International Journal of Computer Applications (0975 – 8887)
Volume 63– No.20, February 2013

Resource Consumption Framework for Fault Diagnosis in Cloud

Chitra.B,
P.G. Scholar,
Dept. of CSE,
Angel College of Engineering and Technology, Tirupur

Selvi.S,
Assistant Professor,
Dept. of CSE,
Angel College of Engineering and Technology, Tirupur

T.Rajendran, PhD.
Professor & Dean,
Dept. of CSE & IT,
Angel College of Engineering and Technology, Tirupur

ABSTRACT

In the heterogeneous parallel and distributed computing environments like cloud there were many related approaches proposed for fault tolerant execution of workflows. Most of the earlier works involved does not depend on failure prediction of the resources that is really hard to achieve with the tracing of historic failure data over years of the desired environment. In this paper, to solve the software fault prediction, unavailability of the resources and monitoring problems we propose a failure prediction model that involves two different methods. In order to predict the failures at the nodes we propose a method using Intelligent Platform Management Interface (IPMI), that monitor the failure at nodes and provide the respective data that is useful for determining likely imminent failures. The other method is to predict the Unavailability of the resources from past behavior that generates some initial results that indicate that nodes are different from one another and their failure is somewhat predictable and monitoring is performed which intimates about the failure.

General Terms
Fault Tolerance, Monitoring.

Keywords
Failure prediction, IPMI, Checkpoint, Task Replication.

1. INTRODUCTION
To improve the fault tolerance of scientific workflow applications, which emerged in the last decade as one of the most successful paradigms for programming e-science applications in highly distributed environments such as Grids and Clouds.

Currently, there are two fundamental and widely recognized techniques to support fault tolerance in distributed environments: Resubmission and replication. Resubmission is the technique which is to re-execute a task after a failure which can significantly delay the overall completion time in case of multiple repeating failures. Replication submits several copies of the same task in parallel on multiple resources which suffers from potentially large resource consumption. To find a

compromise balance between these two complementary techniques, we took a algorithm called Resubmission Impact (RI) that tries to establish a metric describing the impact of resubmitting a task to the overall execution time of a workflow application, and to adjust the replication size of each task accordingly.

The number of dynamic resources in the cloud system increases continuously, so fault tolerance techniques the resubmission and replication becomes a critical property for applications running on these resources. However, in traditional implementations, when a failure occurs, the whole application is shutdown and has to be restarted from the beginning. A technique to avoid restarting of the application from the beginning is rollback recovery which is based on the concept of checkpoint. Checkpoint mechanism is used to reduce the limitations imposed by the high volatility of resources. It periodically saves the application's state to stable storage. So, whenever a failure interrupts a volunteer computation, the application can be resumed from the last stable checkpoint. Checkpoint-recovery techniques make it possible for the job to resume execution from the last checkpoint instead of restarting from the beginning, whenever a failure occurs.

Over provisioning techniques replicate a job in more than one resource to increase the probability of successful execution. Although these techniques address the reliability challenges to some extent, no large-scale study has been done on how effective they are when coupled with scheduling. The Resubmission Impact (RI) mentioned before if a failure occurs it will recover from the beginning so in this paper we propose a checkpoint mechanism that is a failure prediction model which involve two methods Predicting Node failure using IPMI and Predicting Unavailability from Past Behavior.

The paper is organized as follows. Section 2 discusses related work, Section 3 introduces the concepts of IPMI, followed by Section 4 the proposed system is explained which includes the node failure prediction and in Section 5 conclusion is given.

2. RELATED WORK
Considering the faults in scientific workflows the existing techniques available are by Jia Yu,Rajkumar Buyya and Chen Khong Tham [1]minimizes the cost of execution while meeting the deadline.But on the other hand it uses run-time rescheduling to handle service agreement violations and problem occurs while scheduling dynamic pricing. The work [2] not finds traces that include mostly production workflows submitted by real users. L Guo, A S McGough, A Akram, D Colling, J Martyniak, M Krznaric [3] provide a level of QoS through resource selection from priority information along with the use of advance reservation but which will not allow to dynamically change the execution of the workflow once deployed to the engine.

