Fault Identification in Deadline Sensitive Applications in Cloud Computing

Sodyam Bebarta
Department of Computer Science and Engineering
National Institute of Technology
Rourkela, India
sodyam@gmail.com

Tanajit Sutradhar
Department of Computer Science and Engineering
National Institute of Technology
Rourkela, India
tanajit.kgp21@gmail.com

Dr. B. D. Sahoo
Department of Computer Science and Engineering
National Institute of Technology
Rourkela, India
bibhudatta.sahoo@gmail.com

ABSTRACT

Portability is setting the stage for how individuals lap up online media content. Cloud computing has stirred incredible enthusiasm from both scholarly world and in the technical arena quite recently. Because of Service Level Agreement (SLA), the cloud service providers need to ensure the unwavering quality of their frameworks. Nonetheless, erratic equipment or programming disappointments are generally inescapable. For realtime undertakings, failure of the entire system may make them miss their due dates. Thus a robust and a fault tolerant system can effectively evade such cases. The fault tolerant system is generally used to acknowledge adaptation to internal failure by copying an errand into two duplicates a primary one and a secondary piece. The excess presented by reinforcement duplicates brings about additional overhead for cloud frameworks. A media handling model is proposed to empower dynamic scheduling synthesis and cross-pipeline assignment partaking in the cloud for adaptable online processing. The challenge in using cloud administrations for on-request video transcoding, be that as it may, is to keep up a vigorous QoS for watchers and cost-proficiency for spilling administration providers. The model enables application designers to stretch out its ability to empower certain customization for running live channels. In this paper we have discussed the issues that are prevalent in the fault tolerance, elaborated on the two algorithms that we have used to identify the faults.

Keywords—Cloud Computing, Service Level Agreement,Fault Tolerance, Virtual Machines

I. INTRODUCTION

A considerable lot of us ponder with the topic of cloud computing. In an innovatively propelled world, the term cloud computing ought to be well-known to the greater part of us yet the inverse is valid. Cloud computing is a utility that shares assets, programming and data that are in the end gotten to by end clients. Ordinarily, facilitation of these assets happens on virtual servers accessible to the client through the web. Fundamentally, it’s an act of using these servers for capacity, the board and handling of information, instead of utilizing the nearby computer to store information. In seeing distributed computing, one needs to comprehend the advancement of individualized computing. As the web became an indispensible part of our lives, a system design outlines the web as a cloud somewhat to shroud the complexities of this framework and to clarify, its nature. Basically, cloud is really an illustration depicting the web as a space that permits registering in a preinstalled situation that exists as an available support of end clients. A progressively specialized comprehension of this engineering is that distributed computing is a pioneer innovation utilized in joining other figuring arrangements, for example, parallel processing, lattice registering, dispersed processing and an entire host of other virtualization advances that work with utility figuring to achieve end client administrations. With regards to cloud computing, the comprehension of stages is required. In this paper we have discussed the issues that are prevalent in the fault tolerance, elaborated on the two algorithms that we have used to identify the faults.

II. LITERATURE SURVEY

Anju Bala and Inderveer Chana expounded on the current adaptation to internal failure methods in distributed computing dependent on their strategies, apparatuses utilized and investigation difficulties. They also proposed a Cloud virtualized system architecture. In the proposed framework autonomic adaptation to internal failure was actualized. The outcomes showed that the proposed framework can manage different programming issues for server applications in a cloud virtualized condition. [1]

J. Huang , C. Wu and J. Chen proposed a security driven various leveled planning calculation for a cloud-based video gushing framework to accelerate the video transcoding process for ongoing administration applications. It masterminds the structure undertakings and intensely changes the amount of spaces so the cloud packs can execute the methodology even more profitably. Results demonstrated that the proposed planning curb techniques kept up great framework load adjusting.[2]

M. K. Gokhroo , M. C. Govil and E. S. Pilli emphasized on the need to build a robust system for cloud computing to convey the prescribed dimension of unwavering quality. They proposed a novel fault location and alleviation approach. The methodology lied in the technique for identifying the fault dependent on running status of the activity. The identification figuring once in a while checked the progression of work on virtual machines and declared ceased tasks due to inefficient VM to blame inactivity tolerant administrator. This decreased the assets wastage as well as guaranteed convenient conveyance of services.[3]

J. Lee , H. Han and M. set forth a plan that caters to seamless viewers experience against transcoding vitality. They proposed two algorithms that figured out which adaptations ought to be transcoded with the point of augmenting the general QoE. Exploratory outcomes demonstrated that the proposed plan can successfully constrain the measure of transcoding vitality utilization.[4]

