Journal of Theoretical and Applied Information Technology
30th June 2019. Vol.97. No 12

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

SIMULATING FAULT TOLERANCE IN CLOUD COMPUTIN BY MIGRATION USING CLOUDSIM SIMULATOR X

GHAITH AL ROSTOM

Research Scholar, SY

SYRIARIAN VIRTUAL UNIVERSITY, MASTER WEB SCIENCE,

E-mail: ghaith_59687@svuonline.org

ABSTRACT

The concept of cloud computing refers to the resources and computer systems available over the Internet through which we can provide a number of integrated computing services without being restricted to existing local resources, but the dynamic nature of cloud computing can suffer from many faults that affect the quality of service provided to the customer. The ability of the system to respond and deal with these faults determines its ability to tolerate faults and thus affect the quality of service provided.

The main objective of this research is to achieve faults tolerance by using the CloudSim simulator, by using a random model that generates faults in processors (PE) in a virtual machine (VM), and then applying a proposed algorithm in this search which migrates the existing Cloudlets from the disabled virtual machine on a host to another virtual machine on another host.

In the first stage, the fault generation was achieved within the CloudSim simulator, and in the second stage, the faults were processed by implementing migration algorithm. The results showed a high reliability and availability is relatively good when the number of performed cloudlets multiplied tenfold.

Keywords: *Cloud Computing, Fault Tolerance, Fault Detection, Cloudsim Simulator.*

1. INTRODUCTION

The cloud computing is considered as a Technology that depending on converting processing and storage space of computer into what is called: cloud, which is comprising servers machines accessed via internet, programs are converted from being products into services, and thereby, the users can access through internet, without the need to have knowledge and skills and so of controlling equipment.

The fault tolerance of the cloud computing is a vital one that should not be ignored as it provides availability of high service performance and reliability of the cloud. Many researches, currently, are working on achieving a satisfactory fault tolerance. We can tolerate with faults of the system good performance in case of failure even when it low instead of full failure. However, the fault tolerance could not be achieved through expecting the exceptional conditions and on how to deal with them, but it aims at auto-stability rather free from any faults.

The tolerance of faults tries to regard the critical and high availability system particularly, where it indicates its ability to preserve the cloudlets, when parts of the system collapse and then, it could be called as the sympathetic collapse.

If the consequences of the system failure was catastrophic, then, it would be able to be recovered to the Safety mode. The safety failure that designed to deal with defaults in case of total or partial failure, would protect people or properties from injuries or damage, via the safe exit instead of having catastrophic failure that saves the data deterioration in case of failure and so the distinction between good failure and bad one would be possible.

The designed system that ensures a safe failure at a low performance level when the components have failed, for example, at power outage, the building power would work on the low power level instead of complete breakdown, another example, when the frequency range width is insufficient for playing a video on the internet, a copy of less quality would be broadcast instead of

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the high quality one, so in order to ensure tolerance there must be flexibility and reliability of the equipment and software.

The failure of data centers for cloud computing is normal and not an exception and therefore in order to provide a good level of service for cloud computing users must use appropriate technologies to deal with failures.

We have conducted a practical investigation of two famous technologies for dealing with faults

job migration and replication.

These two investigations were included in the Cloudsim Simulator and the tests needed to ensure reliability and availability were considered to be very important to the user.

Building on this work and developing technologies that help cloud service providers improve the level of service agreed upon with the customer.

This work is a part that can be built on to propose a fault tolerant model and compare it with fault tolerance models in the following cloud computing:

AFTRC, which operates according to a -
proactive real-time tolerance scheme.

-LLFT, which tries to solve the cloud computing problem that applications on the cloud are constantly working to provide services to users.

-FTM This model has been proposed to overcome the limitations and methodologies for on-demand service.

-FTWS This model was proposed to schedule tasks by using redundancy and re-executing tasks on a priority basis.

-Candy gives this form a high availability.

-Vega-warden This model provides a unified user management system that is easy to use and highly secure.

-FT-cloud This model combines proactive and interactive tolerance techniques .

magic-cube This model is an iterative storage structure that provides high reliability and performance

Tolerance systems with high transparent faults will warn the users in case a failure occurred in one of the elements even if works continued on a high level and thus to avoid a complete unexpected breakdown and this would help to easier troubleshooting of the main problem and so to find the required solution and ensuring no reoccurrence.

