Resource Utilization and Failure Detection using Deep Learning in Grid Computing Environment

Aliyu, G., Thandekkattu, S.G; Longe, O.B., Onawola, H.J. & Badamasi, B.
School of Information Technology & Computing
American University of Nigeria
Yola, Adamawa State, Nigeria
E-mails: garba.aliyu@aun.edu.ng; george.thandekkattu@aun.edu.ng; olumide.longe@aun.edu.ng; hassan.onawola@aun.edu.ng; bukhari.badamasi@aun.edu.ng
Phones: +234806810107088; +2348050472729; +2348160900893; +2348167730044; +2348036992294

ABSTRACT

The grid has resources distributed into different geographical locations to solve complex user's tasks (jobs) under heterogeneous environment. These resources are likely to fail during execution which leads to loss of computation. Replication and Checkpointing are the two principal techniques used in realizing a fault tolerance system. Reliability is one of the key characteristics in the cloud that must be ensured. Therefore, despite the failure, the execution should continue until the computational goal is achieved. This research tends to proffer a solution using the concept of Deep Learning technique to predict the likelihood of any resource failure in the Grid. Moreover, this guides Scheduler to know ahead of time the available resources to be considered during jobs scheduling and aids in determining the optimal checkpoint interval, thereby increasing reliability, availability, and performance.

Keywords: Checkpointing; Replication; Reliability, Deep Learning, Scheduling.

iSTEAMS Multidisciplinary Conference Proceedings Reference Format
Aliyu, G., Thandekkattu, S.G; Longe, O.B., Onawola, H.J. & Badamasi, B. (2019): Resource Utilization and Failure Detection using Deep Learning in Grid Computing Environment. Proceedings of the 21st iSTEAMS Multidisciplinary GoingGlobal Conference, The Council for Scientific & Industrial Research -Institute for Scientific and Technological Information (CSIR-INSTI) Ghana. 14th – 16th November, 2019
Pp 225-230. www.isteams.net/goingglobal2019 - DOI Affix - https://doi.org/ 10.22624/AIMS/iSTEAMS-2019/V21N1P18

1. BACKGROUND TO THE STUDY

Grid computing is an architecture that connects computer resources from different geographical areas to achieve the principal computational objectives. In grid computing, the computers interconnected together alongside scheduling algorithms to execute user's job within the shortest possible time (Aliyu et al. 2017). These resources are distributed into different locations with different specifications and processing capability and act or function as a supercomputer. Grid computing is one of the emergences that tends to attract more organizations due to its reliability and availability of resources under a heterogeneous environment to get their complicated computational problems solved. However, these resources are likely to fail to due to some unavoidable factors such as power failure, network fault, resource fault (Tanya, Nancy, and Dinesh 2014), system crash among others, this leads to an increase in execution time thereby affect the performance. The concept of fault tolerance has been employed by many researchers to overcome the issues of resource failure within a grid computing environment (Aliyu et al. 2017), example of such is the used of Checkpointing and Replication.

Checkpointing tends to keep the state of the current execution within a time interval and in the case of any failure, there would be no significant loss of computation as the execution restarted from the last checkpointed state. Replication is a typical strategy utilized in a grid computing environment, where fundamental data is distributed and stored in various regions, with the goal that a client can get to the data from a locality in his general vicinity (Dayyani and Khayyambashi 2013). To improve the reliability and performance, a deep machine learning technique can be employed in the area of grid computing thereby prior to scheduling, a scheduler is up to date on the guaranteed resources that are likely not to fail. Those could be strictly considered in the course of scheduling. Moreover, the deep learning could be made to work in real time, whereby a plan on which resource is to take over the execution from the anticipated failed resource would uninterruptedly migrated and assigned to continue with the processing. This may abate the problem of checkpointing overhead. However, this research is not to replace the checkpointing but to combine it with deep machine learning technique to improve performance and reliability.

2. LITERATURE REVIEW

Researchers have focused on many techniques such as fault tolerance to improve the performance and reliability of grid computing. Among which are checkpointing and replication. Each of which has its own advantages and some drawbacks. Checkpointing has a problem of overhead as how frequent checkpoints can be introduced and at what interval. Frequent checkpoints lead to an overhead while large interval lead to loss of some significant parts of computation. State of the art works related to this proposed research have been reviewed.