P. Guo and Z. Xue’s Essential reinforcement show is broadly used to acknowledge adaptation to non-critical failure by copying an undertaking into two duplicates - an essential one and a reinforcement one. The excess presented by reinforcement duplicates acquired additional overhead for cloud frameworks. They proposed a continuous robust scheduling algorithm with improvement in cloud frameworks.[5]

H. Han , W. Bao , X. Zhu , X. Feng and W. Zhou put forth a robust scheduling calculation named ARCHER for hybrid undertakings in cloud which coordinated the customary reinforcement model and checkpoint innovation and which could adaptably decide the execution time of the reinforcement duplicates of errands, so it incredibly upgrades the asset use and creates additional vacancies to execute assignments however many as could be allowed.[6][7]

III. PROBLEM STATEMENT

- The primary issue that cloud service providers encounter is to establish an effective, inexpensive and more over a trust-worthy answer for an uninterrupted reach to the consumers.
- Choosing the type of VMs in the VM positioner to analyse the transformation so that the transcoding is done in an efficient manner and the cost incurred is also less.

IV. PROPOSED METHODOLOGY

Cloud computing regularly comprises of a few segments for example Datacenter, Servers, Storage, Network, Middleware, Software and Applications, and so on. It is a testing errand to deal with the far reaching rundown of flaws for every one of the segments. System to deal with every one of the shortcomings in cloud is hard to assemble. Consequently we have considered a specific instance of taking care of VMs faults.[10] VM is the fundamental part in cloud. Real disappointments happen because of powerlessness to deal with VMs faults. The real parts incorporate monitoring module (to check the framework parameters) and problem screening section. Every part has a particular reason and is quite dependent to its past stage for information yield/input necessities [12]. The main sections of the methodology are explained as follows.

A. Host Machine

Cloud server dispenses the undertakings to various hosts. The host is in charge of dealing with solicitations coming to it and after that designating those solicitations to machines under its control.

B. Datacenter

The data center comprises of cloud servers, computers, network devices, resources which are required to set up a cloud

C. Cloud Server

Cloud server has indistinguishable work from that of a regular server yet the functionalities may vary. Individuals utilizing cloud lease the cloud server instead of buying the cloud server. Every single demand of customers is made through these cloud servers. It isn't in charge of acquiring every one of the solicitations yet additionally reacting to them.

D. Virtual Machine

These host machines are additionally separated into virtual machines to complete diverse errands all the while. Virtual machines are fundamentally the division of one single machine equipment astute chiefly. In the proposed methodology, each virtual machine has certain memory saved for putting away the work done in last checkpoint length.[9] At the point when the edge of a virtual machine is achieved which means that machine is going to over-burden very soon, live movement process begins without stopping the machine, henceforth downtime is decreased. [13]In the event that a fault happens amid relocation then the information is duplicated from this held memory. The VM provisioner decides which task to be allocated to what virtual machine. It is at this point that the fault detection that can benefit from wastage of resources.