2. RELATED WORKS

A group of researchers have proposed in [1] a mechanism for fault generation by using CloudSim simulator through depending on the random distribution that specify the period of time for the fault generation. The researchers have conducted tests on the could environments that designed for the purpose in order to test the effects of the random statistic distributions used on the faults generated. As well as, researchers in [2] depending on [1] and applying the same method in fault generation, however, it focused on simulation more realistically through setting necessary restrictions on simulation.

Many researchers, recently, have proposed methods and algorithms for achieving tolerance with faults, research [3] proposed an algorithm determines the way of dealing with the disabled machines in an environment of a highly changeable cloud.

Research [4] proposes an interactive method for faults tolerance through testing the system in different time periods and returning at faults occurrence. The researcher has achieved all the determined cloudlets for this research where this method omits all the test points during the implementing of the virtual machine that improved its performance.

In [5] the researcher, also, proposed a method of fault tolerance in order to retain high reliability and availability of the cloud through depending on the virtual machine of best performance and keeping it for processing the unexecuted cloudlets later, where he applied these techniques on CloudSim simulator.

The researcher in [6] achieved an algorithm by using load distribution on host taking into consideration the tolerance level with faults on the account of reliability and availability through

working on migrating the virtual machine (VM).

We have developed the CloudSim simulator so that it will be able to generate faults depending on a random faults generator. The generated faults disable the Processors (Processing elements (PE) in a virtual machine (VM) as it will proceed in

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

executing the cloudlets of the virtual machine (VM) until another processors is disabled on the account of implementation time in order to reserve the high reliability and availability of the cloud. When all processors (PE) in the virtual machine (VM) are disabled on the virtual machine (VM), the proposed method will migrate all the cloudlets to another host and then implement it from the beginning. By using this method a high reliability with a high availability will be provided relative to the implantation time and mean time of response for the generated faults of the cloud.

Figure (1) shows the General Structure Of The Applied System, where the broker will transmit one or more of the cloudlets to the datacenter which in turn will tabulate the these cloudlets according to the determined tabulating policy on the Host. Each part of the CloudSim simulator will send a definite event to the other part. By doing so, the faults generator will send a message to the data center, and consequently it would be informed of any fault that might occur in the system.

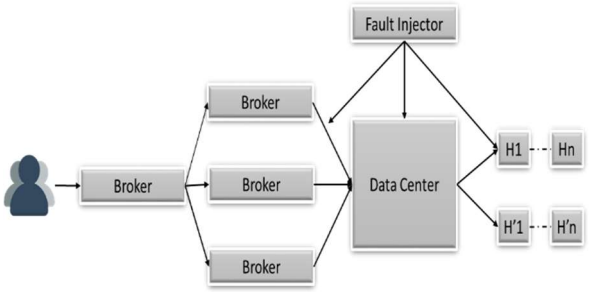

Figure (1) the General Structure of the Applied System

One of the most important properties of the main unit generating the faults is that it generates the faults according to the random distribution calculation, whether it is detached or continuous. The faults generator is the part that will be available for the whole simulation period, in an attempt to input faults upon the calculation chosen for generating random numbers.

3. THE RESEARCH PROBLEM:

Due to the dynamic cloud environment which causes the cloud existed simulators to be unable to avoid and process consequent fault, such problems are experienced, and it results in the decline of service quality and reliability rendered to the customer. And thus the fault processing would be one of the greatest challenges in providing a cloud system tolerant with faults.

4. OBJECTIVE OF THE RESEARCH:

This research aims at achieving faults tolerant by using the CloudSim simulator, through generating faults in the cloud that causes disability of Processors PE and virtual machines. The generation of these faults is made, at the first stage, by using a random generator determines the proper time for faults generation. Dealing with the generated faults, in the second stage, is made by migration of the disabled virtual machines with the cloudlets to another host that ensure the development of the service quality rendered to the customer besides achieving good availability and high reliability.

