THINK INDIA JOURNAL

ISSN: 0971-1260
Vol-22-Issue-16-August-2019

A Brief Survey on various Fault Tolerance Techniques in Cloud Computing Paradigm

Umer Iqbal Wani
Assistant Professor, Department of Computer Science and Engineering, Lovely Professional University, Phagwara, Punjab, India

Ankita Arora
Assistant Professor, Department of Computer Science and Engineering, Lovely Professional University, Phagwara, Punjab, India

Abstract

Reliability is the cause of concern during execution job over the cloud. Reliability enhancement strategies have been researched over that provides mechanism to conserve cost during allocation of job to virtual machines. Data Centre resource consideration is critical in allocating jobs that require prior knowledge of resources required by jobs. To ensure reliability, fault tolerant allocation strategy must be created. This paper presents comprehensive analysis of techniques used to ensure effective allocation off jobs that enhances reliability. Primarily this paper studies and surveys the techniques of checkpointing and replication for the sake of ensuring fault tolerance in cloud computing. Parameters considered for reliability includes mean time between failure and cost. Parametric comparison presented indicates adaptive framework including multiple strategies or hybridization better performance.

Keywords: Reliability, Cost, Mean time between failures, checkpointing, replication

1.Introduction

Cloud computing is consists of different past technologies(Boru et al. 2015) like "grid computing", "distributed computing", "utility computing" or "autonomous computing". Cloud computing gives an easy access to various resources. Cloud computing includes the feature of scaling up as well as scaling down. Cloud computing is a broader term that gives access to services using internet. Reliability has to be taken into consideration using various cloud services. Reliability has a feature that a given thing will finish off its work under a given time frame and taking into consideration all the limitations(Jain & Inderveer 2016).Cloud reliability assures the provision of services even when few parts of the cloud infrastructure aren't working fully. An error free cloud isn't a possibility always.A number of failures arise in a cloud paradigm ,for example, failure in overflow conditions,failure in network ,failure of hardware , failure in software , failure in timing out , missing resources failure, and failure in database(H. Goudarzi 2012).The reliability of the cloud computing is exceptionally basic yet difficult to break down on the grounds that cloud is made up by the mixture of different variables like wide - territory systems, heterogeneous software/hardware segments. There are numerous convoluted co-operations among the different segments of the clouds. Consequently, the reliability models that are characterized for unadulterated software/hardware or customary systems can't be essentially connected to consider and assessing the cloud reliability. (Comput et al. 2012)

In spite of the fact that cloud computing has given an achievement solution for our computing needs these days, reliability issues is as yet a testing things to be solved.(B. Javadi, J. Abawajy 2012) A few associations are as yet running with conventional computing services rather than cloud computing services so as to remain in the safe side. These demonstrating that a few people still not have confidence for cloud services. To overcome this reliability issues there are number of techniques like checkpointing(Singh et al. 2012), replication etc. In checkpointing technique reliability of the cloud can be ensure by providing the data till checkpoint in case of data failure. In any case of data loss the data can be recovered till checkpoint from the log files etc. (Jin et al. 2008)This technique provides a great level of assurance to the user that the data is secure and user can rely on the data that they had saved on the cloud.
Furthermore to enhance the cloud reliability we have some parameters like MTBF(Perspective et al. 2012), Execution Time, Cost, degree of fault tolerance etc. if we enhance these parameters then we can achieve reliability in good manner. The parameters that can be considered for cloud reliability are given below:

ISSN: 0971-1260
Vol-22-Issue-16-August-2019

MTBF-Mean Time Between failure (MTBF) is the predicted elapsed time between failures of mechanical devices like our virtual machines during normal operations. MTBF can be calculated as the arithmetic mean time between failures of a system. To make our cloud reliable a virtual machine with more MTBF will be choose for replication of data.(Pinheiro et al. 2007)

Execution Time- Execution time is the total time taken by a machine to accomplish the task. For reliability in cloud the virtual machine with low execution time will be chosen among all for execution of task.(Yadav & Kushwaha 2014)

Cost-Cost parameter is also plays an important role in the reliability of cloud. A virtual machine that consumes less cost is better than the one with high cost. So to accomplish the task a virtual machine which takes less cost will be chosen.(Rana 2014)

Degree of Fault Tolerance- Cloud is reliable if it provides Fault tolerance and versatility to handle the tasks or resources in case of failures. If the fault tolerance degree will be decreased or the resources are handled without any failure then we can say that our cloud is reliable.(Perspective et al. 2012) The virtual machine with possible management of fault tolerance will be chosen to achieve or accomplish the task on cloud.