In a study conducted by Aliyu *et al.* (2017), the authors proposed an architecture and checkpointing technique that tends to curtail the challenges of checkpointing overhead in order to increase response time and thereby saving processed information from loss. The research tried to come up with the strategy on how interval a checkpoint can be introduced to avoid excessive checkpoints and low checkpoint interval. This always guaranteed availability and reliability. The fact that excessive checkpoints are as well proposed to be addressed, combining with replication may improve availability but there will be a cost to pay for adding a replica. Fault Manager component of the architecture handles and track down any resource that failed or succeed and communicate to the Fault Index to update a status table accordingly. Nevertheless, employing deep learning technique can best detect and predict likely hood of any resource failure. Likewise, the tendency of resource to fail will be observed based upon its behaviour.

This can improve response time and makespan. Similarly, the rising measure of data and rapidly expanding the size of training models, particularly deep neural networks, essentially improve the learning accuracy, as well as notably elongate the training time (Peng et al. 2018). Peng et al. (2018) further uses a customized cluster scheduler called Optimus that targets high training performance and resource efficiency in the area of deep learning clusters, which reduces job training time upon online resource performance models. The convergence of prediction model during the model training, Optimus applies to establish performance models to correctly determine training speed in relation to resource allocation of each job. This is to dynamically allocate resource by placing deep learning tasks to minimize job completion time.

3. PROPOSED GRID SYSTEM ARCHITECTURE

This work built on the existing architecture proposed by Aliyu et al. (2017) as depicted in figure 1 below. Machine Learning introduced to handle the case of resource failure. The fact that these resource may likely to fail amid processing, the checkpoint handler component of the architecture keeps the current state of every resource, so in the case of any failure, the