5. CLOUDSIM SIMUATOR:

The actual tests of the CloudSim hinder the experiments tests and causes the results obtaining to be very difficult mission. Another substitute is to use the simulation tools that allows for the possibility of evaluation of the proposed hypothesis before developing the software in an environment in which tests could be evaluated once and again. The availability of approaches based on simulation have great benefits at clouds computing, because they enable the customers to test their services in a repetitive environment without any cost, and consequently control the performance by the applications of the real clouds. The available simulation environment allows for evaluating several of scenarios for resources renting under variable distribution of loading and pricing, such studies might help the service providers improve the accessibility cost to the resources along with focusing on profits increase. In the absence of such simulation platform, the brokers and providers.

Of loud services to depend either on the inaccurate theoretical evaluation or on approaches of experiment and errors which lead to unreliability of the service and revenue generation [7].

The CloudSim simulator is used for the purpose of the cloud computation simulation that allows for modeling and experimenting on the structure and applications and cloud computing services. The development of this simulator has been performed in the cloud computing laboratories of Melbourne University, where several basic functions are supported such as events processing,

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

construction of cloud system structures (services, host, data center, broker…etc.)

The main target of the CloudSim is to provide a general simulator frame that enable the modeling, simulation and experimenting the infra structures of the existed cloud computing besides the services applications very smoothly.

The researchers and developers were able through using the CloudSim simulator, to focus on issues as the specified system design they targeted to achieve without any worries regarding the low level relative to the infra-structure and the cloud services available, or in other way round to be able to achieve simulation for a real components and consequently help to understand any cloud pattern without the need to apply it on reality along with the study of all the specifications and variety components [8].

Figure (2) clarifies that the CloudSim simulator is comprising for layers, as the data center consists of the host servers and storage servers. The host server can host one or more servers of the virtual machines. The datacenter will allocate virtual machine to host cloudlets basing on the adopted method in the cloudlets tabulating. All methods can be re-identified by the user.

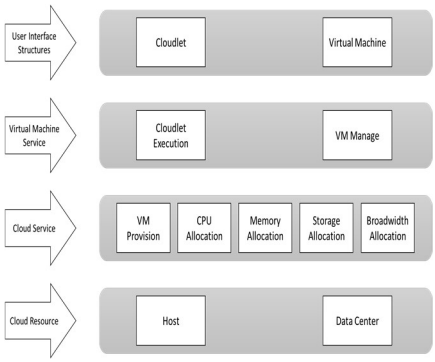

Figure 2. the box diagram of the CloudSim simulator

Some modifications have been made in this research, on the CloudSim simulator environment in order to perform the proposed algorithm experiment in the research according to a virtual scenario.

Figure 3. clarifies the box diagram of the CloudSim simulator with some modifications besides adding fault generator

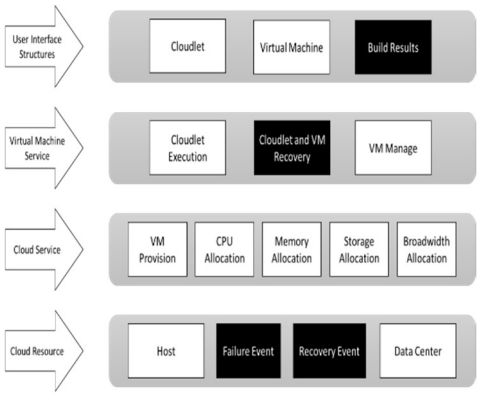

Figure 3. the box diagram of the modified CloudSim simulator

6. FAULT DETECTION

Fault detector is an application or a system used for detecting any failure or fault in the system which can be classified as effective or ineffective according to the given results, so if the result comes from the fault detector accurately will be an active fault detector, and the fault detector will be inactive in case the result is inaccurate or when the result takes a long time to be detected the wrong process which is consequently lead to wrong results in the failed cloudlets processing. Tolerance with fault is required [9].

1. In case of complete breakdown in the system in case of the failure of any component even if it is a single one and work will be on even during repair.
2. Isolating the fault in the component at failure: the system shall be able to isolate the fault, this requires adding some fault detection mechanism, that is not only isolating the fault, in addition to some mechanism to overlapping or refreshment.
3. Containing and restricting the fault spreading: some faults causes system failure by the fault spread in other components of it, this requires protection mechanism for the system parts and isolation mechanism in case of failure.
4. The tolerance system should be flexible to the expected and unexpected faults, because such systems are estimated according to application but not only to machine.