1.1 Cloud Computing Models

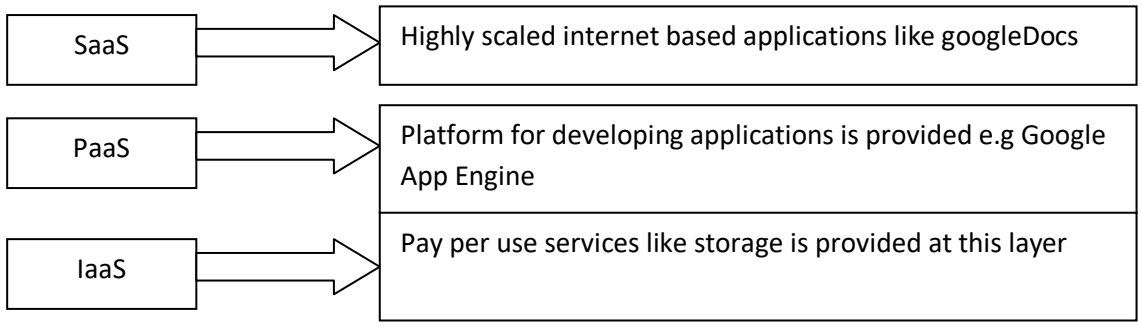


Figure 2: Service Models provided by cloud

Cloud services can be classified as:

1. Software as a Service (SaaS):A full-fledged,ready to use application is provided to the user[14].The application runs centrally on a cloud platform and is accessed by a number of users at the same time[15][16].The customer doesn't spend anything for the development phase of the application, the customer pays only for the usage.Various companies offering SAAS include - Google, Salesforce, Microsoft, Zoho, etc.

2. Platform as a Service (PaaS): In this model, a user is provided with a development environment where he/she can develop applications using the resources provided. in the platform[17]–[19].The end user uses the platform to create applications of his/her own choice[20].PaaS providers offer a blend of various OS's and servers to meet the scalability requirements of the application. Some of the popular PaaS platforms include LAMP platform (Linux, Apache, MySql and PHP), restricted J2EE, Ruby etc. Googles App Engine, Force.com, etc.

3. Infrastructure as a Service (IaaS): IaaS mostly includes provision of infrastructure to the end users. The infrastructure is accessed by the end user mostly for computing and storage purposes[21].IaaS forms the base of various cloud services, all the other cloud services are built on top of IaaS[22],[23].The customer uses the infrastructure for his own specific purposes. The provider is responsible for security of the infrastructure only and not how the user puts the infrastructure in use. Famous IaaS providers include Amazon, GoGrid, 3 Tera, etc.

1.2Cloud Computing Benefits

[31]In order to fully exploit various cloud architecture models, organizations should modify their applications according to the environment. The various cloud computing benefits are listed as:

ISSN: 0971-1260
Vol-22-Issue-16-August-2019

1. Reduced Cost

[32]Cloud computing paradigm offers services on pay-per-use model. The users are not spending on any upfront investment. The maintenance of the services doesn't concern the end user. The user just pays for what he/she uses. Buying services from the cloud providers is always the best possible option for start-ups because of a negligible up-front cost.

2. Increased Storage

[33]Cloud offers unlimited storage to the applications. Based on the need and demand, the storage can be increased as well as decreased.Their is no threat of loss of data or other important user related content in the cloud.The user can get storage on demand basis, anytime, anywhere. The user is least bothered about the workload spikes, as things are managed in a dynamic way.

3. Flexibility

[34]In the cloud paradigm, flexibility is mostly related to the resource scheduling. Enterprises, nowadays have to respond to changing business conditions at a faster pace, because of an increasing customer base. Delivering services on time and within the deadline has become essential for all kinds of organizations. Cloud computing promises businesses to deliver on time and to adapt according to market needs.

1.3 Cloud Computing Challenges

[24]With such a growing and ever-increasing impact, a lot of concerns have to be considered related to the cloud computing model. The various challenges faced are:

1. Data Protection

Data Security assures scrutiny of various resources. Organizations, mostly, don't depend on the vendors assurance of business security[16].The fear involved is mostly related to the user data confidentiality. In most cases the location of data is hardly known to the service user, thus raising valid concerns of misuse.Current models make use of firewalls across data centres to provide access to only authenticated user base. In the cloud model, organizations have to entirely rely upon the service provider for all security related issues.