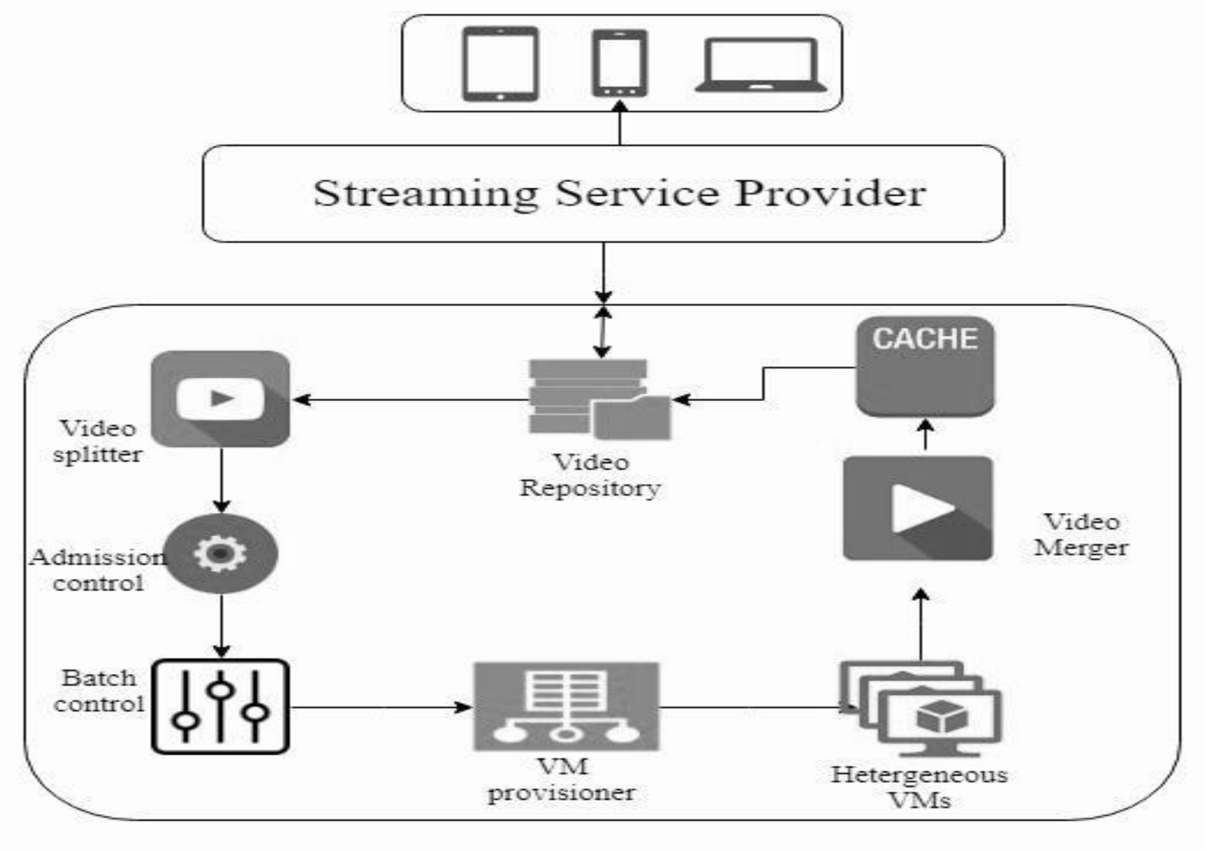

Figure 1: Virtual Machine Provisioning

i: Fixed time Check –point

This method checks for the faults with a fixed interval that has been pre-decided before allocating any jobs to the VM. The Broker marks the start time and the scheduled finish time. If any problem occurs during the transaction it can be found out by the fixed interval check-point and the time of the detection is noted by the Broker.

Algorithm: Allocating Jobs on the Virtual Machines
Input: VMs = {VM1, VM2, VM3 …. VM_n}
Output: Robust Virtual Machine VMs = {VM1, VM_2, VM_3……VM_n }

```
1 Initialize: A fixed time is assumed for the check-point of the VMs
2 The total time taken by a job is divided into segments for the check-point calculation
3 Considering all Virtual Machines Vi that are a part of the execution process do
4 Check if the status of the VMi. = SUCCESS then
5 FailedVMList = Boolean value zero
6 else
7 FailedVMList = Boolean value one
8 Host Machine = on which the VM was allocated initially
9 Taking all Virtual Machine Vi that are allocated on Host Machine do
10 Check if the status of the VMi = FAILED then
11 HostMachineFailedList = Boolean value one
12 Publish Failed VM $V_i$ and  failed Host Machine
13 Savethe time of recognition of the failure
14 Enlist the failed VM $V_i$ to the index of failedVMList and failed host h to the index of failedHostMachineList
```

ii: Increasing time monitoring

The problem with the fixed time check-pointing is that some of the resources are kept busy for checking the progress all the time hence we propose a new algorithm called increasing time monitoring (ITM). The ITM doesn’t go for the check every now and then instead it checks in increasing time intervals. Say suppose the total time is 300 secs, in this the fixed check point of (let’s say) 20 secs would go for the check 15 times while the ITM checks the time doubling the time of the previous check. Let’s assume the initial time is 5 secs, the next would be 10, 20,40…and so on. The total times it’s would check can roughly be around 7 times which is less than half of the fixed check-point algorithm.

Algorithm: Allocating Jobs on the Virtual Machines
Input: VMs = {VM1, VM2, VM3 …. VM_n}
Output: Robust Virtual Machine VMs = {VM1, VM_2, VM_3……VM_n }

```
1 Initialize: The time is fixed by doubling the previously taken time for checking N
2 The total time taken by a job is divided into segments for the check-point calculation
3 for each VM Vi in the execution process where time t= N*2 do
4 Check if the status of the VMi. = SUCCESS then
5 FailedVMList = Boolean value zero
6 else
7 FailedVMList = Boolean value one
8 Host Machine = on which the VM was allocated initially
9 Taking all Virtual Machine Vi that are allocated on Host Machine do
10 Check if the status of the VMi = FAILED then
11 HostMachineFailedList = Boolean value one
12 Publish Failed VM $V_i$ and  failed Host Machine
13 Savethe time of recognition of the failure
14 Enlist the failed VM $V_i$ to the index of failedVMList and failed host h to the index of failedHostMachineList
```

V. IMPLEMENTATION AND RESULTS

Cloudsim is chosen as a simulation tool for the above mentioned algorithms. Cloudsim is a reenactment stage that can be reached out by clients to test their strategies, approaches and instruments for overseeing cloud frameworks.

