). It enables computation to continue even in case of failure. So in order to ensure the correct functioning of the system, fault tolerance becomes a requirement. But every fault tolerant strategy creates performance overhead, so it is required to use a fault tolerant strategy that is able to handle the majority of faults while maintaining system performance.

2. REVIEW OF LITERATURE

Desktop Grid resources are volatile, dynamic and diverse in nature. Faults are expected to occur due to the unpredictable environment of grid (Zheng & Veeravalli 2009; Lee et al., 2009; Sathya & Babu, 2010). Various fault handling strategies are designed to handle failures without affecting the quality of service (QoS). Fault handling strategies can be pro-active or post-active based upon whether faults are handled before or after job execution

Proactive fault tolerance strategies handle the failures before the execution of the task and scheduled with the expectation that task will never fail. Proactive strategies need information concerning i.e. grid tasks, resources, fault types, fault occurrence etc. for effective scheduling. Based on the available information, the decisions to handle the probable failures are taken.

- Q. Zheng et al. (2009) used directed acyclic graphs (DAGS) with communication delays to perform fault tolerance scheduling to handle processor faults. Tasks scheduling is performed in such a way that one processor failure does not affect others.
- An efficient fault-tolerant scheduling approach is proposed by P. Keerthika et al. (2012) in which tasks allocation is based upon fitness value and performance evaluation is performed in terms of makespan and hit rate.
- A fault tolerant scheduling system is proposed by Amoon et al. (2012) in which resources scheduling is based upon scheduling indicator (SI) which is a function of the failure rate and response time of grid resources