7. FAULT TOLERANCE TECHNIQUES:

Many studies have been made about the tolerant systems with fault during the recent years which

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

resulted in achieving new strategies for new benefits and obstacles overcoming with the fault tolerant systems, therefore, hereunder, some techniques used for achieving fault tolerant [10]:

- Retry method: this method is the simplest used one that depends on retry of the cloudlets for several times at fault occurrence and at same source.

- Migration job: this happens when a certain function or a certain change in the system unable for implementation, when failure occurs, migration will be performed to another machine by using a proxy in order to implement it instead of the entire restart.

- Check pointing: this technique is one of the important applications that check any changes in the system regularly and in case of job failure it will restart again, and it will restart from the chosen check point. This is a distinct cost activity and requires a less infra-structure with repetition techniques that make it a good choice for systems. This technique suffers from the high complexity as for availing correct and active check points.

- Replication: is considered one of the strong technique used in open servers, different job duplications are copied and in case a failure in one of them, the spare copy would be used as an ideal substitute since it is applied, at a stage, as the same original copy.

- Self-healing technique: when the jobs number is high, this technique divides it to parts on the basis of disperse is best. These parts have high performance as it easily detect faults with all different application cases.

- Safety technique: These techniques hide orders that do not achieve the safety procedures.

8. STANDARDS USED FOR THE EVALUATION OF FAULT TOLERANCE TECHNIQUE:

The fault tolerance technique in cloud computing are evaluated through the following variables [11]:

- Response time: is the time required for responding via a certain technique whose value should be at its least.
- Expansion liability: is the liability of the system for fault tolerance with any given number of machines.
- Productivity: is used for the calculation of the executed jobs for enhancing the performance of the system that should be high.
- Availability: represents the possibility of the component work in a satisfactory way at a definite point and with a specified time under a stipulated conditions that is the ratio between the taken time for operation and the total operation times with the system off:

The availability range can be defined as the following relation [11]:

$$Availability = \frac{Uptime}{Uptime + Downtime} \quad (1)$$

Availability measurement can be conducted as a percentage between the agreed time of service and stopping time. Where the agreed time of service is the system operation time expected monthly, excluding the stoppage time as planned for the system and clearly calculated from the agreed service time, however the calculation of the system availability range can be made as [11]

$$Availability(\%) = \frac{AgreedServiceTime - Downtime}{AgreedServiceTime} * 100 \quad (2)$$

The range of the system availability also can be determined by using the following [11]:

1- Mean time of failure (MTTF)
2- Mean time between failures (MTBF)
3- Mean time to repair (MTTR): the system availability can be calculated as for MTBF and MTTR can be made as [11]:

$$Availability \frac{MTBF}{MTBF + MTTR} \quad (3)$$

- Reliability: is aiming at giving correct or reasonable results during a period of

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

time determined by environment, formation of the service reliability can be made as the following [11]:

$$Reliability \frac{Successful\ Responses}{Total\ Requests} * 100 \quad (4)$$

9. RESEARCH TOOLS AND METHODS:

Java language is used within work environment (eclipse-neon-3) for writing the programming part special for the application of the proposed technique besides implementing the software parts that generates faults and processing the faults through migrating the disabled virtual machine and its cloudlets from one host to another.

10. THE USED MIGRATING TECHNIQUE

The failure generation is performed randomly for a processor of the hosts inside the data center. Usually the fault is constructed after the construction of the virtual machines in order to facilitate the determination of the virtual machine migration in which the fault occurred.

When a certain host processor fails and when the number of processors contained in the host are sufficient for all virtual machines no any procedures to be taken.

Considering (X) as the number of processor failed, that should be less than the total of the available processors. The (X) processor to be deleted periodically from the disabled virtual machines and consequently some virtual machines will be on with fewer processors. On the other hand, if the number of processors for a host after a failure is less than required for functioning the virtual machines, some of these virtual machines will be destroyed.