2. Data Recovery and Availability

[27]Businesses all around the world need to follow service level agreements which are policies for usage of various services provided by an organization. The various SLA's are managed and governed by an operational team within an organization. In anenvironment related to production, the support provided by operational teams include:

- Clustering and data replication fail-over.
- Monitoring of various system properties like (Transactions monitoring, logs monitoring and others)
- Management and maintenance
- Recovery from disaster and management of performance

The above services, if not properly taken care off will lead to severe disasters.

3. Management Capabilities

[33]With such a large cloud provider base, the management part of the various cloud services is still in its initial stage. Auto-scaling has been one of the main issues currently under consideration. Improvements related to scalability and load balancing is a future prospect on which a lot of improvisation can be done.

4. Regulatory and Compliance Restrictions

[39]In a number of European countries, there is a strict prohibition for disclosing personal customer information, the providers aren't allowed to store customer specific information outside the state boundaries. To comply with

ISSN: 0971-1260
Vol-22-Issue-16-August-2019

the regulations, organizations aren't allowed to set up data centres outside the state provinces, this poses a big challenge to the cloud providers.

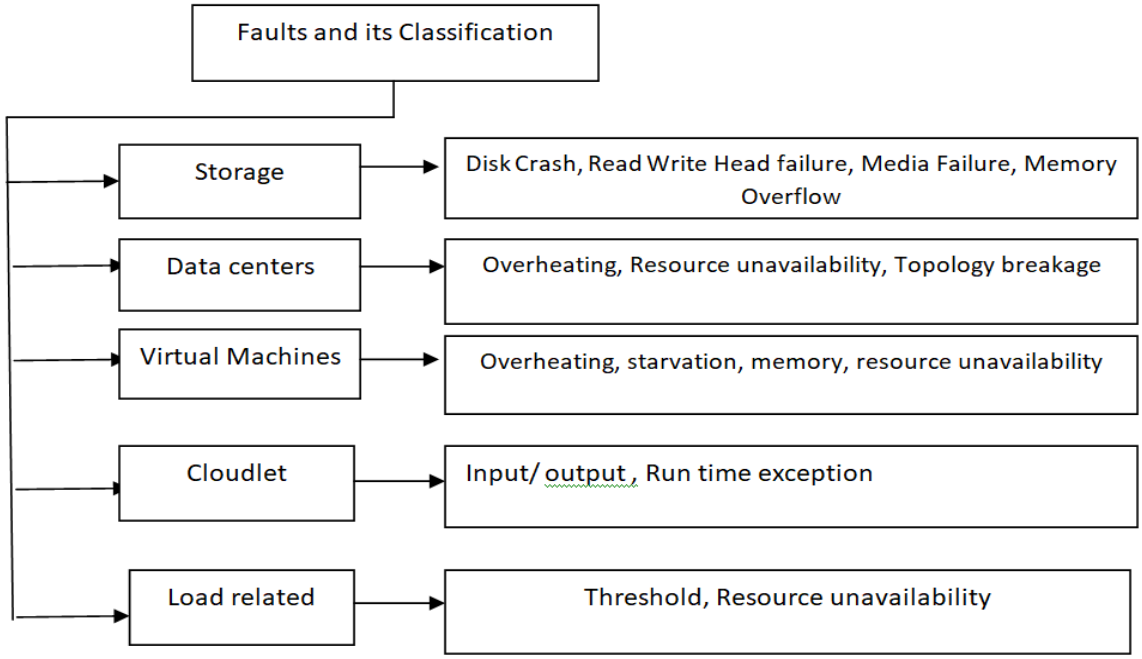
5. Faults and Failures

[40]Primary problem associated with cloud is faults and failures. Faults and failures cause the deviation from actual output. In case deviation is temporary than it is known as failure otherwise it is known as faults.

2. FAULT TOLERANCE IN CLOUD COMPUTING

2.1 Faults and Failures

Faults hamper the performance of the cloud. Cloud hence requires substantial implementation of tolerance strategies to tackle the issues of faults. Faults specific to cloud are discussed in this section.



Storage Faults:

Cloud provides storage facility for the user to be accessed on pay per use basis. [31]Storage holds the key for success of cloud service provider. [32]Storage can be faulted due to number of reasons. These reasons are listed as under

Disk Crash: in case disk head when touches the media can cause the disk crash. [33]Disk crash can be followed by media failure. All the data stored over the disk can be lost due to disk crash.

Read/Write Head failure: this type of fault falls under the category of hardware faults. In this case read/write head will be jammed and does not move to t he desired cell for reading or writing the data.

Media Fault: in this case the media on which data is stored is corrupted. The corrupted region is also known as pits. Media faults falls under the category of hardware faults.

