ISSN: 2278 - 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 1, Issue 8, October 2012

A Survey on Fault Tolerance in Work flow Management and Scheduling

K.Ganga[1] ,Dr S.Karthik[2], A.Christopher Paul[3]

Abstract— **Fault Tolerance is a configuration that prevent a computer or network device from failing in the event of unexpected problem or error such as hardware failure, link failure, unauthorized access, variations in the configuration of different systems and system running out of memory or disk space. The integration of fault tolerance measures with scheduling gains much importance. Workflow management systems support fault tolerance and efficient data handling mechanisms.**

Index Terms—**Cloud computing, Fault Tolerance, Grid Computing, workflow management.**

I.INTRODUCTION

Cloud computing is a comprehensive solution delivers IT as a service. It is an internet based computing solution where shared resources are provided like electricity. The flexibility of cloud computing is a function of allocating resource on demand. Cloud computing is the combination of grid computing and utility computing. Cloud computing has the potential to create irreversible changes in how computers are used around the world. It is a delivery of computing and storage capacity as a service to a community of end-recipients.

Cloud computing is a way of computing where service is provided across the internet using the models and levels of abstraction[6]. Many research issues are fully addressed in cloud such as Fault tolerance, work flow management, workforce scheduling, security, etc. Grid computing is a federation of computer resources from multiple administrative domains to reach a common goal in a single task to solve the grand challenge problem such as protein folding, financial modeling etc. Grid computing is sharing of coordinated resources in a dynamic environment in which multi-institutional organizations involved.

Cloud computing is a way of computing where service is provided across the internet using the models and levels of abstraction[6]. Many research issues are fully addressed in cloud such as Fault tolerance, workflow management, workflow scheduling, security, etc. Grid computing is a federation of computer resources from multiple administrative domains to reach a common goal in a single task to solve the grand challenge problem such as protein folding, financial modeling etc. Grid computing is sharing of coordinated resources in a dynamic environment in which multi-institutional organizations involved.

Fault Tolerance is a major concern to guarantee the availability and reliability of critical services as well as application execution. In order to minimize failure impact on the system and application execution, failures should be anticipated and proactively handled. Fault tolerant techniques are used to predict the failure in appropriate action[2][3]. Fault tolerance is one of the important key issue in cloud. It is concerned with all the techniques necessary to enable a system to tolerate software faults remaining in the system after its development. The main benefits of implementing fault tolerance in cloud computing include failure recovery, lower cost,improved performance metrics etc. When multiple instances of an application are running on several virtual machines and one of the server goes down, there exists a fault and it is implemented by fault tolerance[7].

2. Background: There are various faults which can occur in cloud computing. Based on the fault tolerance policies various fault tolerance techniques can be used such as workflow level and task level.

2.1 Proactive fault tolerance

The principle of proactive fault tolerance policies is to avoid recovery from faults, errors and predict the failure and proactively replace the suspected components from other working components. Some of the techniques based on these policies are Preemptive migration and Software Rejuvenation.

2.1.1 Preemptive Migration: Proactive fault tolerance using preemptive migration relies on a feedback loop control mechanism where application is constantly monitored and analyzed.

2.1.2 Software Rejuvenation: It is a technique that designs the system for periodic reboots. It restarts the system with clean state[5].

2.2 Reactive fault tolerance

Reactive fault tolerance policies reduce the effect of failures on application execution when the failure

Manuscript received Sep 15, 2012.

K.Ganga PG Scholar, 2nd year Computer science and Engineering, SNS College of Technology, Coimbatore.

Dr.S.Karthik, Dean, Department of Computer science and Engineering, SNS College of Technology, Coimbatore.

***A.Christopher Paul**, Asst.professor, Department of Computer science and Engineering, SNS College of Technology, Coimbatore.*

effectively occurs. There are various techniques which are based on these policies like checkpoint/Restart, Replay and Retry.

2.2.1 Check pointing/Restart: When a task fails, it is allowed to be restarted from the recently checked pointed state rather than from the beginning. It is an efficient task level fault tolerance technique for running applications.

2.2.2 Replication: Replication based technique is one of the popular fault tolerance techniques. Replica means multiple copies. Replication is a process of maintaining different copies of a data item or object[12]. In replication techniques, request from client is forwarded to one of replica among the set of replicas. Various task replicas are run on different resources, for the execution to succeed till the entire replicated task is not crashed. Replication adds redundancy in the system. It can be implemented using tools like Ha Proxy, Hadoop and AmazonEC2.