If all the processors are deleted from a virtual machine, they would be destroyed automatically and make a copy to be send to the broker, where the migrated copy can be implemented on another host, and doing so, each cloudlet worked on a virtual machine would be migrated and re-implemented from the beginning on the new host.

If a cloudlet was working inside a virtual machine that was affected by a fault, and this cloudlet required several processors (Y), and the virtual machine does not contain the required processors to execute the cloudlet, then, the cloudlet continues it implementation. However, it would need longer time to complete. For example, if the cloudlet required two processors and by the end of the failure, one processor remained, the cloudlet would need double time to complete the implementation.

Steps that should be followed can be summarized as follows:

1- Determine the required time for making a failure in a host by using random failure generator.
2- The host is to be chosen randomly by using random numbers generator that depends on the seeds used in the previous generator.
3- Determination of the host numbers, on which the failure will occur, shall be made randomly by using an internal generator.
4- The disabled processor will be deleted from the damaged virtual machine, then the virtual machines which do not have processor would be considered as failed.
5- The next failure will be determined in later time by using the previous generator.
6- This process will be repeated until the end of simulation. Figure (4) shows the flow diagram of the proposed migration technique

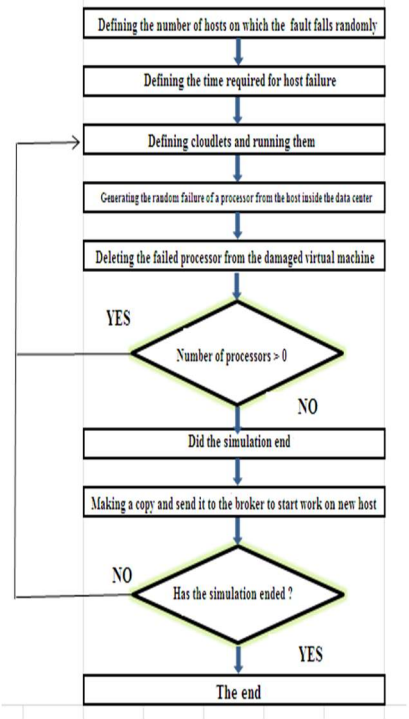

Figure (4) shows the flow diagram of the proposed migration technique

11. RESULTS AND DISUSSION:

After implementing the modifications on the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Cloudsim simulator in the first stage, and after having a platform able for tolerance with faults depending on migrating the disabled virtual machine with cloudlets given to it according to the previously proposed technique, then performing the second stage by testing the simulator tolerance modified with the faults according to the variables clarified in the following table:

Table (1) variable used in Simulation

Name of variable	Value
Number of processors in a processor (Host_PES)	4
Number of processors in a virtual machine (VM_PES)	2
Number of processors required for on (cloudlet-PE)	2
Number of hosts (Hosts)	50
Number of virtual machines (VMs)	8
Mean of failure generation per hour (MEAN_FAILURE_NUMBER_PER_HOUR)	0.02

The simulator is tested after the application of the technique depending on fixing the above variable values with changing the number of cloudlets that should be implemented. We hereunder explain all the events generated during the simulator's work when the number of cloudlets to be executed is twenty ones. Steps of application on the simulator are as follows;

1- The random generator will generate a fault and disable one processor on the host #H1 which contains the cloudlets #C2, #C10, #C18 . The remaining three processors are enough for the work of #C2 which is under operation and #C18. #C10 which are put on the waiting list , in the result work is still on and work will not be affected until all processors are disabled on the host as shown in the following figure:

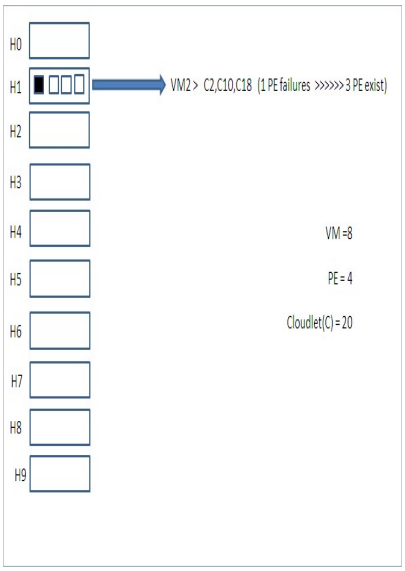

Figure (5) Firsts step of faults generation

2- In the following step of simulation, the random generator will generate the fault and disable two processors on the host #H7 containing cloudlets #C8, #C16, remaining two processors which are enough to run #C8 which is already running. And #C16 which is on the waiting list, so by the result the work will go on running without being affected by disabling all processors, as shown in the following figure:

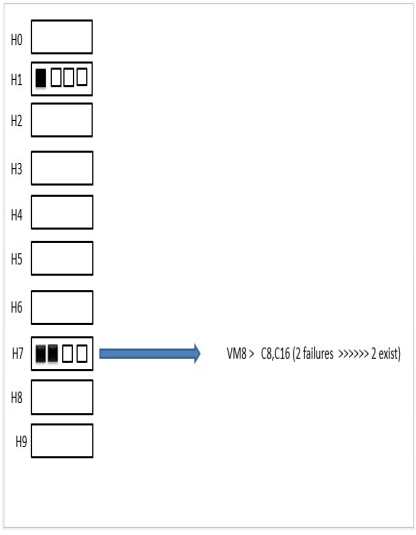

Figure (6) Second step of faults generation

3- In the following step of the simulation, the random generator generates a fault and disable two processors on the host #H7 that contains the cloudlets #C16 and # C8, only one processor remains to run #C8 which is already running and #C16 which is put on the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

waiting list and by result 3VM8 is to be migrated and given another Number #VM80 to the host #H8 and the cloudlets available on the destroyed machine to be copies to the new virtual machine and given new Nos. #C160 and #C80 as shown in the following figure:

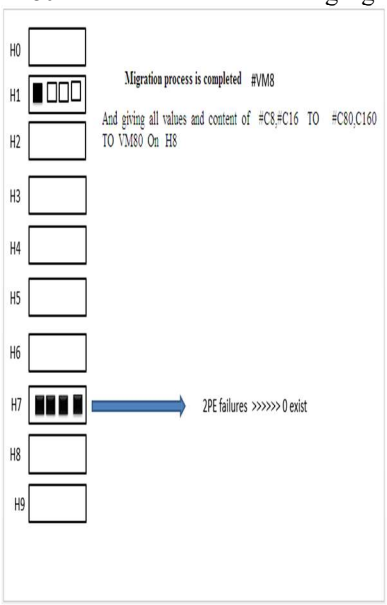

Figure (7) Third step of faults generation

4- In the following step of the simulation, the random generator generates a fault and disable processor on the host #H0 that contains the cloudlets #C17, #C9, #C1 three processors remain to run given the cloudlets of virtual machine and by the result the work still going on and not affected until all processors on it are disabled, as shown in the following figure:

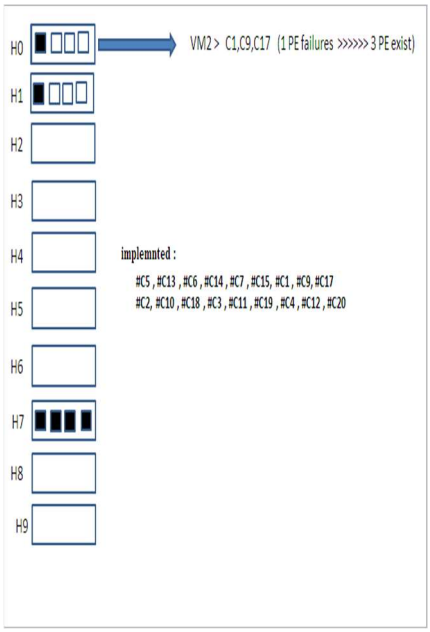

Figure (8) Fourth step of faults generation

5- The previous figure shows that all cloudlets were executed in this stage except the cloudlets on #H8 which are: #C160, #C80 .