The advance reservation can have a major impact on execution time and can increase considerably predictability of a Grid environment and lacks in concentrating dynamic reservation algorithms for heavily loaded Grid environments [4]. Jayadivya S K Jaya Nirmala S Mary Sair Bhanu S [5] proposed an approach with a prioritization of tasks that helps to meet the deadline and reduces resource wastage along with providing fault tolerance for the workflow system. And does not consider the failures like data center shutdowns, network failures that may also added to faults. The technique of query planning is proposed in [6] applications that incorporate kernel-level and user-level checkpointing could receive signals generated by chip monitoring facilities that recognize node failure is imminent, alleviating the need to take periodic checkpoints.

Ozan Sonmez, Nezih Yigitbase, Alexandru Iosup [7] proposed methods cannot improve and adapt to computational grids where the historical runtime and wait time data considered are non-stationary. The approach mentioned in [9] developed to extend detection to lower levels, such as hardware and job execution faults. The performance analysis of list scheduling [10] is done and the total execution of the program is minimized and even though it does not degrade but not improve the worse case and average case performance.

Resubmission Impact (RI) that tries to establish a metric describing the impact of resubmitting a task to the overall execution time of a workflow application, and to adjust the replication size of each task accordingly. This method (RI) has the advantage of applicability to new and unknown environments, however, it often leads to unnecessarily large resource consumption and to large differences between the expected execution time (as promised to the end user) and the real execution time. The RI heuristic is able to schedule workflows with a high success rate while consuming a reasonably low amount of resources. However, RI will schedule the full workflow before it is executed, it cannot react to unexpected periods of high failures during execution and is therefore unable to adjust its replication size accordingly. This can lead to situations where the replication can fail and many tasks have to rely on resubmission. Consequently, the workflow makespan can exhibit large deviations from the expected value, degrading the QoS for the end user.

There is still an extension to the RI heuristic that is able to provide the end user with a realistic estimation of the execution time in the form of a soft deadline. The workflow enactment responsible for submitting the tasks and transferring data according to the control flow and data flow dependencies is enhanced with a monitoring step after the completion of each task, followed by a heuristic that adjusts the RI values and reschedules the workflow remainder if the soft deadline is likely to be missed. The workflow enactment strategy includes two factors with the RI heuristic maximum reschedule cycles defines the maximum number of rescheduling actions to be performed for a given workflow, and rescheduling factor which represents a percentage how much the real execution time is allowed to exceed the initial schedule before rescheduling is performed and the rescheduling threshold that will trigger a first rescheduling action if the real execution time exceeds the expected one by the rescheduling factor.

The other extension of Resubmission Heuristic is the rescheduling heuristic invoked if the enactment engine discovers that the real workflow makespan is too far behind the scheduled workflow execution plan, which implies a heightened probability of missing the soft deadline presented to the user. The role of the rescheduling heuristic is to decrease the probability of violating soft deadlines by adjusting the amount of replication. It brings two advantages over existing methods: 1) it reduces the resource consumption, and 2) it offers improved QoS by meeting soft deadlines.

Still the existing approach does not depend on failure prediction of the resources that is really hard to achieve with the tracing of historic failure data over years of the desired environment. To solve the software fault prediction and unavailability of the resources we propose the Node failure prediction and to predict the unavailability of the resources from the past behavior.

3. INTRODUCTION TO IPMI

3.1 IPMI Specifications

The Intelligent Platform Management Interface (IPMI) specification which defines about the interfaces to computer hardware, and is used for managing the computer system with the capability of monitoring the whole system. The corresponding states of the system can be monitored by the system administrator in order to manage the computer system effectively. The IPMI which is designed by Intel which is not dependent on the operating system and it is not compulsory to be the machine we are monitoring to be powered on. There is a possibility of remote accessing to the respective monitoring system even if failure occurs in the system. It is also possible with this type of system specification that it can find the cause of the failure and helps in taking suitable actions which is help for system recovery.