Memory Overflow: in case data to be stored is more and memory capacity is limited then such faults occurs.

Datacenter Faults:

Datacenter is actually a physical machine hosting resources for cloud users.[17] Data centers due to heavy workloads can suffer from faults. Faults associated with data centers are partitioned into following categories

ISSN: 0971-1260
Vol-22-Issue-16-August-2019

Overheating: the physical components present within the datacenter may become heated causing performance degradation.Overheating can also cause automatic shutdown of resources causing delay in execution of jobs.

Resource Unavailability: resources are required by the jobs. In case resources required are more and resources are limited then this will cause unavailability and hence starvation is introduced within the system.

Topology breakage: Datacenters are connected through the topology. Topology breakage can cause transmission problem as cloudlet moves from source towards desired datacenter for allocation and hence performance degrades.

Virtual machine Faults:

Data centers within the cloud are divided into set of virtual machines. Faults can appear within the virtual machines by the application of load allocation and physical component of data centers failures. In other faults from data centers leads to the problems within virtual machines. The faults in VMs are further divided into following categories.

Overheating: the physical components present within the datacenter may become heated causing problems within the attached virtual machines. VMs thus cannot take the load anymore and other VMs becomes overloaded causing higher waiting time than normal.

Starvations: Such faults are result of unavailability of resources. The waiting time for job becomes higher in this case. Thus deadline oriented jobs cannot be executed in such environment. Constraint jobs thus are limited in such situation.

Memory: such faults are caused in case demand of memory is more and less memory is available. Memory overflow could be the result due to such problems.

Resource Unavailability: resources are required by the jobs. In case resources required are more and resources are limited then this will cause unavailability and hence starvation is introduced result in deadlock. In such situations jobs having desired number of resources cannot be executed in addition to the jobs having lack of resources.

Cloudlets Faults:

Cloudlets are tasks that are processed by the virtual machine. Cloudlet or tasks, if, are not in correct format can leads to system failures. Such faults are further divided into following categories.

Input/output

These faults are generally format oriented. Format of presenting the information to the cloud is fixed. In case that format is not followed than I/O faults appear.

Run Time Exception

Such faults appear when user gives wrong feedback or input. Division by zero, stack overflow etc. are generally falls under this category.

Load Related Faults:

Every Virtual machine is identified by the threshold value of load that it can tackle. [40]In case that threshold value exceeded then load related faults appear. Resource unavailability can lead to load related faults since load of faulted VM is to be shifted over to the next VM in sequence.

2.2 Fault Tolerance

As the cloud computing frameworks keep on growing in scale and many-sided quality, it is imperative to guarantee the steadiness, accessibility, and unwavering quality in such frameworks. The differing execution situations, expansion and expulsion of parts, escalated workload on servers, are the essential reasons that can create disappointments in powerful conditions of cloud computing. The unwavering quality of such frameworks can be effectively implied if the proactive measures are not taken against the conceivable disappointments in cloud frameworks. To guarantee dependability and accessibility of administrations running in the virtual

ISSN: 0971-1260
Vol-22-Issue-16-August-2019

machines a fault tolerant empowered load adjusting calculation is actualized in this paper. Sooner or later there might be numerous clients to use the server farm or free their assets to exit from the server farm. Along these lines, there would be lopsided burdens among the hosts in the server farm. To achieve stack adjusting, some VMs ought to be moved from the high-stack hosts to the low-stack has. The fault tolerance level of the administration is ensured by appropriating the VMs of the administration onto different physical hosts. Fault - tolerant level can be depicted as: if benefit I can work typically when it has separate , the fault-tolerant level of administration I is characterized as ki. To give dependable administrations, the fault-tolerant level ought to be guaranteed while VMs are moved to adjust loads. Before diving into the usage points of interest of the heap adjusting calculation we initially examine the difficulties and the related advances of a decent load adjusting calculation. The difficulties of load adjusting calculations for VM position on have relies upon the accompanying variables: Overhead, Performance.

2.3Requirement of Fault Tolerance in Cloud Computing

In spite of the fact that cloud computing has been generally received by the business, still there are many research issues to be completely addressed like fault tolerance, work process planning, work process administration, security and so forth. Fault tolerance is one of the key issues among all. It is worried about every one of the procedures important to empower a framework to endure programming faults staying in the framework. At the point when a fault happens, there are couple of methods which give instruments to the product framework to counteract framework failure event. The basic pros of taking fault tolerance into consideration in a cloud computing model include emergence from failure, data recovery, lowering of cost, improved execution measurements and so on. At the point when different cases of an application are running on a few virtual machines and one of the server goes down, there is a need to actualize an autonomic fault tolerance method that can deal with these kinds of faults.