Consistencies among replica,replica management, replica on demand, degree of replica etc are some important issues in replication based fault tolerance technique. A replication protocol must ensure the consistency among all replicas of the same object. Multiple copies of same entity causes problem of consistency due to update of any copy by one of the user. Primary-backup replication, voting and primary-per partition protocol are some of the replication protocol. Large number of replicas will increase the cost of maintaining the consistency.

2.2.3 Task Resubmission: It is the most widely used fault tolerance technique in current scientific workflow systems. Whenever a failed task is detected, it is resubmitted either to the same or to a different resource at a runtime[4][6].

2.2.4 User defined Exception Handling: In this user specifies the particular treatment of a task failure for work flows

Hwang proposes a multi-layered approach for fault tolerance in work flows[4]. The former tries to hide faults that happen during the execution of single tasks at the work flow level, while latter manipulates the structure of the work flow to deal with faults dynamically. Besides these, several layers can be identified where detection as well as recovery and prevention may exist,

2.2.5 Hardware level: lowest level, machine crashes and network connectivity errors can happen

2.2.6 Operating system level: Tasks may run out of memory or disk space or exceed CPU time limit. Other faults like network congestion or file nonexistence can also happen.

2.2.7 Middle ware level: non-responding services probably caused by too many concurrent requests. Authentication, file staging or job submission failures can happen and submitted jobs could hang in local queues or even lost before reaching the local resource manager.2.2.8 Tasks level: job related faults can happen, like deadlock, live lock, memory leak, uncaught exceptions, missing shared libraries or job crashes, even incorrect output results could be produced.

3.Scientific Work flow

Scientific work flow is concerned with the automation of scientific processes in which tasks are structured based on their data and control dependencies[4][15]. The work flow paradigm for scientific applications on grid offers several advantages, such as (a) ability to build dynamic applications with distributed resources. (b) Utilizing resources that are located in a particular domain to increase throughput or reduce execution costs. (c) Execution spanning multiple administrative domains to obtain specific processing capabilities and (d) Integration of multiple teams involved in management of different parts of the experiment work flow. The scientific work flow systems consists of two elements in the grid work flow management system namely work flow design and work flow scheduling. Jayadivya S.K proposed a fault tolerant work flow scheduling (FTWS) algorithm using task replication and resubmission based on the priority of the task to achieve a good success rate by satisfying the deadline.

3.1 Work flow Design

Workflow design determines how work flow components can be defined and composed.

3.1.1 Workflow Structure: A work flow is defined as a task or set of task processed in a predefined order based on their data and control dependencies. Workflow structure indicates the temporal relationship between the tasks. In general, a workflow can be represented as a Directed Acyclic Graph (DAG) and non-DAG.

3.1.2 Workflow Model:Workflow model is also called as Workflow specification defines a workflow including its task definition and structure definition. It consists of two types namely, abstract and concrete. In the abstract in which the workflow is specified without referring to specific resources for task execution.

In DAG-based workflow, workflow structure can be categorized as sequence, parallelism and choice. Sequence is defined as an ordered series of tasks, with one task starting after a previous task has completed. Parallelism represents tasks which are performed concurrently rather than serially. Choice represent the task selected is to be executed at run time. In non-DAG workflow determines the iteration structure. 3.1.2 Workflow Model:Workflow model is also called as Workflow specification defines a workflow including its task definition and structure definition. It consists of two types namely, abstract and concrete. In the abstract in which the workflow is specified without referring to specific resources for task execution.

3.1.3 Workflow Composition

Workflow composition systems are designed for enabling users to assemble components into workflows. It is

a high level view for the construction of grid workflow applications and hide the complexity of grid systems. It is of two categories namely, user directed and automatic. User directed systems require users to edit workflow directly. Users can use workflow languages such as Extensible Markup Language (XML) for language based modeling and the tools such as Kepler for graph based modeling to compose workflows. Automatic systems generate workflows for users according to their higher level requirements such as data products and initial input values.

