

Support Vector Machine Based Fault Tolerant Grid

Joshua Paul Winraj. A
PG Scholar, Hindustan University

S. Sathyalakshmi
Professor, Hindustan University

Abstract

Developing an effective and proper fault tolerant system has always been a challenge in the Grid Environment, because of the heterogeneous and dynamic nature of the resources. Many systems and techniques have been developed to help the Grid continue functioning with little or no decrease in the performance even after a node has failed. In this paper, to improve the reliability of the fault tolerant system a technique known as the SVM is used. According to various studies, the SVM has proved to be very effective in classification and prediction, and even used in Grid Scheduling. Prediction helps to improve the placement decisions of the grid. Initially, the SVM is used for predicting whether a node is about to fail in the near future or not, based on this the scheduler can choose some other node. Since electronic components are susceptible to failure, a node which has been allocated may fail. If a node fails, generally the task is migrated to some other node and is restarted from the beginning or is said to resume from the recent checkpoint. But some nodes can be recovered from the failure very soon and in this case the task can be resumed on the same node without migrating. This technique is known as Local Node Fault Recovery (LNFR). In some other cases, the recovery takes too long compared to the migration time and in those cases the task migration is preferred to LNFR.

1. Introduction

Grid computing is a technology advancement that brought about a revolution in the sharing of resources. There are basically two kinds of Grid: computational grid and data grid. Scheduling is a part of the grid computing paradigm that is most essential in the functioning of any kind of grid, whether it is a data grid or a computational grid. Scheduling in a grid is done by making use of applications known as Grid Schedulers. Basically, a Grid scheduler receives

applications from Grid users, selects feasible resources for these applications according to acquired information from the Grid Information Service module, and finally generates application-to-resource mappings, based on certain objective functions and predicted resource performance. Unlike other traditional schedulers made use of in distributed systems, Grid schedulers usually cannot control Grid resources directly, but work as brokers or agents.

Grid Scheduling is most of all dynamic in nature i.e., the resource can be added or removed anytime from the common resource pool. This raises a concern for the availability of the resources in the grid. Also electronic components are susceptible to failure and can fail even during the execution of a subtask by a resource. In this case proper mechanism to continue the execution of the subtask must be installed in place to enable the task to be completed. Check pointing is very useful in such cases. A task may take several days to complete its execution and after completing a major part of it if the resource fails then it is painful to execute the task from the beginning. Check pointing is used to store the state of the task and related details, so that if a resource fails the task can be migrated to another resource and can resume execution from the last checkpoint, by this the task need not be restarted from the beginning. But the migration time and cost have to be managed here.

Predicting resource failure is helpful to the Scheduler in allocating resources. This kind of prediction of resources saves the scheduler a great deal of trouble. When a resource has the possibility of failing in the near future, the scheduler need not allocate any tasks to it. Some common techniques used in the resource failure prediction are SMART tools, Intelligent Platform Management Interface (IPMI) and Support Vector Machine (SVM) techniques. All these are proved to be efficient and the SVM technique especially has proved to be much efficient in predicting resource failures. In some cases, the SVM makes use of the system log files for prediction. The idea that is made use of here is the fact that a sequence of the log files is a prelude to the

occurrence of an event in many cases. Since the system log messages are generated for every event happening in the system it is difficult to trace the events, but a sub-sequence of the log messages along with the help of the tag values will be of great help in determining the event that is about to take place.

In the event of a failure, the completion of the task has to be ensured to prove the dependability of the grid. Some common alternatives are the task is migrated to another node where it is said to resume from the nearest checkpoint if check pointing is used otherwise the task is executed from the beginning on the migrated node. The time and cost of migration are factors to be considered here.

Another alternative in the event of failure is to check whether the node failure is a recoverable or unrecoverable one. Recoverable failures are those in which the node can be recovered and unrecoverable failure is one in which the node cannot be recovered and the task has to be migrated to continue execution. When a failure is recoverable the task is made to resume execution on the same node itself instead of migrating since the time and cost of migration in most cases may be high. This technique is known as Local Node Fault Recovery (LNFR).

2. Related Work

This section briefs some of the work done in this area in a varied combination. Both reliability of the Grid system and the prediction of the resources for unavailability and failure have much work done yet there is room for considerable improvement

Suchang et al[2] came up with a technique to improve reliability in the Grid systems. In their paper a technique known as Local Node Fault Recovery (LNFR) to solve the reliability problem in the Grid system and a modified Ant Colony Optimization (ACO) Algorithm to solve the task scheduling problem were introduced. Usually when a node failure occurs, the task is migrated to another node which is available. This may involve considerable migration time which might be a burden for larger tasks, but in the LNFR the node is check whether it is a recoverable or unrecoverable one. If it is found to be a recoverable one then and the then the task is made to resume execution on the same node. Hence, saving much migration time and cost.