The management and monitoring in a serial line, a local area network, or a serial over LAN that is connected to the monitoring client is supported by this IPMI specification. The system can get the health and status itself by querying to this client even though the power is on or off. There is a central controller called Base-board Management Controller (BMC) and peripheral controllers or sensors. This controllers communicate through a special bus called Intelligent Platform Management Bus/bridge.

3.2 IPMI components

A subsystem in an IPMI which consists of a main controller called the Baseboard Management Controller (BMC) as mentioned earlier and some of the controllers which is used for management is scattered among various system modules which can be called as satellite controllers. There is a Intelligent Platform Management Bus/Bridge which is an interface used for satellite controllers within the same platform connect to the platform. It is an improved implementation of Inter-Integrated Circuit. The BMC connects to satellite controllers or another BMC in another chassis via the Intelligent Platform Management Controller (IPMC) bus or bridge. It may be managed with the Remote Management Control Protocol (RMCP), a specialized wire protocol defined by this specification. RMCP+ (a UDP-based protocol with stronger authentication than RMCP) is used for IPMI over LAN.

Baseboard Management Controller is a specialized microcontroller embedded on the motherboard of a computer, generally a server. The BMC manages the interface between system management software and platform hardware.

Different types of sensors built into the computer system report to the BMC on parameters such as temperature, cooling fan speeds, power status, operating system (OS) status, etc. The BMC monitors the sensors and can send alerts to a system administrator via the network if any of the parameters do not stay within preset limits, indicating a potential failure of the system. The administrator can also remotely communicate with the BMC to take some corrective action such as resetting or power cycling the system to get a hung OS running again. These abilities save on the total cost of ownership of a system.

4. PROPOSED SYSTEM

In our work, to solve the software fault prediction and unavailability of the resources we propose the Predicting the Node failure using Intelligent Platform Interface (IPMI) and to predict the unavailability of the resources from the past behavior. Section A explains about how the failure of a particular node can be predicted and Section B to identify the unavailability of the resources with five different states as follows.

4.1. Predicting Node Failure Using IPMI

There are more chances for node failures when users use more resources for more than a limited amount of time or for a long time. There will be more differences in the features of the node failure when multiple different resources residing in the same involved collection. Many fault tolerant solutions introduced currently will lack in supporting more dynamic environments due to two significant reasons. One is current solutions for check pointing always needs storage centrally and therefore it is not scalable. Second is that schedulers involved in cloud do not take into account the failure nature in making decisions in failure.

In order to reduce the total count of checkpoint data used and to support the monitoring and maintenance as whole for a large computational cluster an accurate still a simple tool must monitor for the failed nodes and supply the data useful for evaluating the likely imminent or forthcoming failures. This monitoring tool would allow the application to take decisions at runtime whether a failure is expected to occur and simultaneously test whether a checkpoint is necessary at that point.

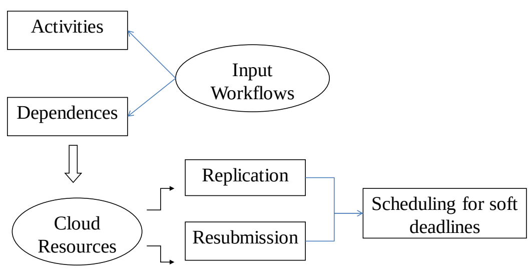

Figure 1. Architectural Diagram

From Figure 1 the architectural diagram, the Input workflows which include the activities, dependences and work and the workflow processed in cloud. The cloud Resources can be CPU, memory and processor. Depending upon the Resubmission impact the workflows will be resubmitted to do the task or the copies of the data will be taken from the replicated copy. Users always want to know about the execution time before they start executed which is referred as soft deadlines here and in our work while recovering from the failure this deadline should also be considered and the software fault prediction to find the node failures and to predict the unavailability of the resources of crossing the deadline.