3.2 Workflow Management System

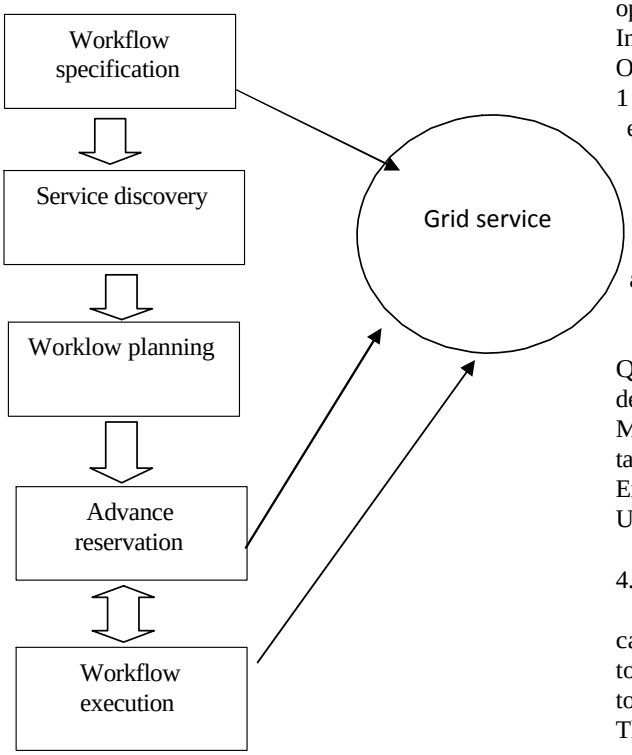

Figure No 1

Figure 1 represents the architecture of workflow management system. User first submits workflow specifications with their QoS requirements. The System then discovers appropriate services for processing the tasks and schedule the task on the service. It consists of three major steps in workflow scheduling: Performance estimation, Workflow planning and workflow execution with run-time scheduling [3][8]. Workflow planning is to select a service and execution time slot for every task in the workflows based on QoS constraints, capability and availability of the services. At workflow execution time, the contract between a service provider and workflow management system may be violated by many reasons such as resource failure.

3.3 Workflow Scheduling

Workflow scheduling focuses on mapping and managing the execution of workflow tasks on shared resources that are not directly under the control of workflow systems.

3.3.1 Scheduling Architecture

The architecture for the scheduling infrastructure is very important for the scalability, autonomy, quality and performance of the system. There are three major categories in workflow scheduling architecture namely centralized, hierarchical and decentralized scheduling schemes. The main advantage of hierarchical architecture is different scheduling policies can be deployed in the central manager and lower level schedulers[7][8]. Decentralized scheduling is more scalable to generate optimal solutions for overall workflow performance.

3.3.2 Jia Yu proposed a Scheduling algorithm for cost optimization within users' deadline

Input: A workflow graph A(x ,y, z)
Output: A schedule for all workflow tasks
1 request processing time and price from available ervices
convert A into task partition graph G(V, E, D)
distribute deadline D over V and G
Repeat
S get unscheduled task partition whose task partitions ave been scheduled
for all i √ S do
ompute ready time of i
Query available time slots during ready time and sub-deadline on available services
Make advance reservations with desired services for all tasks in i adjust sub-deadline of i
End for
Until all partitions have been scheduled.

4.Conclusion

We have presented a survey on fault tolerance techniques is capable to prevent the further loss due to faults. Fault tolerance is concerned as a setup or to enable a system to tolerate software faults in the system after its development. This paper discussed the fault tolerance techniques in cloud computing and focus on workflow management systems and workflow scheduling.

5.References

[1] Antonina Litvinova, Christian Engelmann and Stephen L. Scott," A Proactive Fault Tolerance Framework for High Performance Computing", 2009.

[2] Elvin Sindrilaru,,Alexandru Costan,, Valentin Cristea,"Fault Tolerance and Recovery in Grid Workflow Management Systems", 2010 International Conference on Complex, Intelligent and Software Intensive Systems.

[3] Yang Zhang1, Anirban Mandal2, Charles Koelbel1 and Keith Cooper," Combined Fault tolerance and Scheduling Techniques for Workflow Applications on Computational Grids "in 9th IEEE/ACM international symposium on clustering and grid, 2010.