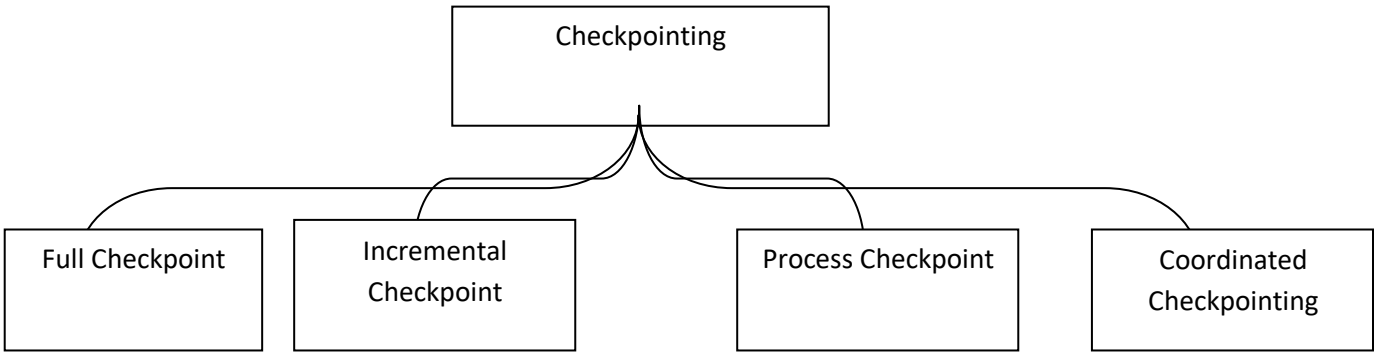
3. Fault Tolerance Strategies

A. Responsive Fault Tolerance

Reactive adaptation to internal failure strategies are utilized when a disappointment has happened in the framework. It helps in recuperation of framework state from an insecure state to stable state with the goal that the framework can again begin attempting to give wanted outcomes. [31]Different receptive adaptation to non-critical failure strategies are talked about beneath.

1) Restart: It works on program, or application level. In the event that an undertaking does not finish inside a given measure of time it is suspected to have fizzled and is, thusly, prematurely ended and restarted. The timeout after which we need to prematurely end and restart the undertaking must be deliberately picked on the grounds that on the off chance that it is too short then the errand may be prematurely ended just before consummation[15] while in the event that it is picked too long one must hold up pointlessly.

2) Checkpointing: It has a preventive part i.e. sparing a checkpoint and additionally a receptive part i.e. rollback recuperation. Checkpointing frameworks spare the framework state in customary or sporadic time interims. Upon disappointment the framework recuperates by moving back to latest checkpoint. The work performed since the latest checkpoint is lost with a disappointment. Checkpointing system is additionally named takes after.



ISSN: 0971-1260
Vol-22-Issue-16-August-2019

a) Full Checkpoint: It is an instrument in which checkpoints to a procedure which is pursuing are connected a settled interim of time which spares the condition of procedure to a few media. [59]On the off chance that there is a disappointment of process amid execution, it can be recuperated from the last spared checkpoint [24]state instead of restarting it. It ought to be viewed as that checkpoints ought to be connected after certain interim with the goal that it doesn't cost much overhead and increment its execution time.

b) Incremental Checkpoint: This system helps in lessening the checkpoint overhead by sparing those pages in which there has been any change as opposed to sparing the entire procedure.

c) Process Checkpointing-The objective of process checkpointing is t o spare t he current condition of a procedure.[34] In current HPC applications, a procedure comprises of numerous client level or framework level strings, making it a parallel application without anyone else's input. Process checkpointing strategies by and large utilize a coarse-grain locking instrument to hinder quickly the execution of the considerable number of strings of the procedure, giving them a worldwide perspective of its present state, and lessening the issue of sparing the procedure state to a successive issue.

d) Coordinated Checkpointing-Distributed checkpointing conventions utilize process checkpointing and message going to plan rollback-recuperation systems at the parallel application level. The objective of this convention is to manufacture a predictable conveyed depiction of the disseminated framework. [38]An appropriated depiction is an accumulation of process checkpoints (one for each procedure), and a gathering of in-flight messages (a requested rundown of messages for each point to point channel). The conventions accept requested misfortune less correspondence channel; for a given application, messages can be sent or got after or before a procedure took its checkpoint. [11]A message from process p to process q that is sent by the application after the checkpoint of process p however got before process q checkpointed is said to be a vagrant message. Vagrant messages must be stayed away from by the convention, since they will be recovered by the application, if it somehow happened to restart in that depiction.