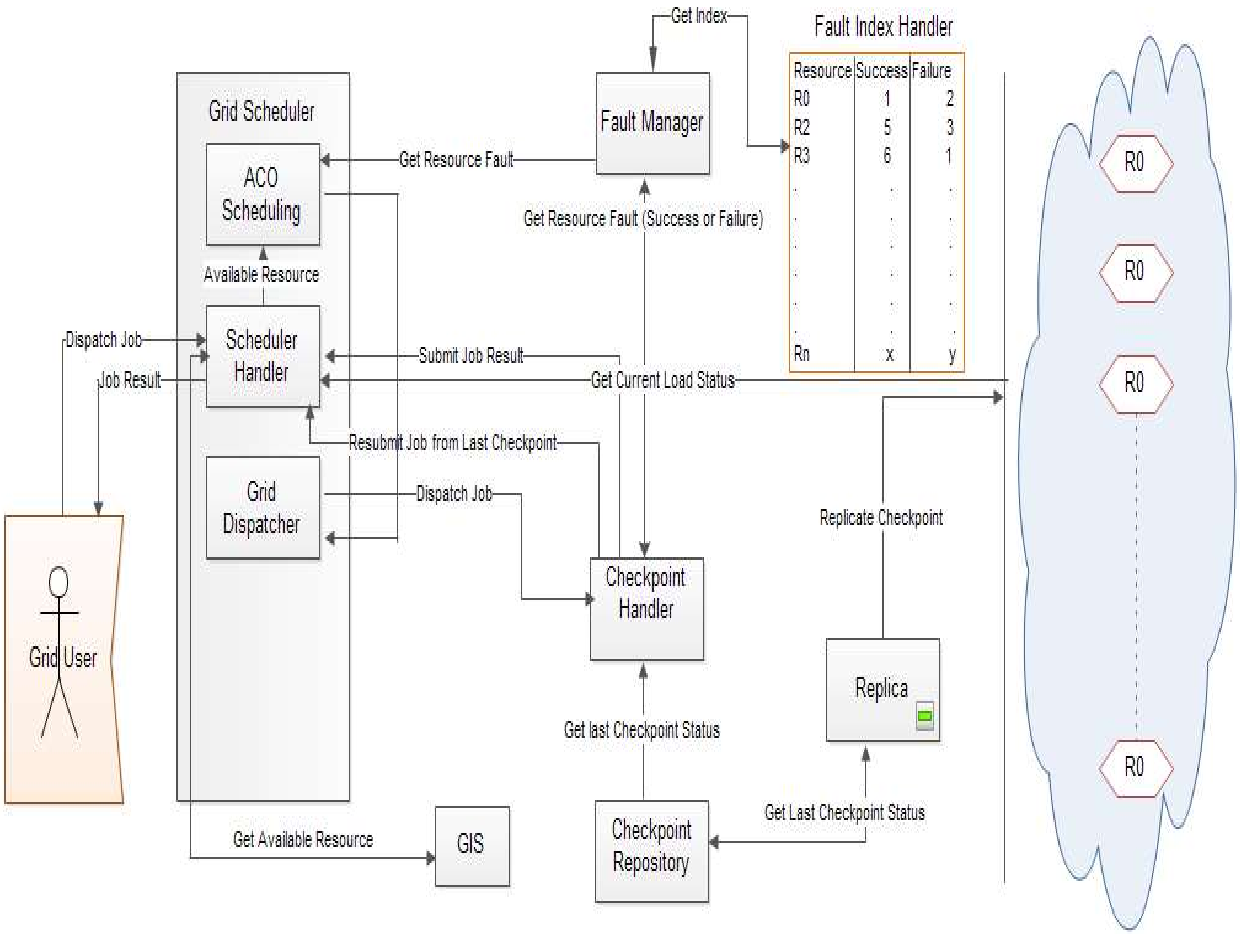

Figure 9: Existing Grid System Architecture

processing may not have to restart but rather continue from the last checkpointed state. The major drawback of a checkpoint is that having excessive checkpoints may exacerbate more overhead, while long checkpoint intervals may lead to loss of significant part of computation in the case of failure. Consider the diagram as illustrated in figure 2 below, suppose checkpoint1, checkpoint2, checkpoint3, checkpoints are very close to one another, at every checkpoint, the processing machine must have its state stored, that causes an overhead of storing (writing) and restoring (fetching) checkpoint files. This problem affects the performance of the grid system. In the situation whereby, for example, suppose there is no checkpoint between checkpoint1 and checkpoint3, and there is a failure, the execution has to restart from the last checkpoint (checkpoint1), thereby, losing all computational values in between them. This problem affects the reliability of the grid system.