[4] Kassian Plankensteiner, Radu Prodan, Thomas Fahringer,"A New Fault Tolerance Heuristic for Scientific

Workflows in Highly Distributed Environments based on Resubmission Impact", Fifth IEEE International Conference on e-Science,Austria,2009
[5] Manish Pokharel and Jong Sou Park, "Increasing System Fault Tolerance with Software Rejuvenation in E-government System", IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.5, May 2010
[6] Wenbing Zhao, P. M. Melliar-Smith and L. E. Moser,"Fault Tolerance Middleware for Cloud Computing",2010 IEEE 3rd International Conference on Cloud Computing.
[7] S.ThamaraiSelvi, Ponsy R.K.SathiaBhama, S.Architha, T.Kaarunya and K.Vinothini, (2010) "Scheduling inVirtualized Grid Environment Using Hybrid Approach" International Journal of Grid Computing & Applications (IJGCA) Vol.1, No.1.
[8] Ivan Rodero, Francesc Guim, Julita Corbalan, 2009, Evaluation of Coordinated Grid Scheduling Strategies, 11th IEEE International Conference on High Performance Computing and Communications, DOI 10.1109/HPCC.2009.28.
[9] N.Malarvizhi, Dr.V.Rhymend Uthariaraj. (2009): A Minimum Time To Release Job Scheduling Algorithm in Computational Grid Environment, IEEE Fifth International Joint Conference on INC, IMS, IDC.
[10] Sameer Singh Chauhan, R. C. Joshi, (2010), QoS Guided Heuristic Algorithms for Grid Task Scheduling, International Journal of Computer Applications (0975 - 8887),Volume 2 - No.9
[11] He X , Sun, X., Laszewski, G.V., (2003). Qos guided min-min heuristic for grid task scheduling, Journal of Computer Science and Technology 18, 442-451.
[12] Q. Zheng, B. Veeravalli, and C. Tham.(2007): Fault-tolerant Scheduling for Differentiated Classes of Tasks with Low Replication Cost in Computational Grids, ACM, HPDC 07, June 25-29, 2007, Monterey, California, USA.
[13] Singh, G., C. Kesselman, and E. Deelman (2005). Optimizing gridbased workflow execution. In: *Journal of Grid Computing*, 3(3-4):201-219.
[14] Yu, J., and R. Buyya (2005). A taxonomy of workflow management systems for grid computing. In: *Journal Grid Comput.*, 3(3-4):171-200.
[15] Bowers, S., Ludäscher, B., Ngu, A., Critchlow, T.: Enabling Scientific Workflow Reuse through Structured Composition of Dataflow and Control-Flow. IEEE Workshop on Workflow and Data Flow for Scientific Applications, 2006.
[16] Jayadivya S.K, Jaya Nirmala S, Mary Saira Banu. Fault Tolerant workflow scheduling based on Replication and Resubmission tasks in Cloud, International Journal on computer Science and Engineering Vol 4 No.06 2012
[17]Ananthi.S, Senthil MS, Karthik.S " Privacy preserving keyword search over encrrpted cloud data" in Advances in computing and communication page 480-487.

Ganga .K received B.Sc degree in Physics from Government Arts College for Women, Thanjavur and also received MCA degree from Bharathidasan University, Tiruchirapalli. She is currently pursuing M.E degree in Computer Science and Engineering at SNS College of Technnology, Coimbatore. Her area of interests are Cloud Computing, Software Engineering and Networking.

Christopher Paul.A received B.E degree in computer science and Engineering Sri Ramakrishna Institute of Technology, Coimbatore and also received M.E Computer Science and Engineering from Anna University, Coimbatore. He has presented five national conference and one International conference. His area of interests are Software Engineering and MANET.

Professor Dr.S.Karthik is presently Professor & Dean in the Department of Computer Science & Engineering, SNS College of Technology, affiliated to Anna University-Coimbatore, Tamilnadu, India. He received the M.E degree from the Anna University Chennai and Ph.D degree from Ann University of Technology, Coimbatore. His research interests include network security, web services and wireless systems. In particular, he is currently working in a research group developing new Internet security architectures and active defense systems against DDoS attacks. Dr.S.Karthik published more than 35 papers in refereed international journals and 25 papers in conferences and has been involved many international conferences as Technical Chair and tutorial presenter. He is an active member of IEEE, ISTE, IAENG, I ACSIT and Indian Computer Society.