3) Shadow VM's Replication: It is a procedure of making duplicates of comparable information and after that put away at various areas. Replication can be additionally delegated takes after.[31]

a) Semi-Dynamic Replication: Each reproduction is given info or state data, the essential copy and in addition reinforcement imitation perform execution on the information gave. Yet, the yield is given by the principle copy. [28]On the off chance that the primary imitation comes up short, yield is given by reinforcement copy. Adaptation to non-critical failure administrator (FTM) makes a comparable reproduction for every copy which has fizzled and refreshes its state. VMware's Fault Tolerance falls under semi-dynamic replication classification. [20]

b) Semi-uninvolved Replication: The principle imitation performs visit checkpoints over state data and by sparing information parameters in the middle of each of the checkpoints. Replication is then done by moving the data of its state to all the reinforcement copies. [37]Crafted by reinforcement replicas is to spare the most recent state acquired by essential imitation they don't play out any execution. [11]If there should be an occurrence of disappointment of essential imitation reinforcement copy starts and updates as essential reproduction with some loss of execution.[5]

c) Passive Replication: reinforcement is made on which state data of a virtual machine example is put away consistently. [12]In the event that there is a disappointment, adaptation to non-critical failure administrator decommissions another virtual machine occasion which re-establishes the last spared state. [8]The reinforcement can utilized for a specific application or it can be set up to share the condition of a few virtual machine examples, the virtual machine appointing procedure should likewise be possible with the assistance of need esteem which is doled out to each virtual machine case. VMware's High Availability arrangement is a case of this method.[8]

4) Job Migration: During disappointment of an undertaking the task which has fizzled can be moved to another machine.[14]

5) Task Bowing: If there is a fizzled energy recognized in a framework, the fizzled undertaking is resubmitted to another or same asset.

ISSN: 0971-1260
Vol-22-Issue-16-August-2019

6) User Defined Exception Handling: For a specific treatment of a disappointment of undertaking the client indicates a work process in this procedure

7) Deliverance Workflow: Even if a task comes up short safeguard work process enables it to proceed until or unless it isn't conceivable to execute without considering the fizzled assignment.

B. Proactive Fault Tolerance:

The proactive adaptation to internal failure strategies keeps away from different blames by anticipating them before they happen and supplant the speculated segment with new or non-defective segment before it really happen. Different systems for proactive adaptation to internal failure are talked about underneath.

1) Reinvigoration: It is issued before the framework falls flat. Suspicions are made regarding when the framework would come up short if no measures were taken. Preferably, the restoration interim would dependably end just before the framework falls flat. A traditionalist decision of the restoration interim will choose short interims. In any case, revival accompanies a cost of sparing the working condition and all procedures, restarting the framework and reinitializing the working condition and all procedures. On the off chance that revival is performed again and again the restoration cost aggregates pointlessly, if the framework is restored at too long interims it will frequently fall flat.

2) Self-Healing: Failures are taken care of consequently on those applications if various cases of that application are keeping running on various virtual machines.

3) Pre-emptive Invigorating: For this situation work which is executed is acquired, its state is then spared and afterward it is moved to another framework.

4) Load Neutralizer: Whenever the heap (% usage) of CPU and memory surpasses a specific point of confinement. For instance, if 75% usage of CPU is considered as farthest point. The heap from that CPU is exchanged to other CPU which does not surpass its utmost.

4. Comparison Table

Reference	Energy Consumption	Execution Time	Accuracy	MTBF	MTTR	Fault tolerance degree	Remarks
[28]	**Yes**	**Yes**	**No**	**No**	**No**	**Yes**	**A low-overhead two-state checkpointing (TsCp) scheme for fault-tolerant is used and TsCp with dynamic voltage scaling (DVS) to achieve up to mark result.**
[29]	**No**	**Yes**	**no**	**No**	**Yes**	**Yes**	**It integrates fault tolerance and load balancing within a distributed system based on CORBA to make system reliable.**
[30]	**Yes**	**No**	**No**	**Yes**	**No**	**Yes**	**Adaptive checkpointing is combined with a dynamic voltage scaling scheme to achieve power reduction.**