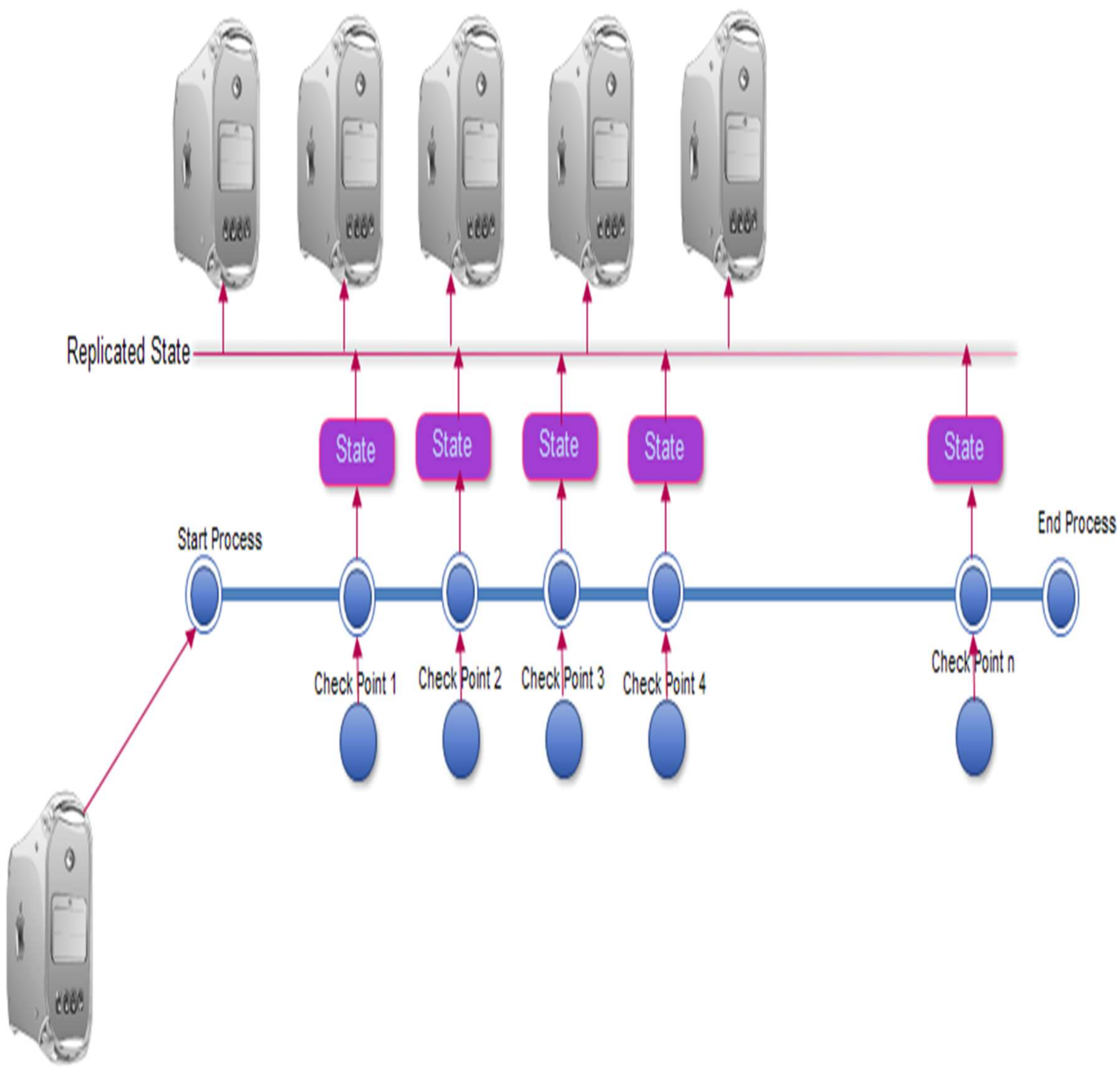

Figure 10: Checkpointing Technique (Aliyu 2016)

The proposed architecture as depicted in figure 3 below, is capable of addressing the major drawback of checkpoints as well as to curtail and avoid the frequently failed resources from job scheduling process before execution starts or amid execution. Scheduling in wide-scale computing settings such as Grid is difficult due to its dynamic and heterogeneity nature of it and the need for central control.