CLOUDLET ID	STATUS	DATA CENTER ID	VIRTUAL MACHINE ID	TOTAL TIME	START TIME	FINISH TIME
0	SUCCSESS	01	0	323.3	0.1	323.4
5	SUCCSESS	01	0	323.7	0.1	323.8
10	SUCCSESS	01	1	324.1	0.1	324.1
1	SUCCSESS	01	1	324.5	0.1	324.6
6	SUCCSESS	01	2	324.9	0.1	325
2	SUCCSESS	01	2	325.3	0.1	325.4
11	SUCCSESS	01	4	324.7	0.1	324.8
7	SUCCSESS	01	4	325.1	0.1	325.2
4	SUCCSESS	01	5	325.5	0.1	325.6
9	SUCCSESS	01	5	325.9	0.1	326
3	SUCCSESS	01	3	326.3	0.1	326.4
8	SUCCSESS	01	3	326.7	0.1	326.8

Table I

The VM list and the Jobs list are initially put up in the broker. The dedicated broker then allocates the tasks to the required VMs dependent on occupations request, cost and VMs accessibility parameters. Each one of the tasks are planned and completed effectively on the six Virtual Machines and the simulation results are stored in the table I as displayed. It indicates attributes of all tasks which has run splendidly fine with no faults and getting totally executed with conclusive success status.

The simulation results after providing the algorithm with faults which can be detected by it.

CLOUDLET ID	STATUS	DATA CENTER ID	VIRTUAL MACHINE ID	TOTAL TIME	START TIME	FINISH TIME
0	SUCCSESS	01	0	323.3	0.1	323.4
5	SUCCSESS	01	0	323.7	0.1	323.8
10	SUCCSESS	01	1	324.1	0.1	324.1
1	SUCCSESS	01	1	324.5	0.1	324.6
6	FAILED	01	2	0.1	0.1	0.2
2	FAILED	01	2	0.1	0.1	0.2
11	SUCCSESS	01	4	324.7	0.1	324.8
7	SUCCSESS	01	4	325.1	0.1	325.2
4	SUCCSESS	01	5	325.5	0.1	325.6
9	SUCCSESS	01	5	325.9	0.1	326
3	SUCCSESS	01	3	326.3	0.1	326.4
8	SUCCSESS	01	3	326.7	0.1	326.8