ISSN: 0971-1260
Vol-22-Issue-16-August-2019

[31]	**Yes**	**Yes**	**No**	**Yes**	**No**	**Yes**	**It uses hash-based incremental checkpointing to lower overheads and increased efficiency.**
[32]	**Yes**	**No**	**Yes**	**No**	**No**	**Yes**	**It uses proactive process- level migration approach; however it does not rely on a spare node or redundant computing prior to prediction of a failure.**
[33]	**Yes**	**Yes**	**No**	**No**	**Yes**	**Yes**	**Checkpointing is used to minimize energy consumption to achieve better result.**
[35]	**No**	**Yes**	**No**	**Yes**	**Yes**	**No**	**For reliable fault tolerance in a system, optimal number of checkpoints is applied and save the system from complete re-execution and energy minimization can also be achieved.**
[36]	**Yes**	**Yes**	**No**	**Yes**	**No**	**Yes**	**Load Balancing with VM migration is achieved to achieve optimal result.**

Conclusion and future scope

Resource utilization becomes a hot concern among researchers. This paper is an extension of mechanism used to ensure reduced resource utilization to conserve cost and enhances efficiency in terms of load balancing degree. The load balancing degree is considerably improved when throttle load balancing strategy is implemented at the broke level. Broker analysis the availability of VM and in case of unavailability, broker puts the cloudlet into the queue and cloudlet is not lost. As and when resource becomes available, resources are allotted to the job. Result in terms of through, load balancing degree and throughput shows significant improvement.

4. References

[1] J. Han, W. Susilo, Y. Mu, and J. Yan, "Attribute-Based Data Transfer with Filtering Scheme in Cloud Computing," *IEEE Access*, vol. 57, no. 4, 2014.

[2] X. Jin, F. Zhang, L. Wang, S. Hu, B. Zhou, Z. Liu, and P. Device, "Joint Optimization of Operational Cost and Performance Interference in Cloud Data Centers," *IEEE Access*, vol. 7161, no. c, 2015.

[3] G. A. Kaur and S. Sharma, "Fault Tolerance in Cloud Computing : A Review," *ACM*, vol. 8491, pp. 94–98, 2017.

[4] T. Gouasmi, W. Louati, and A. H. Kacem, "Cost-Efficient Distributed MapReduce Job Scheduling across Cloud Federation," in *2017 IEEE International Conference on Services Computing (SCC)*, 2017, pp. 289–296.

ISSN: 0971-1260
Vol-22-Issue-16-August-2019

[5] A. Ibrahim, H. Jin, A. A. Yassin, D. Zou, and P. Xu, "Towards Efficient Yet Privacy-Preserving Approximate Search in Cloud Computing," *IEEE Access*, vol. 57, no. 2, 2014.

[6] N. Wahidah, B. Ab, K. Jenni, S. Mandala, and E. Supriyanto, "Review On Cloud Computing Application In P2P Video Streaming," *Procedia - Procedia Comput. Sci.*, vol. 50, pp. 185–190, 2015.

[7] A. Sharma and S. Sharma, "Credit Based Scheduling Using Deadline in Cloud Computing Environment," *ACM Comput. Surv.*, pp. 1588–1594, 2016.

[8] O. K. Hussain and F. K. Hussain, "A User-Based Early Warning Service Management Framework in Cloud Computing," *IEEE Access*, 2014.

[9] C. Cheng and H. Liao, "A Malicious-Resilient Protocol for Consistent Scheduling Problem in the Cloud Computing Environment," *IEEE Access*, vol. 58, no. 2, 2015.

[10] O. Rebollo, D. Mellado, and E. Fernandez-medina, "ISGcloud : a Security Governance Framework for Cloud Computing," *IEEE Access*, 2014.

[11] R. R. Expósito, G. L. Taboada, S. Ramos, J. Touriño, and R. Doallo, "Performance Evaluation of Data-Intensive Computing Applications on a Public IaaS Cloud," *IEEE Cloud Comput.*, 2014.

[12] A. V. Dastjerdi and R. Buyya, "An Autonomous Time-Dependent SLA Negotiation Strategy for Cloud Computing," *IEEE Access*, 2015.

[13] C. Lin, "Scheduling for Time-Constrained Big-File Transfer Over Multiple Paths in Cloud Computing," *IEEE Commun. Lett.*, vol. 2, no. 1, pp. 25–40, 2018.

[14] F. Permissions, "A Multidimension Taxonomy of Insider Threats in Cloud Computing," *IEEE Access*, 2016.

[15] R. Buyya, R. Ranjan, and R. N. Calheiros, "InterCloud : Utility-Oriented Federation of Cloud Computing Environments for Scaling of Application Services," *IEEE Access*, pp. 1–20, 2017.

[16] R. Buyya, C. S. Yeo, and S. Venugopal, "Market-oriented cloud computing: Vision, hype, and reality for delivering IT services as computing utilities," *Proc. - 10th IEEE Int. Conf. High Perform. Comput. Commun. HPCC 2008*, pp. 5–13, 2008.

