Survey On Proactive And Reactive Based Reliability-Aware Technique In Cloud Computing

Vivek Kumar Gupta

Abstract : As cloud grows in popularity, thousands of companies are simply rebranding their non cloud products and services as “cloud computing”. Number of clients uses services offered by Cloud computing.. According to user demands increases day by day, there is a need to enhance the cloud services to deliver timely and accurate services. Cloud services must be reliable in order to provide a quality to the clients requirements. The need of physical size of high performance computing environment is also increase rapidly. Higher the need of physical size , more failure are likely to occur that result in poor reliability of system. To deal with reliability, services providers must know the failure issues of cloud computing. In some cases fault tolerance is the major challenge for cloud environment. Some fault tolerance technique like self healing, job migration, are established but they are not fully reliable for cloud computing. In order to better handle the failure using fault tolerance aware techniques at the time of scheduling the application tasks. Thus, in this paper we presented the survey of fault tolerance aware scheduling technique. This survey helps the researchers to work in the area of fault tolerance aware scheduling to take best scheduling decisions and increase the reliability and efficiency of application.

Index Terms: Virtualization, Fault tolerance, CSP, IaaS, SaaS, PaaS.

1. INTRODUCTION

CSP, or cloud service provider, is a company which provides basic services as infrastructure as a service(IaaS), software as a service(SaaS), or platform as a service(PaaS) to other business or individual. Public, Private Hybrid are different type of cloud providers which are responsible for providing services to other business or individual. Today cloud computing grows in popularity because of its enormous features such as resource pooling and elasticity, on demand services, resource provisioning and virtualization and quality of service. Although cloud computing provides many services and makes computing very reliable, fast, easy and dynamic, it is still facing many problems due to its large size and complex architecture. As users need to pay for required resources on basis of pay as you go model, cloud service provider must provide reliable services as per the QoS [1] requirements of the users.In cloud computing environment, reliability and efficiency of applications depend upon many factors like scheduling of application task on resources and occurrence of failure of resources. The main aim of scheduling algorithm is to assign the task to the available resource with the objective to maximize the reliability and efficiency of an application. For this number of resources have been carried out on scheduling problems. Since scheduling problem is NP hard in nature, the possible solution are the different approaches like heuristics, meta-heuristics or approximation. Failure of resources decreases the reliability of application execution that affects the both client and service providers. Garraghan et al. [2] put efforts in knowing the failure characteristics of wide-scale cloud applications. They investigated on the Google Cloud Trace Log for measuring the failure character- istics of cloud servers and tasks, comprising more than 12,500 servers that took 29 days to operate. They observed that the failures’ rate of application tasks follow different theoretical distribution like Lognormal or Weibull. Further, analysis on distributed systems [3–5] indicates that the computing resources have Poisson failure distribution. In order to improve the efficiency, it is necessary to maximize the reliability of the system. For this, understanding of foundering and rectifying attribute of cloud resources are very crucial.

In order to improve the efficiency and reliability of

2. APPLICATION EXECUTION FAILURES CAN BE HANDLED BY FAULT-TOLERANCE-AWARE TECHNIQUES WHICH ARE CLASSIFIED EITHER IN PROACTIVE MANNER OR REACTIVE MANNER.

