Fault Tolerance Detection, Mitigation and Performance Improvement in Cloud

[1]S. Veerapandi, [2]Dr. K. Alagarsamy
[1]*Research Scholar,* [2]*Associate Professor Rtd*
School of information Technology
Madurai Kamaraj University

Abstract:

In the modern world computer become a necessary one. To work with computer system, software and hardware are necessary component. Due to technology development software and hardware prices has been increased nothing like anything. Purchasing new hardware & software become tedious process and expensive. Research found that cloud will overcome this kind of problem using the concept of usage based payment which growing as trendy technical solution and moving one. In cloud computing there are many problems persist but in our paper we have taken fault tolerance issue and sustainability. In this paper we have proposed a novel idea to improve the cloud fault tolerance and performance.

Keywords

Cloud Computing, MPI, Indexing Table

I. INTRODUCTION

Cloud based technology is sharing of resources on a larger scale which makes cost effective and location independent. Resources on the cloud can be uploaded by the vendor and used by the end user. Cloud also shares necessary software's andhardware's on-demand tools for various IT Industries.Amazon has predicted the importance of Cloud computing at first then after Google and IBM. It is a model for enabling convenient, on-demand resource access to a shared pool of configurable computing resources likeapplications,networks, storage,servers, and services.

Cloud computing reached as a new computing methodology which provides reliable, and quality of service guaranteed computing lively environments for end-users. The growing demand for flexibility in obtaining and releasing computing resources in a cost-effective manner has resulted in a wide adoption of the Cloud computing.

Fault tolerance is a methodologywhich can continue to correctly perform theirprocess in the presence of hardware failures, software or to operate satisfactorily in the presence of faults.

Thus, an important motivation for differentiating between faults and failuresis the need to describe the fault tolerance of a system. An observer inspecting the internals ofthe system would say that the faulty component had failed, because the observer's viewpointis now at a lower level of detail.

II. REVIEW OF LITERATURE

Uesheng Tan, DengliangLuo, and Jingyu Wang, "Cc-vit:Virtualization intrusion tolerance based on cloud computing," [38] proposed a replication methodology for VMFT, exclusively managed by the cloud provider. It describesto improve efficiency by using passive VM replicas, which become active when a failureis detected. A methodologywas introduced to transferthe state of VM. This solution is similar to the exclusive FT

VishonikaKaushal and VishonikaKaushal, "Autonomic fault tolerance using haproxy in cloud environment," [35] proposes an FT solution in the cloud at thecustomer level by replicating servers queries, based on theHA-Proxy load distributor. Other researches such as Qin Zheng, "Improving mapreduce fault tolerance in thecloud,"[36]and Magdalena Slawinska, JaroslawSlawinski, and VaidySunderam,"Unibus: Aspects of heterogeneity and fault tolerancein cloud computing," [37] propose

a collaborative solutions for specifics applications. However, they do not considerthe splitting of the cloud between a provider and itscustomers, so their works are only applicable to anSaaScloud.

OpenNebula, "Opennebula.org: The open source toolkit forcloud computing," the Microsoft Windows Azure platform[33] offers an exclusive FT management at the cloudlevel. Windows Azure replicates each VM so that a VMfailure is covered by its replicas. The Azure solution islimited to web applications developed in the Windows Azureplatform. Moreover, no solution is proposed to repair thefailed VM. In addition, for VMs which are not instantiatedby the Azure development platform, the entire responsibilityof FT management is left to the customer. This is also the case in the Amazon EC2 [34] cloud platform.

III. PROPOSED METHOD

After deep review of literature review we have found that main fault tolerance issue in cloud computing are recovery and detection. To solve these issues, many techniques were proposed by many authors but still fault tolerance is the major issue in cloud environment due to internet based, virtual mechanism, too many servers, lots scheduling, message passing system, etc. based on the study we have proposed a novel approach for detecting the fault tolerance and sustain in the fault tolerant environment. This is also very useful to reduce the future fault tolerance occurrence.

