IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-5, May 2019 E-ISSN: 2347-2693

Reliability Optimization Using Distance Check pointing Through Dynamic Request and Requirement Aware Mechanism

PreetKawal Kaur[1*], Kamaljit Kaur[2]

[1]Department of Computer Engineering and Technology,Guru Nanak Dev University, Amritsar, Punjab, India
[2]Department of Computer Engineering and Technology,Guru Nanak Dev University, Amritsar, Punjab, India

[*]Corresponding Author: preetkawal123@gmail.com , Tel.: 7837202049

DOI: https://doi.org/10.26438/ijcse/v7i5.13511359 | Available online at: www.ijcseonline.org

Accepted: 21/May/2019, Published: 31/May/2019

Abstract- In cloud computing the datacenters are utilized to coordinates the distinct tasks where tasks may require more resources and source of these cloudlets could be from thousands of users. These datacenters aims to deliver reliable services. But the equal reliability to all the users at the same time according to their requirements may be a difficult task and may vary. So in this paper we purpose a technique that considers optimized elastic reliability in cloud computing using distance based server selection policy. In our scheme reliability enhancement through distance dependent check pointing with resource maximization mechanism is distance aware checkpoint. Resource maximization mechanism uses division policy which divides the jobs by looking at the capacity of virtual machine on which load is to be dispersed. This operation can efficiently solve the problem of reliability. it improves the resource usage in the datacenters and also gives optimized reliability to the user.

Keywords—Reliability,Checkpointing,datacenters,cloudcomputing

I.INTRODUCTION

The cloud computing is an enhancement of distributed computing models in which the services are provided on demand and has feature of pay –per –use. [3]this type of features of cloud computing facilitate an organization to avail quick resource according to its requirement and with less service cost of cloud services. The main idea behind cloud services is to provide appealing business opportunities for developing organizations. In other words, developing organization not required to purchase the resources rather resources can be acquired on pay per use basis hence saving large amount of cost overhead on purchase of these equipments. The cloud services ought to be solid to hold the Quality of Service (QoS) necessities.[4]Different virtual machines is constituted by the cloud computing at the practical level(VMs) running crosswise over physical machines that might be put in various geological areas to shape a virtual cloud group which contains multi-occupants. There may be one or more such virtual cloud group can be framed progressively directly associated in achieving a solitary mission. This multi-tenure model makes the cloud computing
more inclined to different security ruptures than other appropriated computing model.
[5]among those, the most difficult perspective is fault which decrease the performance by increasing the delay and decreasing the performance enhancement of task execution. The attack cannot detect easily which can be in forms of logical errors. Also it can spread starting with one VM then onto the next rapidly. However different faults can be distinguished yet load oriented faults stays tricky and causes genuine harms, since the framework continues working even with the instigated faults. Also Byzantine fault can be utilized as the method of spread cloud failure in VM, server, system, application and finish. However Cloud services are on-request and paid for each unit time. In this way the affirmation of reliability offered is high however because of failures, like software failure, hardware failure, and network and so on it turns out to be difficult to keep up the required level of execution in satisfying the QoS. This could cause the clients discontent towards the CSP.

[6]the various fault handling mechanism are utilized that are categorized as detection of fault, its prevention, fault prediction and tolerance.

[7]the fault detection is very crucial task and it initializes the fault handling technique. However because of the hesitant idea of the load oriented faults it ends up repetitive to distinguish it in the underlying state. In this manner in the past work an elastic mechanism for handling fault has been distance aware checkpoint which is turned out to be fit for distinguishing the faults successfully. As other techniques that deal to manage the faults at different stages either in proactive or responsive way is given. Among those the prevention of fault is imperative since it tries to keep the faults with legitimate cautious component. As often the various faults are created that would beat the current

protection framework hence fault tolerance ends up essential in the cloud frameworks as a possibility for handling fault. Aside from different strategies Fault tolerance is an endeavor to guarantee benefit congruity regardless of the fault events.