2.1 Proactive Fault Tolerance

In order to minimize the effect of failure in proactive manner, in cloud computing environment , tasks are scheduled to reliable VM according to prior failure information. Different Proactive fault tolerance techniques are proposed for scheduling the tasks in order to maximize the reliability of application execution. Predictive- Based Fault Tolerance Aware Scheduling TechniqueHistoric Log Basic: Historic log Basic Scheduling of tasks considering reliability of processors and communication links under the Possion failure distribution of network. In Ref. [6], proactive approach is proposed for resource reliability considering three parameters: CPU (MIPS), memory (RAM), and bandwidth (BW). Proactive approach is proposed for resources reliability considering three parameters: CPU(MIPS), Memory(RAM) and bandwidth(BW). The system model first finds out the reliability of each VM on prior information like available memory, available MIPS ratio and available BW ratio assigns the cloudlets to best reliable VM. Weibull failure distribution is considered in Ref. [7] for maximizing the reliability by scheduling the tasks to reliable VM. Monte Carlo Estimation Technique: Monte Carlo Failure Estimation algorithm is developed to investigate the future patterns for scheduling the tasks by estimating the failure of VM using Weibull failure distribution in cloud Failure-Aware Resource Scheduling (FARS) algorithm is proposed to schedule the tasks to various virtual machines. In this, reliability is taken into consideration to map the tasks in workflow applications. Since the more reliable VM are always selected for execution, this reduce the schedule length of FARS. Mapping Strategy-Based Reliability Aware Technique: Since scheduling problem is NP hard in nature, heuristic and meta-heuristic are the preferred. The heuristic based approaches are problem specific and may not find optimal solution. Meta-heuristic approach are used to handle

- *Vivek Kumar Gupta, Department of Computer Science, R.R.M.K Arya Mahila Mahavidyalaya,Pathankot,Punjab,India,E-mail: vk01gupta@gmail.com*

problem of local optima. Heuristic Approach: In Ref [8], a heuristic algorithm is developed to achieve one objective, i.e., maximizing reliability under the constraint of another objective, i.e., minimizing end-to-end delay (EED) for distributed computing systems. A heuristic algorithm is developed to achieve maximum reliability under the constraint of another objectives and minimize end-to-end delay(EED) for distributed systems. This algorithm is developed to achieve one objective i.e reliability is maximum under time bound constraint. In centralized mapping scheme only one server collects all the information related to network and decides about to mapping but in heuristic approach all the distributed servers participate in taking the decision of mapping scalability while keeping the status of failure rates. Meta-heuristic approach: The meta-heuristic approach is developed to achieve maximization of the reliability. NSGA-11 approach is used with ENLU technique to avoid applying the non-domination sorting from raw data and ENLU helps in sorting the solution by using existing knowledge of current population for scheduling the application system and maintain the reliability of the system. Criteria-Based Reliability Aware Technique: Single Objective: in This algorithm particles can communicate with each other . This is the main feature of particle swarm optimization(PSO). Hill climbing heuristic approach is applied with in particle swarm optimization to maximize the system reliability. To maximize the reliability of the system hill climbing approach is used within particle swarm optimization. Multiobjective: Various multiobjective algorithms [9,10] have been developed based on multiple QoS parameters that include reliability, energy consumption, makespan, deadline, scalability. In this algorithm number of QoS parameters have been used which include reliability, scalability, energy consumption. To maximize the reliability and minimize the energy consumption of the system bi-objective genetic algorithm is used. This is done by executing task parallelly as a combinatorial optimization problem. To maximize the reliability of the system hill climbing approach is used within particle swarm optimization. A scheduling algorithm is developed to optimize the conflicting objectives that make span of task along with the maximization of reliability by using meta-heuristic approach.