IV. PROPOSED ALGORITHM

1) Start
2) INT Task_CNT= 1;
3) Cloud System = True;
4) Index value = 0;
5) CNT = Task.CNT
6) If CNT > 1
7) Initiate job scheduler
8) Call MPI
9) If System = ok
10) Allocate Task
11) Update Index Table
12) Else
13) Not allot & Execute
14) Update Index Table
15) Continue until all task over
16) Else
17) No cloud Action

There are multiple tasks will run in the cloud environment, which controlled by job scheduler, which executes the tasks in cloud environment. Whenever scheduler executes the job in cloud computing, message passing interface sends the message to the system to check the availability of the system to execute the tasks. It will check the following attributes for the availability of the system.

- Speed
- RAM size
- Processor Bus
- Disk type
- Disk size
- Cache
- CPU Type
- SCSI Controller

While checking, there may be a chance to numerous system may not fit for executing which make system idle or non-suitable. In the existing method whenever task allotted scheduler will check every time to get the status of system, which makes the time consuming process and energy waste. To avoid this we have combined the concept of indexing table from the data structures. Indexing table will make the process simple, index node denote the availability of system and every sub table will contain the above parameters which are based on the previous tasks history. Index table will be automatically updated every time for every task response it will reduce the response time for new tasks. It was tested in amazon web services.

Tasks	Existing Method (MPI) Response Time	Proposed Method Response Time
7	0.438	0.281
14	0.527	0.306
21	0.605	0.425
28	0.623	0.467
35	0.681	0.498

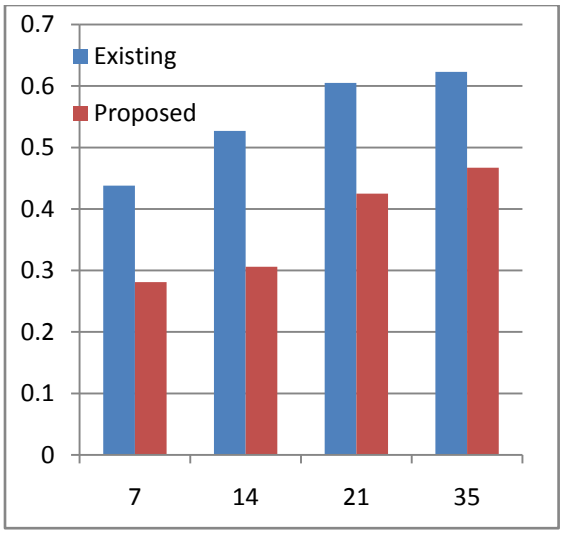

Fig 1: Response Time Analysis

Tasks	Existing Method (MPI) Check Point Requirement	Proposed Method Check Point Requirement
27	3	1
54	3	1
81	5	2
105	8	4

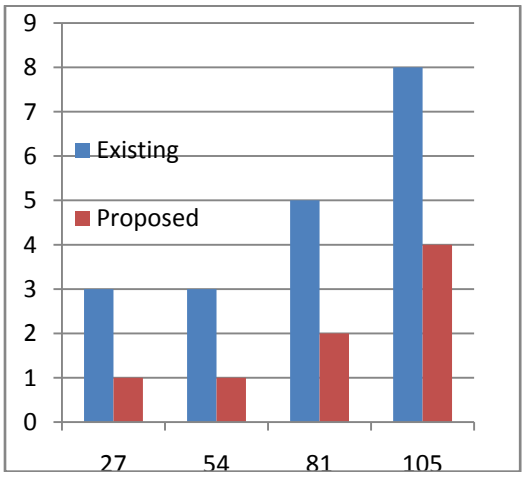

Fig 2: Check Point Requirement

V. CONCLUSION

There are numerous methods available for fault tolerance detection and task allocation. In this method we have proposed a novel approach based on message passing interface with indexing table for the improvement of fault tolerance detection and system availability check. We have simulated our idea with amazon web services and analyzed the performance of our algorithm which showed better response time and minimum checkpoints required compare with existing one. We believe that this research will be useful to modern technical society.