Table II

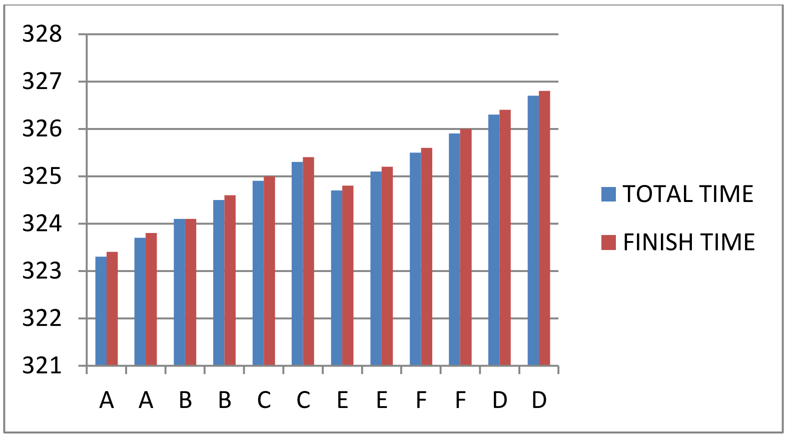

Figure 2: Comparison between the total time and finish time

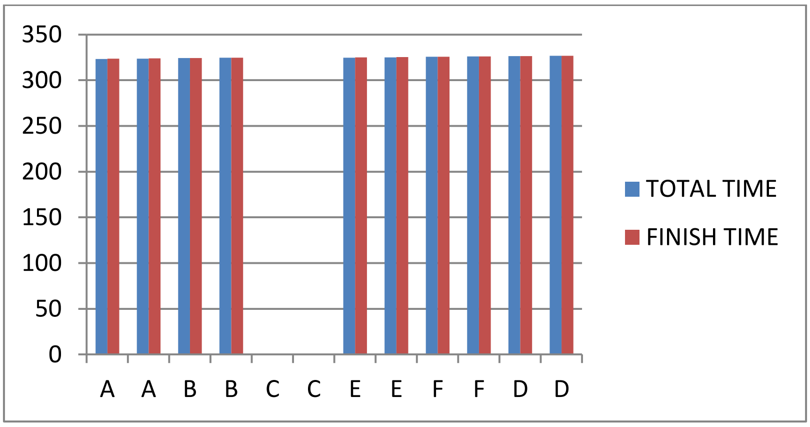

Figure 3: Comparison between the total time and finish time with faults

VI. CONCLUSION AND FUTURE WORK

Early location and relief of flaws in the distributed computing condition is an indispensable issue. On the off chance that the deficiencies go undetected, at that point they may transform into lethal disappointments. It might prompt loss of business, and thusly legitimate activities may transform into weighty fiscal punishment and loss of validity for cloud specialist organizations. Continuously evaluating the frameworks, the accuracy of results depends on the logicality of the outcomes as well as on the time at which the outcomes are created. We have identified the faults that were detected the algorithms that have been discussed in this paper. The identified Faults can be further mitigated to the nearest neighboring virtual machines.

REFERENCES

[1] Anju Bala and Inderveer Chana "Fault Tolerance-Challenges ,Techniques and Implementation in Cloud Computing", *2015 IEEE 23rd International Symposium on Quality of Service (IWQoS)* 2015, June

[2] J. Huang , C. Wu and J. Chen, "On high efficient cloud video transcoding", *2015 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, 2015,November

[3] M. K. Gokhroo , M. C. Govil and E. S. Pilli, "Detecting and mitigating faults in cloud computing environment", *2017 3rd International Conference on Computational Intelligence Communication Technology (CICT)*,2107,May

[4] J. Lee , H. Han and M. Song, , "Balancing Transcoding Against Quality-of-Experience to Limit Energy Consumption in Video-on-Demand Systems", *2017 IEEE International Symposium on Multimedia (ISM)* 2017, December

[5] P. Guo and Z. Xue, "Real-time fault-tolerant scheduling algorithm with rearrangement in cloud systems" *2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*,2017,May

[6] H. Han , W. Bao , X. Zhu , X. Feng and W. Zhou, , "Fault-Tolerant Scheduling for Hybrid Real-Time Tasks Based on CPB Model in Cloud", *IEEE Access*, 2018

[7] D. Pak , S. Kim , K. Lim and S. Lee,"Low-delay stream switch method for real-time transfer protocol", *The 18th IEEE International Symposium on Consumer Electronics (ISCE 2014)*, 2014, June

[8] I. P. Egwutuoha, S. Chen, D. Levy and B. Selic, "A Fault Tolerance Framework for High Performance Computing in Cloud," *2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012)*, Ottawa, ON, 2012, pp. 709-710.

[9] B. Mohammed, M. Kiran, I. Awan and K. M. Maiyama, "Optimising Fault Tolerance in Real-Time Cloud Computing IaaS Environment," *2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud)*, Vienna, 2016, pp. 363-370.

[10] I. P. Egwutuoha, S. Cheny, D. Levy, B. Selic and R. Calvo, "Energy Efficient Fault Tolerance for High Performance Computing (HPC) in the Cloud," *2013 IEEE Sixth International Conference on Cloud Computing*, Santa Clara, CA, 2013, pp. 762-769

[11] B. Song, C. Ren, X. Li and L. Ding, "An Efficient Intermediate Data Fault-Tolerance Approach in the Cloud," *2014 11th Web Information System and Application Conference*, Tianjin, 2014, pp. 203-206

[12] J. Soniya, J. A. J. Sujana and T. Revathi, "Dynamic Fault Tolerant Scheduling Mechanism for Real Time Tasks in cloud computing," *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, Chennai, 2016, pp. 124-129.

[13] J. Fan, R. Li and X. Zhang, "Research on fault tolerance strategy based on two level checkpoint server in autonomous vehicular cloud," *2017 7th IEEE International Conference on Electronics Information and Emergency Communication (ICEIEC)*, Macau, 2017, pp. 381-384.

[14] S. Antony, S. Antony, A. S. A. Beegom and M. S. Rajasree, "Task Scheduling Algorithm with Fault Tolerance for Cloud," *2012 International Conference on Computing Sciences*, Phagwara, 2012, pp. 180-182.