6- Work still on and in the following step of simulation a fault and disabling of a processor is made on the host #H8 that contains the cloudlets #C160, #C80 and remains only three processors enough for run the cloudlets of the virtual machine and by the result work will continue to be on without being affected until all processors on it are disabled as shown in the following figure:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

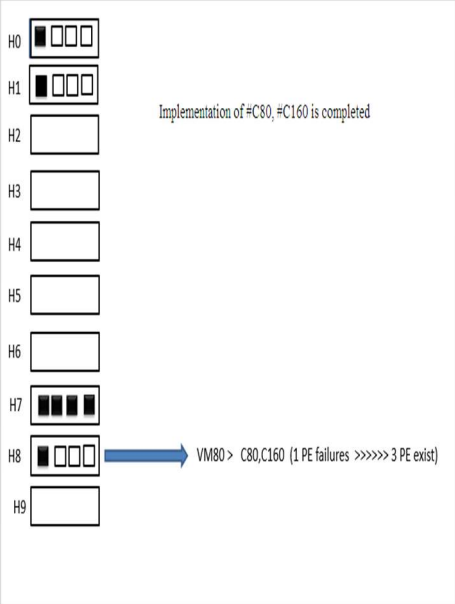

Figure (9) fifth step of faults generation

several consequent times with a change in the numbers of the executed cloudlets

And the results are recorded in the following table:

We note out of the above table that when the cloudlets are increased the availability rate and reliability rate are not affected, therefore the technique depended in migrating the cloudlets from a disabled virtual machine to another one existed on another host when executed cloudlets are increased, retains high reliability and good availability, and all faults are processed satisfactorily upon the hoists because any increase or decrease in the number of the hosts will lead to a change in the results, and, thereby, we would have achieved a fault tolerance in this research, this is shown in the following figure.

Table (2) the Experimental Results

Cloudlets	Migrated virtual machines	Faults	Implementing time (Hour)	MTTR (Minutes)	MTBF (Minutes)	Availability	Reliability
10	0	3	1166.4001	0.00	1166.4001	% 100.00	% 100.00
15	0	3	1166.4001	0.05	1166.4001	% 100.00	% 100.00
20	2	5	1862.1538	15.7500	1846.4038	99.1542%	% 100.00
25	1	7	2401.2354	15.7500	2385.4854	99.3441%	% 100.00
30	1	7	2401.2354	15.7500	2385.4854	99.3441%	% 100.00
35	1	7	2987.6232	15.7500	2971.8732	99.4728%	% 100.00
40	2	9	5241.1646	40.9167	5200.2479	99.2193%	% 100.00
45	2	9	5241.1646	40.9167	5200.2479	99.2193%	% 100.00
50	2	9	5700.7869	40.9167	5659.8702	99.2823%	% 100.0

7- The previous figure shows that all cloudlets were implemented in this stage except the cloudlets on #H8 which are: #C160, #C80, and thereby all cloudlets would have been implemented and so the simulation is completed.

Similarly when the execution of 20 cloudlets were discussed the simulator is ran for

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

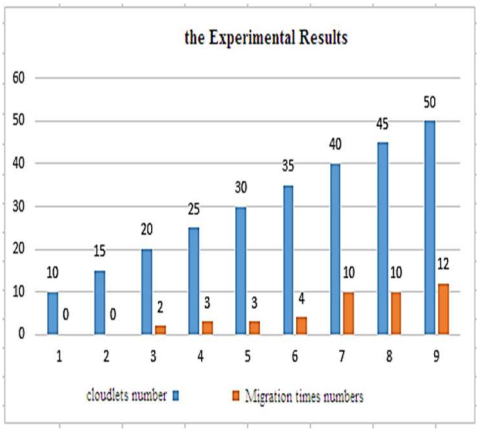

Figure (10) the Experimental Results

The following figure shows the Time of Implementation completion of /20/ cloudlets executed on a cloudsim according to the above example, where the cloudlets #C8, #C16 are the final cloudlets executed due to its migration from the primary virtual machine placed to it.