Another approach to improve the reliability by predicting failure is given by Errin et al [3]. In this paper a SVM approach which makes use of the system log messages to predict failure in the system

resources is suggested. The SVM approach classifies the resources into one of the two states: a fail state or non-fail state and thereby helping in allocation of the resources in accordance to it. Whereas a single log message may not be of much help, but a sequence of log messages may be a prelude to the occurrence of a particular event. The frequency of occurrence of these events is then sent to SVM classifier which classifies the events into the above mentioned states. This method has proved to be very effective in prediction of failure events.

Predicting the resource failure and unavailability has greater value when the size of the Grid increases. This is displayed much eminently in the paper put up by Brent et al[4]. The main idea here is to provide information to the grid schedulers for better placement decisions which in turn help in improving the reliability of the systems in the grid. Predicting the failure of the resources is done by making use of the advancement in the processor technology and the predicting the unavailability of the resources is done by making use of the past behavior of the resources. Also the prediction helps in better check-pointing and failure aware scheduling.

Yong et al[5] proposes a machine learning approach based on the SVM. The system is made up of two components: a SVM Scheduler and a dynamic learner. A multiclass SVM is used by the scheduler to map task to machine based on the current load on the machines and the tasks waiting to be dispatched. A dynamic learning system which incorporates new knowledge into the scheduler is also designed to support dynamic adaptation of the scheduler.

3. Proposed System Architecture

The system flow diagram is given below. The SVM technique is used in predicting before assigning the tasks to the resources as well as in classifying the node failure as recoverable or non-recoverable node. If the node is recoverable then the recoverable time is compared with the task migration time. If the recoverable time is less than the task migration time then the LNFR technique is used to recover the local node and resume task execution on the same node else the task is migrated.

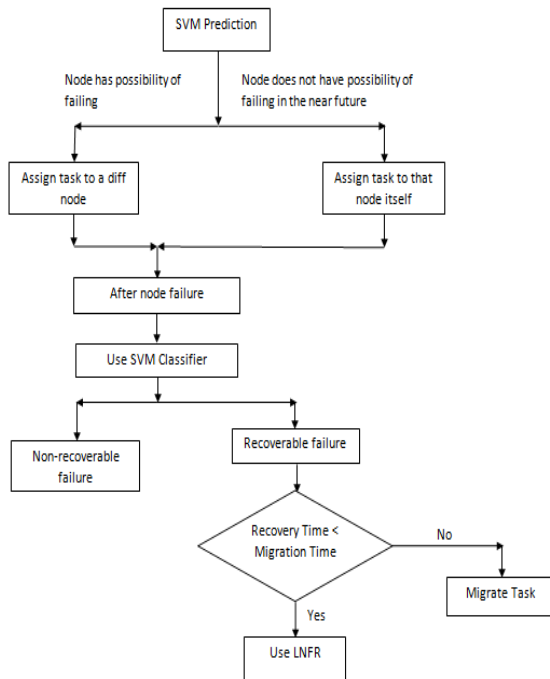


Figure 1. System Flow diagram

4. Conclusion and future work

The outcome of this paper is to increase the reliability of a fault tolerant grid. The SVM technique is used in combination with the LNFR technique to improve the reliability of the systems in the Grid Environment. In this combination the SVM is used as a predictor and classifier, thereby contributing to the enhancement of reliability in the Grid and the LNFR is used in case of failure. The SVM prediction saves time by predicting the susceptible failure of the resources so that the scheduler will not waste time in allocating task to such resources or helps in allocating less critical task to such resources which have the possibility of failing in the near future. The SVM classification helps the scheduler to classify the failures into recoverable and unrecoverable ones. In case a recoverable failure is encountered whose recoverable time is less than the task migration time then LNFR is used else task migration is performed.

References

- [1] Fangpeng Dong and Selim G. Akl. (2006), "Scheduling Algorithms for Grid Computing: State of the Art and Open Problems". Technical Report No. 2006-504, pp. 1-55.
- [2] Suchang Guo, Hong-Zhong Huang, Zhonglai Wang, and Min Xie (2011), "Grid Service Reliability Modeling and Optimal Task Scheduling Considering Fault

Recovery", IEEE Transactions on Reliability, Vol. 60, No. 1, pp. 263-274.

[3] Errin W. Fulp, Glenn A. Fink, Jereme N. Haack (2008), "Predicting Computer System Failures Using Support Vector Machines". Proceedings of the First USENIX conference on Analysis of system.

[4] Brent Rood, John Paul Walters, Vipin Chaudhary, and Michael J. Lewis (2007), "Failure Prediction and Scalable Checkpointing for Reliable Large-Scale Grid Computing". The 16th IEEE International Symposium on High Performance Distributed Computing.

[5] Yong-won Park, Kenan Casey and Sanjeev Baskiyar (2010), "A Novel Adaptive Support Vector Machine Based Task Scheduling". Proceedings of Parallel and distributed computing and networks.