VI. FUTURE WORK

Every research has some benefit and loophole for future enhancement. Young research minds can take our research as benchmark and can develop some new fault tolerance sustainable mechanism using some new data structure technique. We hope that this research will be use full for young innovators.

REFERENCES

[1] Benjamin Lussier, Alexandre Lampe, Raja Chatila,JérémieGuiochet, Félix Ingrand, Marc-Olivier Killijian, David Powell, "Fault Tolerance in Autonomous Systems: How and How Much?" LAAS-CNRS 7 Avenue du Colonel Roche, F-31077 Toulouse Cedex 04, France

[2] Jianlin, Xiaoyi Lu, Lin Yu, YongqiangZou and Li Zha"Vega Warden: A Uniform User Management System for Cloud Applications "2010 Fifth IEEE International Conference on Networking, Architecture, and Storage.

[3] Mladen A. Vouk, "Cloud Computing – Issues, Research and Implementations", Department of Computer Science, North Carolina State University, Raleigh, North Carolina, USA, Journal of Computing and Information Technology - CIT 16, 2008, 4, 235– 246doi:10.2498 /cit.1001391

[4] AnjuBala, InderveerChana," Fault Tolerance- Challenges, Techniques and Implementation in Cloud Science Issues, Vol. 9,Issue 1, No 1, January 2012 ISSN (Online): 1694-0814

[5] Ravi Jhawar, Vincenzo Piuri and Marco Santambrogio"A Comprehensive Conceptual System level Approach to Fault Tolerance in Cloud Computing" IEEE

[6] Jean-clandeLaprie "Dependable computing and fault tolerance: concepts and terminology" LAAS-CNRS 7 Avenue du ColonelRoche, 31400 Toulouse, France

[7] GeoffroyVallee, KulathepCharoenpornwattana, Christian Engelmann, AnandTikotekar, Stephen L. Scott," A Framework for Proactive Fault Tolerance".

[8] SheheryarMalikandFabriceHuet "Adaptive Fault Tolerance in Real Time Cloud Computing" 2011 IEEE World Congress on Service

[9] GolamMoktaderNayeem , Mohammad Jahangir Alam," Analysis of Different Software Fault Tolerance Techniques", 2006.

[10] Wenbing Zhao, P.M. Melliar and L.E. Mose" Fault Tolerance Middleware for Cloud Computing" 2010 IEEE 3rd International Conference on Cloud Computing.

[11] Jayadivya S K, JayaNirmala S, Mary SairaBhanus"Fault Tolerance Workflow Scheduling Based on Replication and Resubmission of Tasks in Cloud Computing" International Journal on Computer Science and Engineering (IJCSE)

[12] Fumio Machida, Ermeson Andrade, Dong SeongKim and Kishor S. Trivedi"Candy: Component-based Availability Modeling Framework for Cloud Service Management Using Sys-ML" 2011 30th IEEE International Symposium on Reliable Distributed Systems.

[13] Sun Microsystems, Inc. "Introduction to Cloud Computing Architecture" White Paper 1st Edition, June 2009

[14] Greenberg, A., Hamilton, J., Maltz, D., and Patel, P. (2009) 'The Cost of a Cloud: Research Problems in Data Center Networks,' ACM SIGCOMM Computer Communication Review, vol. 39, no. 1, pp. 68-79.

[15] ITProPortal, http://www.itproportal.com/2012/07/12/o2outage-latest-string-major-it- infrastructure-failures/, 2012.

[16] Gagne, M, 2007, 'Cooking with Linux—still searching for the ultimate Linux distro?',Linux Journal, vol. 09, no. 161.

[17] Koren, I., & Krishna, C. M. (2010). Fault-tolerant systems: Morgan Kaufmann.

[18] Bauer, E., & Adams, R. (2012). Reliability and availability of cloud computing: John Wiley & Sons.

[19] Bilal, K, Malik, S., Khan, S. U., &Zomaya, A. (2014) 'Trends and Challenges in Cloud Data Centers,' IEEE Cloud Computing Magazine, vol. 1, no. 1, pp. 10-20, 2014.