4.2. Predicting Unavailability from Past Behaviour

Some high performance distributed computing like cloud environments, there is a necessary and support for the knowledge of the unavailability features of different component resources. With the help of the past behavior of resources in dynamic environments like cloud we can distinguish and categorize the resources into classes and can predict their future availability. These assumptions and predictions can assist the cloud based schedulers to improve the performance of applications and utilize the cloud resources efficiently. When the applications are running and if the chosen resource fails before the completion of the application then the significance of properly predicted the node diminishes with the lowest future load, cloud related schedulers would also consider the future accessibility of the candidate resources. We just generate some preliminary results that point that nodes fail differently from each other and the failure occurred is somewhat knowable.

In our work we find their movement with five various states and the resources availability can be tracked through it. The resources can also be classified depending upon the behavior of each one in means of these states over a stipulated amount of time. A predictor is also available which is useful for checking the probability of the next state being attained by a respective resource with more importantly has a great improvement in the accurateness over the other predictors.

5. SIMULATION RESULTS

To evaluate the performance of our fault tolerant system we have simulated the different structures and workflows by using CloudSim 3.0.

CloudSim is a framework for simulation and modeling of cloud computing environments mainly used for resource allocation and scheduling. It includes data centers, Virtual machines, brokers, cloudlets and hosts.

Here in the below table we have taken 5 virtual machines and its corresponding parameters

Table 1. Virtual machine with its parameters

VMid	Mips	CPU	RAM	BW
0	350	1	512	1250
1	350	2	2048	700
2	500	3	1024	1500
3	400	1	3056	1000
4	350	3	512	800

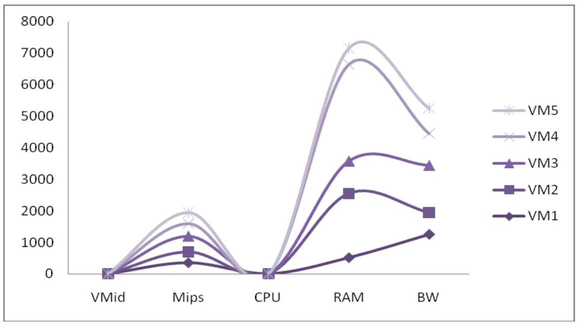

Figure 2. Analysis of virtual parameters

Here 20 workflows are considered with 5 Virtual Machines and further depending upon the prediction of earlier completion of tasks the workflows are scheduled subsequently.

6. CONCLUSION

We proposed in this paper the software failure prediction to increase fault tolerance of the activities involved in parallel and distributed environments such as clouds in the lack of failure models. The technique is based on a combination of task replication and task resubmission using a RI metric that describes the impact of task resubmission on the overall workflow makespan and we have a proposed a checkpoint mechanism that is a failure prediction model which involve Predicting Node failure using IPMI.

The prediction of unavailability of the resources is also possible here with the periodic monitoring of the past behaviors of the resources which are consumed here. This prediction which also involve the random predictor which helps in assuming the future availability of the resources and gains the advantage that the providers will be knowing the presence of resources that are to be consumed by the users which enhances the monitoring by our work. In future, we have an idea of applying effective monitoring algorithms to further improve the performance of task scheduling.

7. REFERENCES

[1] J.Yu,T.Buyya and chen Khong Tham, "*Qos- Based Scheduling of Workflow Application on Service Grids*," Proc.IEEE First Int'l Conference e-Science GridComputing (eScience'05),Jan.2005

[2] S. Ostermann, R. Prodan, T. Fahringer, A. Iosup, and D. Epema, "*A Trace-Based Investigation of the Characteristics of Grid Workflows*," From Grids to Service and Pervasive Computing, pp. 191-203, Springer, http://www.springerlink.com/content/x21m42878m456338/fulltext.pdf, Aug. 2008

[3] L.Guo, A.McGough,A.Akram,D.Colling, and J.Martyniak, "*Qos for Service Based Workflow on Grid*," Proc.Conf.UK e-Science 2007 All Hands Meeting, January 2007.