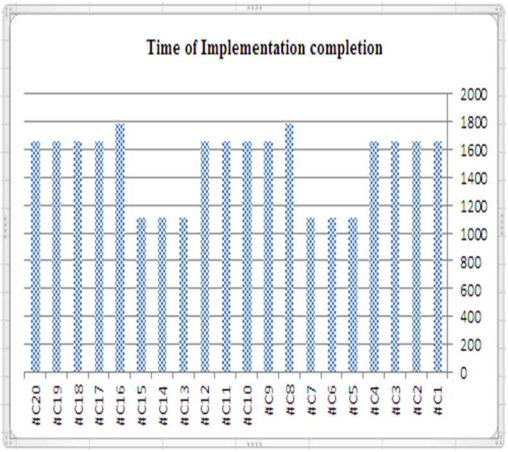

Figure (11) Time of Implementation completion

Figure (11) is the number of disabled virtual machines when (20) cloudlets are executed on a Cloudsim simulator according to the above experimental example, where it shows that no machine is disabled when the number of the cloudlets (10,15), and the number of disabled machines is only one when the number of cloudlets are (20,25,30,35) where the number of the disabled machines are two only when the number of cloudlets are (40,45,50)

12. CONLUSION AND RECOMMENTION:

The tolerance technique which is applied with the faults in this research, is the technique of Migration, where this research stressed on the benefit of using techniques that lessen the faults on the cloud computing and therefore ensuring the gain of high availability and reliability. We may state the following outlines as conclusions for this research:

1- The good quality of the service rendered to the customers in the cloud computing would be higher when faults tolerance is used.
2- The different methods and techniques help in finding various solutions of disforest faults and enable avoiding the unbeneficial improvement process conducted in the existing machines.
3- Steps of other adopted techniques can be used to achieve tolerance with the faults and applying it in the CloudSim simulator.
4- The proposed migration technique avail the best availability and the best reliability as long as the number of the cloudlets implemented are less than the hosts existed in the cloud.
5- The proposed migration technique is tested on the consideration of the existence of /1000/ cloudlets and still, the technique, keeps on giving a good availability and high rate of reliability, however it is on the account of the cost of increasing the hosts number.

We might state the following outlines as a recommendation for our work:

1- To work on modifying the proposed technique in order to solve the extra cost problem in case when the number of the executed cloudlets exceeds the number of the hosts existed in the cloud.
2- This research can be a proposal nucleus of many techniques to be studies on a CloudSim simulator or any other simulator.
3- Incorporating the applied migration technique with any other existed technique.

Achieving these two technologies helps build a fault tolerant model that achieves more goals than previously proposed models for fault tolerance.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REFERENCES:

[1] C. Mihaela, P.Florin, M. Mariana, C.Valentina, 2014-FIM-SIM: Fault Injection Module for CloudSim Based on Statistical Distributions, Journal of telecommunications and information technology, Vol.4.

[2] B. Adrian, N.Mihaela-Catalina, P. Florin, C.Valentin,2014-Cloud Simulation under Fault Constraints, Faculty of Automatic Control and Computers, University Politehnica of Bucharest, Romania.

[3] K.Jasbir, K.Supriya, 2014-Efficient Algorithm for Fault Tolerance in Cloud Computing, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol(5).

[4] Kalanirnika G R, V.M.Sivagami, 2015- Fault Tolerance in Cloud Using Reacive and Proactive Tecniques, Vol.3. Issue 3, P1159-1164.

[5] S.suguna and K.Devi, 2016- VMFT:virtual Machine Fault Tolerance In Cloud Comuting , Vol.22 , No.2 , PP(256-265).

[6] G.Gayathri, R.Latha, 2017, Implementing a Fault Tolerance Enabled Load Balancing Algorithm in the Cloud Computing Environment, Vol.5, Issue.

[7] C.Rodrigo, R. Rajiv, B. Anton, D. C´esarAF, and RajkumarBuyy , 2011- Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and Experience, 41(1):23–50

[8] C.Rodrigo, R.Rajiv, R.C´esarAF, and RajkumarBuyya , 2009 - Cloudsim: A novel framework for modeling and simulation of cloud computing infrastructures and services.

[9] Arif Sari, Murat Akkaya,2015- Fault Tolerance Mechanisms in Distributed Systems, Int. J. Communications, Network and System Sciences, Vol (8),PP (471-482).

[10] M.Seyyed and A.Mostafa, 2015- Fault-Tolerance Techniques in Cloud Storage: A Survey International Journal of Database Theory and Application , Vol.8, No.4 , p183-190 .

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