2.2 Reactive Fault Tolerance :

When the failure occurs, the effect of it on application execution is handled by the reactive fault tolerance polices. Different reactive fault tolerance technique are used for handling the failure of application tasks. Checkpointing Based Fault Tolerance Technique: To established recovery points, a method is used that records the system state periodically, called checkpointing. Whenever the failure occurs, computation always restarts from the last one saving state. For example, if an application needs to access 1000 files to complete its execution and after reading of 999 files, there is some failure, then all its computation gets lost. In order to avoid this, restart it from last saving state. Through checkpointing there is a method for strorage the snapshot system state to be done locally or globally. Storing process status in shared disk increases achieves high reliability but increase the rest of migration process in case of storing in local disk. Various checkpointing techniques have been introduced by researchers like: Full checkpointing: In full checkpointing, at every fixed interval of time, a complete state of process is saved to some media. In order to maximize application execution time after optimal interval, checkpoints should be applied so that cost overhead is minimum. In Ref. [11], author developed an algorithm for optimal checkpointing by efficiently selecting checkpointing storage and recovery server. Incremental Checkpointing: In incremental checkpointing , instead of saving the whole process, only those pages are saved in which there have been any changes. This helps in reducing the checkpoint overhead by firstly saving the state of system as full checkpoint and then after that some method is used and applied on pages which have been update them. Checkpointing based on Read only and Read Write part in virtual machine image approach is developed. At first, the read only part of the state is saved only once while read write modification will saved by the rest of checkpoint. Since checkpoint can parallel recover from different nodes this method is efficient as rather than one task recovery, several task are recovered simultaneously. Uncoordinated Checkpointing and Coordinated Checkpointing : In this , checkpointing is considered independently on each process. Process does not have any type of synchronization between them to form a uniform global-checkpoint. Hence the effect called ‘domain effect’ is the main drawback of it. In fault tolerance in distributed applications, coordinated checkpointing is an method where to make system constant all the process synchronize their checkpointing. This can be used to bound the rollback. User-level checkpointing: In user level checkpointing, recovery from failue, explicit linking is needed necessary with the user level library. Different requirement related to each task is specified by the user. Using coordinated checkpointing scheme for each task synchronously checkpointing all its VMs. At the same time it provides equal reliability to multiple users using peer to peer checkpointing. It is the main advantage of this algorithm. Replication based fault tolerance technique: Replication based fault tolerance technique is widely used fault tolerance technique. In this replication works on single resource as a guard against the failure rather on failure of multiple resources simultaneously. The main concern is number of replicas that is required. Because if number of replicas increases, the fault will also increase due to which management of backup system is very costly. Job replication, component replication, data replication are different type of replication approaches studied under cloud. Active Replication : in active replication, job will considered as succeeded only when there is atleast one processor which run the job completely. Since each and every job is replicated on several processors where all the similar processors are invoked simultaneously. For the reliability of the system on algorithm is developed called a Byzamine fault tolerance framework. A byzamine fault tolerance group is chosen in BFT cloud. This group consists of one primary and 3k replications of task. Response to current request are judge by cloud module which will be either committed or not committed. In BFT group, execution of task will be done again on newly selected primary node and replicas in case of ‘k’ fault nodes. After its recognition, fault resources will be adjusted with other. Chance of failure of BFT group members is very low but it achieves higher throughput. In Ref. [12], MaxRe algorithm is developed for handling the

fault during execution. During execution, fault can be handled. This can be done by MaxRe algorithm. Passive Replication: If there is any event in primary processor, for completing its execution, the task will be scheduled on backup process. For rescheduling strategy algorithm is developed. In this number of replicas depend on number of failue but in Maxre it depends upon execution time. But there is a problem with this algorithm, it is too costly and for multiple failure it is not a suitable one. If once there is any failure is detected, multiple replication of task are executed. Another algorithm is developed with double strategy. To maximize the efficiency and reduce schedule time, backup copies are overlapped with the back copy of precedence tasks which is on the same processors. Checkpoint along with replication: In Ref. [13], algorithm dynamically selects the most suitable reactive fault tolerance technique for task execution. For selecting suitable reactive fault tolerance technique for the execution of task dynamically an algorithm is developed. This algorithm identifies numerous replication of tasks based on history and its length of checkpointing interval. As in general, on the increasing of number of task, reliability will decreases but there is an algorithm suggested called RMSR Replication- Based scheduling for maximizing system reliability. In this algorithm , if the number of task increases, the reliability will increases by dynamically replicating task determine by user according to threshold value. Primary Backup replication : In Ref. [14], an algorithm FASTER is developed for real-time applications in virtualized cloud. For real-time applications in virtualized cloud an algorithm is developed called FASTER. This algorithm extended the primary backup based scheduling by adding the cloud feature like elasticity and virtualization. An algorithm considers an dependent and independent task with primary backup approach. It considers replications cost and response time for both dependent and independent tasks. In case of dependent task, backup copies can be scheduled with precedence task and in case of independent task, these copies can be scheduled with any back up copy of another task on same processor with less cost and replication time.