[8]the practical implication for cloud reliability is fault tolerance that upgrades it. There are different fault tolerance components, for example, checkpointing, replication, undertaking Migration, Self-Healing, Safety-sack checks, Retry, Task Resubmission, Reconfiguration, Masking and so on. Among the available fault tolerance mechanisms, the Checkpointing is a generally mostly used fault tolerance mechanism. The checkpointing procedures as often as possible spares the condition of each VM as backup image and are promptly deployable if any VM or group of VMs comes up to fail . However to keep up aggregate checkpointing for a fault inclined circumstance where failure is normal, requires exceptional checkpointing for each limited time interims. The small time interim turns out to be more space and tedious the checkpointing can be. Besides dividable jobs, for example, if there should be an occurrence of enormous data investigation if any one VM creates wrong yield it can be passed on to others and in this way the whole yield can be debased.

[9]the checkpointing in such case is repetitive and can cause overheads in performance. However there is just set number of research work is present for improving checkpointing strategy. Checkpointing still has issue of triggering at uniform or discrete time intervals... Consequently checkpointing is done as a comprehensive task which implies if a failure is normal then the number of checkpoints are expanded for each VMs constantly. This means number of checkpoints at every distinct interval can be increased depending upon the level of faulty environment. Higher the faulty environment more will be the checkpoints.

[10]Requirements and services comparing to the organization may vary according to client necessities and the sort's occupation. Faults and failures may arise due to topology break, physical deterioration of hardware components, etc. could lead to loss of data and progress made at VM. To handle such issues fault tolerant strategies like checkpointing can be used. Checkpointing in heavy faulty environment must be at uniform time intervals that lead to slow progress of job processing. In other words in case of heavy faults, checkpoints will be more significant that lead to slow execution of jobs or tasks. The jobs can be further sub isolated as undertakings which are made appropriate for VM level handling . The above analysis of literature suggests effectiveness of checkpoints but it may not always produce best results.. However the activity is divided into different energy preservation mechanisms to suit the VMs load allocation.

Organization of this paper is given as under: Section II describes the literature survey, section III presents the distance aware checkpoint system, section IV gives the result and performance analysis, section V gives conclusion and future scope and last section gives the references.

II. LITERATURE SURVEY

Classification of different mechanisms for hardware checkpointing and present formulas for estimating the hardware overhead is distance aware checkpoint. Moreover, this literature reveals a tool that takes over the burden of modifying hardware modules for checkpointing. Task sorting is missing in this literature.

[12]A. Guermouche in distance aware checkpoint a mechanism to store the checkpoint image on the server in uncoordinated manner. In other words, server selection is on the basis of storage space. Larger storage space servers can host more checkpointing images and recovery initiates in case deteriorated VM is found out. Performance however can be further improved in case server selection considers resource availability other than storage.

D. Jung, in distance aware checkpoint a mechanism to minimize the checkpoint trials required to recover the progress made through the virtual machine. Cloudlets progress is monitored through the broker. In case deterioration is detected, progress made by VM becomes transferred to checkpoint server. Uncoordinated checkpointing approach ensures less trials and fast recovery.

[13]The research work in aims to minimize the expectation of the total energy consumption, while enforcing a deadline on the execution time, that should be met either in expectation (soft deadline), or in the worst case (hard deadline). Deadline jobs can be tackled by allocating them to VM having minimum possible waiting time associated with them.

R. Rajachandrasekar in distance aware checkpoint a power aware checkpointing mechanism. server selection in earlier work does not considered energy required to transfer the checkpoint image to the server. To tackle the issue, distance between the virtual machines is considered before migrating progress to the checkpoint server. Result in terms of energy and makespan shows improvement. Job partitioning however not considered in this approach.

[14]M. V Santiago in distance aware checkpoint and attack the dependability problem in a distributed manner, a fault tolerant solution based on communication-induced checkpointing (CiC) for interactive BPEL processes(Business Process execution language).

The research work in an optimal checkpoint method is distance aware checkpoint with edge switch failure-aware fault tolerance. The edge switch failure-aware checkpoint method includes two algorithms. The first algorithm employs the data center topology and communication characteristic for checkpoint image storage server selection. The second algorithm employs the checkpoint image storage characteristic as well as the data center topology to select the recovery server. Selection process however can be further improved by cloudlet and VM sorting process.