[17] X. Cui, B. Mills, T. Znati, and R. Melhem, "Shadows on the cloud: An energy-aware, profit maximizing resilience framework for cloud computing," *CLOSER 2014 - Proc. 4th Int. Conf. Cloud Comput. Serv. Sci.*, pp. 15–26, 2014.

[18] Y. Chang and J. Chen, "A Hybrid Cloud for Effective Retrieval from Public Cloud Services," *IEEE*, pp. 61–69.

[19] L. Shakkeera and L. Tamilselvan, "Energy-Aware Application Scheduling and Consolidation in Mobile Cloud Computing with Load Balancing," *IEEE Access*, no. Mcc, 2016.

[20] H. Shen, S. Member, and L. Chen, "Management in Datacenters Using Finite-Markov Decision Process," *IEEE*, vol. 25, no. 6, pp. 3836–3849, 2017.

[21] T. Bogodorova, S. Member, L. Vanfretti, and S. Member, "Identifying Uncertainty Distributions and Confidence Regions of Power Plant Parameters," *Springer*, pp. 19213–19224, 2017.

[22] S. S. Lakshmi, "Fault Tolerance in Cloud Computing," *IEEE*, vol. 04, no. 01, pp. 1285–1288, 2013.

[23] R. Kaur, "A Hybrid Approach for Virtual Machine Migration in Cloud Computing Environment," *IEEE*, no. 9, pp. 30–35, 2017.

[24] X. Zheng, S. Member, P. Martin, K. Brohman, and S. Member, "CLOUDQUAL : A Quality Model for Cloud Services," *ACM*, vol. 10, no. 2, pp. 1527–1536, 2014.

ISSN: 0971-1260
Vol-22-Issue-16-August-2019

[25] J. Aguilar, P. Valdiviezo-dı, and G. Riofrio, "ORIGINAL ARTICLE A general framework for intelligent recommender systems," *ACM*, pp. 147–160, 2017.

[26] T. G. Rodrigues, S. Member, and K. Suto, "Hybrid Method for Minimizing Service Delay in Edge Cloud Computing through VM Migration and Transmission Power Control," *IEEE Access*, vol. 9340, no. 10, pp. 1–11, 2016.

[27] A. S. Chaleshtari, "Resource Tardiness Weighted Cost Minimization," *ACM Comput. Surv.*, vol. 2017, 2017.

[28] N. Nickbakhsh, M. Reza, and S. Aghaei, "Resource discovery algorithm based on hierarchical model and Conscious search in Grid computing system," *IEEE*, vol. 2017, no. 1, pp. 24–43, 2017.

[29] A. Beloglazov and R. Buyya, "Managing Overloaded Hosts for Dynamic Consolidation of Virtual Machines in Cloud Data Centers Under Quality of Service Constraints," *IEEE Commun. Lett.*, vol. X, no. X, pp. 1–14, 2012.

[30] M. Kaur, K. Kaur, and L. Kapoor, "An Efficient Resource Provisioning Technique in Inter-Cloud using Peer-to-Peer Approach," *ACM*, 2017.

[31] J. Xu, "Cost-aware Resource Management for Federated Clouds Using Resource Sharing Contracts," *ACM*, 2017.

[32] T. Chen, S. Member, A. G. Marques, S. Member, and G. B. Giannakis, "DGLB : Distributed Stochastic Geographical Load Balancing over Cloud Networks," *IEEE*, vol. 9219, no. c, 2016.

[33] Y. Chu, N. Huang, S. Member, and S. Lin, "Quality of Service Provision in Cloud-based Storage System for Multimedia Delivery," *IEEE*, vol. 8, no. 1, pp. 292–303, 2014.

[34] G. Fan, H. Yu, and L. Chen, "A Formal Aspect-oriented Method for Modeling and Analyzing Adaptive Resource Scheduling in Cloud Computing," *IEEE*, vol. 11, no. 4, pp. 1–14, 2016.

[35] C. Feng, H. Xu, and B. Li, "An Alternating Direction Method Approach to Cloud Traffic Management," *IEEE Access*, vol. 9219, no. 1, pp. 1–13, 2017.

[36] V. Dumitriu, S. Member, L. Kirischian, and V. Kirischian, "Run-Time Recovery Mechanism for Transient and Permanent Hardware Faults Based on Distributed , Self-organized Dynamic Partially Reconfigurable Systems," *IEEE Access*, vol. 9340, no. c, pp. 1–14, 2015.