ISSN 1746-7659, England, UK
Journal of Information and Computing Science
Vol. 12, No. 4, 2017, *pp*.243-247

Reducing Checkpoint Overhead in Grid Environment

A. S. Faki*, R. G. Jimoh
Computer Science Department, Faculty of Communication and Information Science,
University of Ilorin, Ilorin-Nigeria
E-mail: faki.silas@binghamuni.edu.ng.
(Received June 19, 2017, accepted September 01, 2017)

Abstract. Grid Computing has become major player in super-computing community. But due to the diversity and disruptive nature of its resources, failure of jobs is not an exception. However, many researchers have come up with models that enhance jobs survivability. Popular among this model is checkpoint model which have the ability of saving already computed jobs on a stable secured storage. This model avoids re-computing of already computed jobs from the scratch in case of resources failure. But the time a job takes in checkpoinitng also becomes another task which adds overheads to computing resources thereby reducing the resources performance. In order not to add too many overheads to computing resources, the number of checkpoints must be minimized. This study proposed checkpoint interval models which is implemented based on fault index history of computing resources. Failed jobs are re-allocated from their last saved checkpoint using an exception handler. The study observed that arithmetic checkpoint model is better used when fault index of computing resources is high while geometric checkpoint model is better when fault index of resources is low.

Keywords: Arithmetic Checkpoint, Exception Handler, Fault Tolerance. Geometric Checkpoint.

1. Introduction

Grid has proven to be a great computational resource in computing circle. But due to its unstable and unpredictable nature of its resources, resources and job failure is a norm rather than exception. Systems and networks can fail, resources can be turned on and off and the introduction of more users can result in resource starvations which significantly affect quality of service (QoS). To maintain a tolerable QoS level for users, fault tolerance mechanisms are incorporated. Providing fault tolerance in a grid environment while on the other hand optimizing resources utilization and job execution time is a challenging task [1]. This is because fault tolerance measures increase computing overheads to grid resources. This study adopts checkpoint and recovery system as a measure to increase job life line thereby optimizing job completion time. The goal of this method is aimed at increasing the survivability of interrupted jobs by re-scheduling them at last save checkpoint and minimizing the number of checkpoints in order not to incur too many overheads. This model therefore, reduces the number of checkpoints and increase the survivability of jobs thereby optimization resources performance [2]

2. Literature Review

The desire of grid checkpoint service is to meet basic requirements which are: ability of software to exchange information among resources, ability of grid middleware and infrastructures to exchange and maintain vital information, and availability of checkpoint data to all resources [3]. But achieving interoperability in grid is difficult due complexity and unstable nature grid resources. The function of fault tolerance as opined by [4] is to preserve the delivery of expected service despite the presence of faulty processors and declining resources capability within the system. This means errors within grid system should be detected and corrected promptly. Permanent fault that could cause great havocs should be located and remove quickly. All these are to allow grid resources deliver a tangible and acceptable QoS. Due to technological advancement and increase in autonomous systems, many researchers are exploring ways to design and implement fault detective models that can predict and perform recovery from crash processors of computing resources. These models aim at improving resources performance and job survivability in the presence of failed processors, their effectiveness largely depends on tuning runtime parameters such as the checkpoint interval and number of replicas [5]. Fault tolerance schemes in grid can either be pro-active or post-active [6, 7]. In pro-active mechanism, the resources make a failure prediction process before jobs are scheduled with the hope that it will follow the prediction process. Whereas post-active mechanism handles the job failure after it has occurred. Most approaches applied to fault tolerance in grid environment are on post-active rather than the pro-active approaches [8].

According to [9], the most popular fault-tolerance model in use is checkpointing which involved periodic saving of snapshot of job progress on a stable storage device which can survive failures (hard disk). The information stored on the stable storage is called a checkpoint. Any time a processor or job crashes, the last saved checkpoint is used to restart the job or resource rather than from beginning. There exist varieties of checkpoint model but the popular ones could either be categorized as coordinated, uncoordinated and communication induced checkpointing [10].

The main advantage of checkpointing is that, it is a general technique which can be applied to any type of parallel applications. The time to take a snapshot or make a checkpoint is also a task or job which adds execution time overheads to a resource even when crashes have not occurred. This overhead is dependent on the frequency at which checkpoints are taken and in turn depends on the programmer or middleware designed.

To improve fault tolerance of grid system, [11] proposed hierarchical performance of checkpoint protocols grid computing which discussed protocols based on rollback recovery that was classified into two categories: checkpoint based rollback recovery and message logging protocols. However, the performance of protocols was observed to depend on the characteristics of the system, network and applications running. In situations where the computational intensity is low, the Algorithm Based on System checkpoint (ABSC) model is applicable. Meanwhile, if the computational intensity is high, Algorithm based on application Checkpoint (ABAC) model is more suitable though with production of slight overheads in fault free situations and very reliable in faulty situations [3]. Adaptive fault tolerant scheduling utilizes an adaptive number of job replicas according to the grid failure history. This technique composed of Adaptive Job Replication (AJR) and Backup Resource Selection (BRS) where AJR determines number of replicas according to selected resources.

To reduce checkpoint overheads, [12] proposed Optical Checkpoint Automation (OCA). The goal of this model is to make the most effective use of grid resources and also to improve throughput value in the mist of fault. This model focus on minimizing the effect of grid faults and reduces fault recovery time using optimal automation of checkpoint. To evaluate the performance of the model, fault index of resources were kept and analyzed and jobs were assigned to resources based on next sequence of pattern of failure. The failure patterns were predicted by using Hidden Markov Model (HMM) to assign checkpoint interval and also to provide automatic failure replica (context file of checkpoint) to the grid resource. The proposed OCA model provide a better performance as compare to adaptive algorithm though the model depends on previous history of computing resources which implies that checkpoint interval of new resources are unpredictable.

Checkpointing is the most common method to achieve fault tolerance. Though, there exist research issues on how improve its efficiency and overheads can be, [13] proposed a novel solution on how checkpoint can operate on parallel application in grid. The model allows checkpoint on applications at regions where there no inter-process communication thereby reducing the checkpoint overheads and checkpoint size. In [14], a novel technique to analyze the performance of checkpointing algorithms is proposed. The model was implemented in a suitable cloud environment with six service node with analysis made in terms of parallel jobs execution.

3. Arithmetic and Geometric Checkpoint Model (Agcm)

This model allows users submit jobs to the grid scheduler. Each submitted job is then assigned to a computing resource that matches the requirement by the scheduler as specified by user. As computing resources continue execution of job, statistics about each job and their resources are sent to the Grid Information Service (GIS) for storage. Resource capacity, checkpoint activities and current load are some of the information's that are stored in the GIS.