[15]K. N. Devi and A. Tamilarasi in describes work process merchants of static checkpoint Grid Scheduling Systems are resistance instrument which causes wasteful timetables of use dispersed resources and it likewise compounds the use of different resources including system transmission capacity and computational cycles. In such models, crucial obligation, for example, resource revelation is assigned to the unified server machines, in this way they are related with surely understood impediments in regards to single purpose of disappointment, versatility and system blockage at joins that are prompting the server. With a specific end goal to conquer these issues, we execute another approach for decentralized agreeable work process scheduling in a progressively dispersed resource sharing condition of Grids. At the point when there is a disappointment of job, it will move to another computational hub and resume from the last put away checkpoint. A Glow worm Swarm Optimization (GSO) for job scheduling is utilized to address the issue of heterogeneity in adaptation to internal failure of computational grid however Weighted GSO that beats the position refresh flaws of general GSO in a more effective way appeared amid correlation investigation. This framework underpins four sorts of adaptation to internal failure instruments, including the job movement, job retry, registration and the job replication systems additionally considering hazard nature of Grid processing condition. The hazard connection amongst jobs and hubs are characterized by the security request and then confide in level. Our assessment based recreation comes about demonstrate that our calculation has shorter make span and more effective. We additionally investigate the productivity of the distance aware checkpoint approach against a brought together organized work process scheduling system and demonstrate that our approach is more effective than the incorporated strategy as for accomplishing exceedingly planned calendars.

[16]P. Graubner in distance aware checkpoint an approach for enhancing the vitality effectiveness of framework as-a-benefit clouds is introduced. The approach depends on performing live movements of virtual machines to spare vitality. As opposed to related work, the vitality expenses of live movements including their pre-and post-handling stages are considered, and the approach has been executed in the Eucalyptus open-source cloud registering framework by proficiently joining a multi-layered document framework and disseminated replication square gadgets. To assess the distance aware checkpoint approach, a few short-and long haul tests in light of virtual machine workloads created with regular working framework benchmarks, web-server copies and in addition diverse Map Reduce applications have been led. The outcomes demonstrate that vitality investment funds of up to 16 percent can be accomplished in a beneficial Eucalyptus condition.

The grid figuring in empowers the clients to share the heterogeneous resources which are dispersed topographically. The fundamental preferred standpoint of grid is to use the unused resources in other words a compelling use of resources. At the point when considering about the resource usage the emphasis is on the distinctive methodologies and techniques which are actualized for successful scheduling .This paper shows a broad overview about the distinctive systems accessible in scheduling the resource.

S. Di et al. in goes for advance adaptation to non-critical failure methods in light of a checkpointing/restart component, with regards to cloud processing. Our commitment is three-crease. We infer a new recipe to figure the ideal number of checkpoints for cloud jobs with differed dispersions of disappointment occasions. Our investigation isn't just non specific with no presumption on disappointment likelihood circulation, yet in addition appealingly easy to apply by and by. We outline a versatile calculation to upgrade the effect of checkpointing with respect to different costs like checkpointing/restart overhead. We assess our advanced arrangement in a genuine bunch condition with several virtual machines and Berkeley Lab Checkpoint/Restart instrument. Errand disappointment occasions are imitated through a generation follow created on a vast scale Google server farm. Analyses affirm that our answer is genuinely appropriate for Google frameworks. Our improved recipe beats Young's equation by 3-10 percent, diminishing divider clock lengths by 50-100 seconds for every job by and large.

Checkpointing with rollback recuperation in distance aware checkpoint a discrete voltage frequency levels for assignment of checkpointing image to the server. Checkpointing image storage and recovery considering power specified through DVFS levels enhances performance. Fast execution of cloudlets accomplishes with the distance aware checkpoint system. More cloudlets can be executed as the powered VM was considered for allocation of tasks. Task partitioning however is not considered that can further enhance the performance in terms of reliability.

Literature work suggests that least amount of work is done towards the partitioning of task depending upon the virtual machine availability. Performance in terms of reliability and execution time can be improved considering the job partitioning considered in the distance aware checkpoint system.

This area portrays the writing of static checkpoint instrument considered for better unwavering quality in cloud based condition.