[4] M. Wieczorek, M. Siddiqui, A. Villazon, R. Prodan, and T. Fahringer, "*Applying Advance Reservation to Increase Predictability of Workflow Execution on the Grid*," Proc. IEEE Second Int'l Conf. e-Science and Grid Computing (E-SCIENCE '06), 2006.

[5] K. Plankensteiner, R. Prodan, T. Fahringer, A. Kertesz, and P.Kacsuk, "*Fault-Tolerant Behavior in State-of-the-Art Grid Worklow Management Systems*," Technical Report TR-0091, Inst. On Grid Information, Resource and Worklow Monitoring Services,CoreGRID—Network of Excellence, Oct. 2007.

[6] Y.Zhang, D. Wong, and W. Zheng, "*User-level Checkpoint and Recovery for LAM/MPI*," SIGOPS Oper. Syst. Rev., vol. 39, no. 3, pp. 72–81, 2005.

[7] J. Yu and R. Buyya, "*A Taxonomy of Scientific Workflow Systems for Grid Computing*," ACM SIGMOD Record, vol. 34, no. 3, pp. 44- 49, 2005

[8] Seoko Son and Kwang Mong Sim ,"*A price and time slot Negotiation mechanism for Cloud Service Reservations*" in IEEE Transactions on Systems,Man, Cybernetics, June 2012

[9] K.M.Sim and B.Shi, "*Concurrent negotiation and coordination for Grid resource coallocation*" IEEE Trans.Syst.,Man,Cybern.B,Cybern.,vol.40,no.3, pp.753-766,May2010

[10] G.Kandaswamy, A. Mandal, and D.A. Reed, "*Fault Tolerance and Recovery of Scientific Workflows on Computational Grids*," Proc. IEEE Eighth Int'l Symp. Cluster Computing and the Grid (CGDRID '08), pp. 777-782, 2008.

[11] Bernd Grobauer, Siemens CERT Tobias Walloschek, Siemens IT Solutions and Services Elmar Stöcker, Siemens IT Solutions and Services, "*Understanding cloud computing vulnerabilities*", IEEE Security and Privacy.

[12] Tharam Dillon, Chen Wu and Elizabeth Chang, "*Cloud Computing: Issues and Challenges*", 2010 24th IEEE International Conference on Advanced Information Networking and Applications.

AUTHOR'S PROFILE

The Author, Chitra,B is a final year student doing Master of Engineering in Computer Science at Angel College of Engineering and Technology,Tirupur and received Bachelor of Engineering degree in Information Technology from M.Kumarasamy College of Engineering,Karur in the year 2011. Her area of interest is Cloud Computing. She has presented over 5 papers in National and International Conferences. She has also published a paper in an International Journal in the domain Data Mining.

The Author, Mrs.S.Selvi finished her UG in Kongu Engineering College and her P.G in Velalar College of Engineering and Technology. She is currently pursuing Ph.D in the area of Cloud Computing under Anna University, Chennai. She has about 5 years of teaching experience in Engineering Colleges.She started her career as a Lecturer in Sasurie College of Engineering from August 2005 to May 2009 and she is currently working as an Assistant Professor in Angel College of Engineering & Technology. In her academic career, she has produced 100% results in 6 subjects. She has presented 6 papers in National Conferences and 2 papers in International Conferences. Her 2 papers are yet to be published in Journals.

The Author, Dr. T.Rajendran completed his PhD degree in 2012 at Anna University, Chennai in the Department of Information and Communication Engineering Tamilnadu, India. Now he is working as a Dean for Department of CSE & IT at Angel College of Engineering and Technology, Tirupur. His research interest includes Distributed Systems, Web Services, Network Security and Web Technology. He is a life member of ISTE & CSI. He has published more than 51 articles in International/ National Journals/Conferences. He has visited Dhurakij Pundit University in Thailand for presenting his research paper in International conference. He was honored with Best Professor Award 2012 by ASDF Global Awards 2012, Pondicherry.