[20] Pannu, H S, Liu, J, & Fu, S 2012, 'AAD: Adaptive Anomaly Detection System for Cloud Computing Infrastructures', In IEEE 31st Symposium on Reliable Distributed Systems (SRDS), pp. 396-397.

[21] Hakkarinen, D, & Chen, Z, 2013, 'Multi-Level Diskless Checkpointing', IEEE Transactions On Computers, vol. 62, no.4, pp. 772-783.

[22] Kwak, SW, & Yang, JM, 2012, 'Optimal checkpoint placement on real-time tasks with harmonic periods', Journal of Computer Science and Technology, vol. 27, no. 1, pp. 105-112.

[23] Nicolae, B, &Cappello, F, 2011, 'BlobCR: efficient checkpoint-restart for HPC applications on IaaS clouds using virtual disk image snapshots', ACM International Conference for High Performance Computing, Networking, Storage and Analysis.

[24] Malik, S, R, Khan, S, U, and Srinivasan, S, K, 2013, 'Modeling and Analysis of State-of-the-art VM-based Cloud Management Platforms', IEEE Transactions on Cloud Computing, vol. 1, no. 1, pp. 50-63.

[25] Goiri, I, Julia, F, Guitart, J, & Torres, J, 2010, 'Checkpoint-based fault-tolerant infrastructure for virtualized service providers', In IEEE Network Operations and Management Symposium (NOMS), pp. 455-462.

[26] Sun, D, Chang, G, Miao, C, & Wang, X 2013, Analyzing, modeling and evaluating dynamic adaptive fault tolerance strategies in cloud computing environments, The Journal of Supercomputing, 1-36.

[27] Zhang, Y, Zheng, Z, &Lyu, M R 2011, 'BFTCloud: A byzantine fault tolerance framework for voluntary-resource cloud computing', In IEEE International Conference on Cloud Computing (CLOUD), pp. 444-451.

[28] Zhao, W, Melliar-Smith, PM, & Moser, L E 2010, 'Fault tolerance middleware for cloud computing', In IEEE 3rd International Conference on Cloud Computing (CLOUD), pp. 67-74.

[29] Malik, S &Huet, F 2011, 'Adaptive Fault Tolerance in Real Time Cloud Computing', In IEEE World Congress on Services (SERVICES), pp. 280-287.

[30] Ko, S Y, Hoque, I, Cho, B, & Gupta, I 2010, 'Making cloud intermediate data fault-tolerant', In Proceedings of the 1st ACM symposium on Cloud computing, pp. 181-192.

[31] Cogo, V V, Nogueira, A, Sousa, J, Pasin, M, Reiser, HP, &Bessani, A 2013, 'FITCH: Supporting Adaptive Replicated Services in the Cloud', In Distributed Applications and Interoperable Systems, pp. 15-28. Springer Berlin Heidelberg.

[32] Zheng, Q. 2010, 'Improving MapReduce fault tolerance in the cloud', In IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), pp. 1-6.

[33] OpenNebula, "Opennebula.org: The open source toolkit for cloud computing," http://opennebula.org,.

[34] Amazon, Inc, Amazon Elastic Compute Cloud (Amazon EC2). Available: http://aws.amazon.com/ec2/#pricing,.

[35] VishonikaKaushal and VishonikaKaushal, "Autonomic fault tolerance using haproxy in cloud environment," International Journal of Advanced Engeneering Sciences and Technologies,vol. 7, 2010.

[36] Qin Zheng, "Improving mapreduce fault tolerance in the cloud," in Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), pp. 1-6, 2010.

[37] Magdalena Slawinska, JaroslawSlawinski, and VaidySunderam, "Unibus: Aspects of heterogeneity and fault tolerance in cloud computing," in Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), pp. 1-10, 2010.

[38] Uesheng Tan, DengliangLuo, and Jingyu Wang, "Cc-vit:Virtualization intrusion tolerance based on cloud computing," in 2nd International Conference on Information Engineering and Computer Science (ICIECS), pp. 1-6. Wuhan, China,2010.