3 CONCLUSION

Since Cloud computing offers variety of services to its client. As the user demands increases, there is a need to enhance the cloud services to deliver timely and accurate services. For this, the need of physical size of high performance computing environment is also increase rapidly. Higher the need of physical size , more failure are likely to occur that result in poor reliability of system. To deal with reliability, services providers must know the failure issues of cloud computing. Thus, in this paper we identified different types of fault tolerance aware scheduling techniques.

REFERENCES

[1] Patel, P., Ranabahu, A.H., Sheth, A.P.: Service Level Agreement in Cloud Computing (2009)

[2] Garraghan, P., Townend, P., Xu, J.: An empirical failure-analysis of a large-scale cloud computing environment. In: 2014 IEEE 15th International Symposium on High-Assurance Systems Engineering (HASE), pp. 113–120. IEEE (2014)

[3] Attiya, G., Hamam, Y.: Task allocation for maximizing reliability of distributed systems: a simulated annealing approach. J. Parallel Distrib. Comput. 66(10), 1259–1266 (2006)

[4] Rehani, N., Garg, R.: Meta-heuristic based reliable and green workflow scheduling in cloud computing. Int. J. Syst. Assur. Eng. Manag. 1–10

[5] Zhou, A., Wang, S., Cheng, B., Zheng, Z., Yang, F., Chang, R., Buyya, R.: Cloud service reliability enhancement via virtual machine placement optimization. IEEE Trans. Serv. Comput. (2016)

[6] Charity, T.J., Hua, G.C.: Resource reliability using fault tolerance in cloud computing. In: 2016 2nd International Conference on Next Generation Computing Technologies (NGCT), pp. 65–71. IEEE (2016)

[7] Zhou, A., Wang, S., Cheng, B., Zheng, Z., Yang, F., Chang, R., Buyya, R.: Cloud service reliability enhancement via virtual machine placement optimization. IEEE Trans. Serv. Comput.

[8] Cao, F., Zhu, M.M.: Distributed workflow mapping algorithm for maximized reliability under end-to-end delay constraint. J. Supercomput. 66(3), 1462–1488 (2013)

[9] Dongarra, J.J., Jeannot, E., Saule, E., Shi, Z.: Bi-objective scheduling algorithms for optimizing makespan and reliability on heterogeneous systems. In: Proceedings of the nineteenth annual ACM symposium on Parallel algorithms and architectures, pp. 280–288. ACM (2007)

[10] Wang, X., Yeo, C.S., Buyya, R., Su, J.: Optimizing the makespan and reliability for workflow applications with reputation and a look-ahead genetic algorithm. Fut. Generat. Comput. Syst. 27(8), 1124–1134 (2011)

[11] Zhou, A., Sun, Q., Li, J.: Enhancing reliability via checkpointing in cloud computing systems. China Commun. 14(7), 1–10 (2017)

[12] Zhao, L., Ren, Y., Xiang, Y., Sakurai, K.: Fault-tolerant scheduling with dynamic number of replicas in heterogeneous systems. In: 2010 12th IEEE International Conference on High Performance Computing and Communications (HPCC), pp. 434–441. IEEE (2010)

[13] Amoon, M.: Adaptive framework for reliable cloud computing environment. IEEE Access 4, 9469–9478 (2016)

[14] Zhu, X., Wang, J., Guo, H., Zhu, D., Yang, L.T., Liu, L.: Fault-tolerant scheduling for real-time scientific workflows with elastic resource provisioning in virtualized clouds. IEEE Trans. Parallel Distrib. Syst. 27(12), 3501–3517 (201