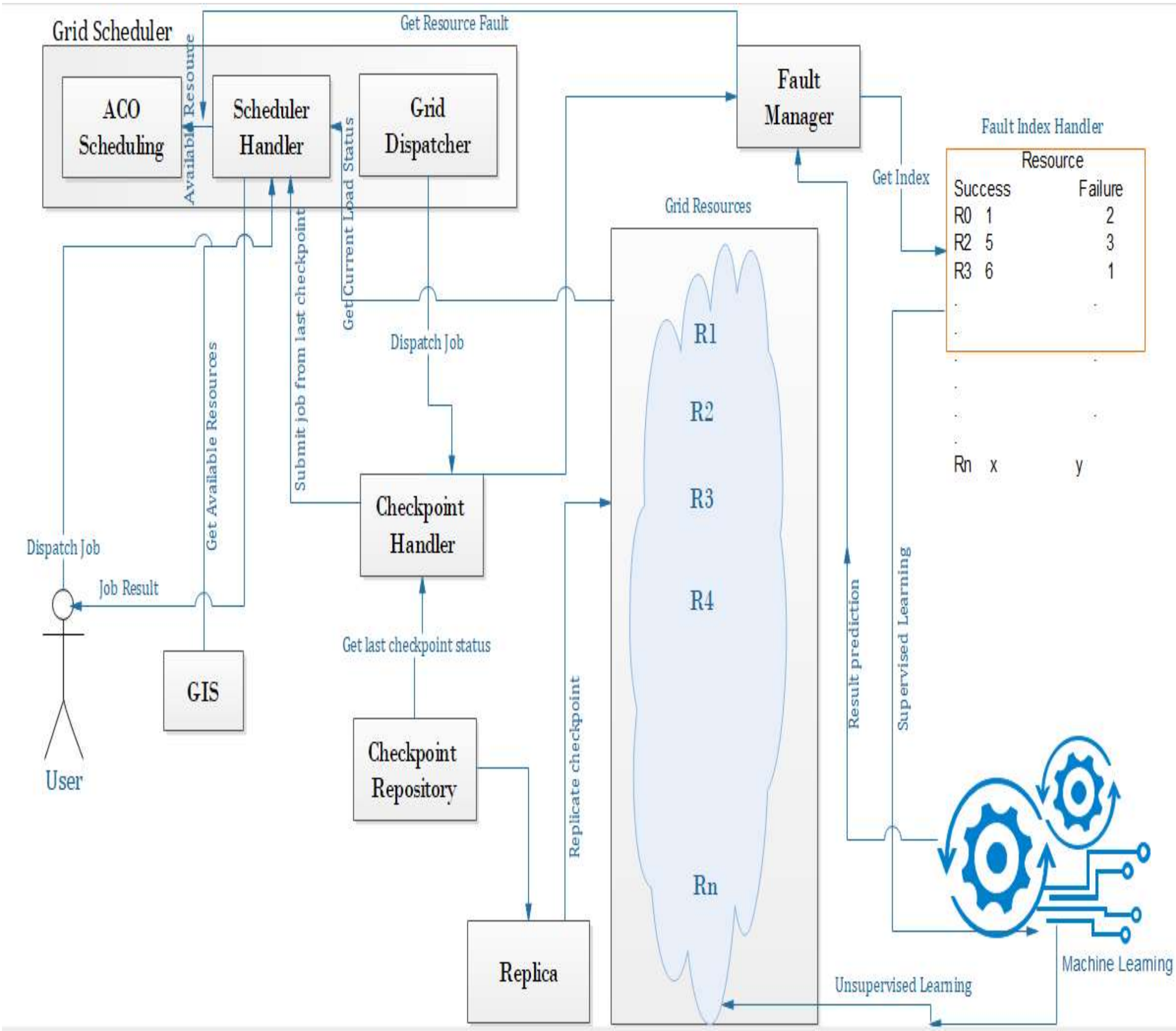

Figure 11: Proposed Grid System Architecture

In such circumstances, performance predictions act an essential function in rendering support information needed by a Grid scheduler (Li 2007). This work tends to use supervised and unsupervised machine learning to train different models. The supervised learning technique is to get data from Fault Index Handler (labelled data) about the success and failure of resources and feed into the model for learning. The unsupervised learning technique is to interact with resource pools to observe in real-time the behavior of any resource that may likely to fail. In whichever way, the Fault Manager receives the results of prediction sent by the machine learning component. The Fault Manager updates the Fault Index table and communicates to the Anti Colony Optimisation (ACO) Scheduling component to reschedule upon results received. At this stage, the rescheduling will be mainly for those reliable resources. The User in the architecture submits and receives job to the Scheduler Handler.

7. CONCLUSION/FUTURE WORKS

The implementation of this proposed research work is expected to reduce the overhead caused by checkpointing and improve performance and reliability of resources. However, there may be a price to pay as a result of incorporating machine learning during a training model. It is expected that machine learning will improve performance and reliability as compared with checkpointing with an overhead. The fact that the cost of the training model may affect performance and supervised and unsupervised learning may be at a high cost. In the future, detail analysis of supervised and unsupervised learning models can be carried out to compare and observe the most contributing cost and avoid.

REFERENCES

1. Aliyu, G. (2016). Development of an Enhanced Check Pointing Technique in Grid Computing Using Programmer Level Controls [Student]. Retrieved from Ahmadu Bello University, Zaria-Nigeria website: http://kubanni.abu.edu.ng/jspui/bitstream/123456789/8949/2/development%20of%20an%20enhanced%20checkpointing%20technique%20in%20grid%20computing%20using%20programmer%20level%20controls.pdf
2. Aliyu, G., Kana, D. A., Junaidu, B. S., & Aminu, M. B. (2017). Enhanced Checkpointing Technique In Grid Computing Using Programmer Level Controls. Yusuf Maitama Sule University, Kano Faculty of Science.
3. Dayyani, S., & Khayyambashi, M. R. (2013). A Comparative Study of Replication Techniques in Grid Computing Systems. 11(9), 10.
4. Li, H. (2007). Performance Evaluation in Grid Computing: A Modeling and Prediction Perspective. Seventh IEEE International Symposium on Cluster Computing and the Grid (CCGrid '07), 869–874. https://doi.org/10.1109/CCGRID.2007.84
5. Peng, Y., Bao, Y., Chen, Y., Wu, C., & Guo, C. (2018). Optimus: An efficient dynamic resource scheduler for deep learning clusters. Proceedings of the Thirteenth EuroSys Conference on - EuroSys '18, 1–14. https://doi.org/10.1145/3190508.3190517
6. Tanya, P., Nancy, & Dinesh, K. (2014, July). (PDF) Fault Tolerant ACO using Checkpoint in Grid Computing. http://dx.doi.org/10.5120/17223-7465