Table1:Comparision table

REFERENCE	FAULT TOLERANCE TECHNIQUE	PARAMETERS	What is achieved	Tradeoffs
[17] Using Proactive Fault - Tolerance Approach to Enhance Cloud Service Reliability	Virtual Cluster Allocation mechanism to reduce network resource allocation and CPU temperature is monitored to detect deteriorating machine	CPU Temperature Network Resource Utilization	1. Deterioration is avoided by reducing network resource utilization 2. CPU temperature is monitored for reallocation	Monitoring temperature increases overhead and cost associated with suggested technique
[18] Energy-efficient data replication in cloud computing datacenters	Proactive approach is used to replicate files based on popularity and reactive approach is used to track popularity changes at smaller interval of times	Data locality Turnaround time Mean slowdown	1. Replication enhances fault tolerance 2. Dynamic files priority decreases chances of starvation	Reactive and proactive approach in combination enhances overhead
[19] Quality-of-service in cloud computing : modeling techniques and their applications,	Quality of Service modelling for fault tolerance in cloud system	Capacity Allocation Load Balancing Admission Control	1. Quality of service modelling described efficient mechanisms to provide fault tolerance through reducing overhead	No new techniques are suggested rather static checkpoint techniques are surveyed
[20] ACCFLA : Access Control in Cloud Federation using Learning Automata	Access Control using finite automata	Risk Trust Access	1. Access Control mechanism ensure stoppage of unauthorised access 2. Finite automata is used to design fault tolerant mechanism ensuring reduced overhead	Both deterministic and non deterministic mechanisms are used hence cost of fault tolerance is significantly increases
[21] Accelerating incremental checkpointing for extreme-scale computing	Incremental check pointing approach for fault tolerance	Computation Overhead Number of Checkpointing	1. Checkpointing technique which is used is dynamic since size of checkpointing can be scaled depending upon requirements	Cost and overhead in terms of energy efficiency is not considered
[22] Two-State Checkpointing for Energy-Efficient Fault Tolerance in Hard Real-Time Systems,	Checkpointing for fault tolerance and energy efficiency	Power Consumption Energy Consumed	1. Checkpointing mechanism is used to support energy efficiency. 2. Time critical applications can be solved efficiently through this approach	Large and complex problems like workflow problems are not examined trough this approach
[23] Checkpointing Towards Dependable Business Processes,	Design and implementation of infrastructure for checkpointing/restart fault tolerance	Latency Overhead	1. Infrastructure related issues are resolved using checkpointing 2. Latency is reduced and overhead is reduced	Complex applications such as workflow applications are not handled using this approach
[24] Fusion of Hyperspectral and LiDAR Data for Classification of Cloud-Shadow Mixed Remote,	Shadow Computing mechanism to enhance fault tolerance	Energy Efficiency Fault tolerance	1. Shadows are maintained so that in case of failure copy of progress can be stored within the shadow 2. Energy efficiency is achieved	Overhead in terms of cost is enhanced
[25] Fixed-Priority Allocation and Scheduling for Energy-Efficient Fault Tolerance in Hard	Allocation strategy based on fixed priority in hard real time environment	Fault monitoring Energy Consumption	1. High priority applications are stored within the cloud at first place hence enhancing reliability. 2. Tim bound applications are efficiently handled	Cost and overhead is high and can be further reduced
[26] Energy Ef fi cient Cloud Computing	Voltage monitoring for enhancing reliability and reducing energy efficiency	Reliability Energy Efficiency	1. Voltage is varied to reduce energy consumption during allocation of task. 2. Reliability is enhanced along with energy efficiency through the application of examined literature	Hardware reliability is more focused as compared to software reliability
[9]CEFIoT: A Fault-Tolerant IoT Architecture for Edge and Cloud	CDFIOT strategy	Bandwidth ,delay	it overcomes the physical node failure and long-distance network problems by providing replication-based local fault	It does not work in clustered environment so it need large number of both edge and cloud clusters.

			tolerance on edge devices	
[27]Fault Tolerance in Cloud Using Reactive and Proactive Techniques	Copy-on-Write Presave in cache algorithm	Number of faults Runtime or execution time	Low overhead and better fault occurrence rate	This approach reacts after the fault has occurred. Though the algorithm tolerates the fault, there is always performance degradation in the system due to the occurrences of the fault.
[28]Using imbalance characteristic for fault-tolerant workflow scheduling in Cloud systems	ICFWS algorithm for fault free workflow	Runtime Average data size tackled	Achieving fault free workflow in cloud	Besides fault, the performance fluctuation of VMs also plays a negative effect to the execution of workflow
[29]Using Proactive Fault-Tolerance Approach to Enhance Cloud Service Reliability	PSO-Based PM Selection	Transmission overhead Execution time	Reliability Execution time	Fault tolerance mechanism with full coordinated checkpointing mechanism can further be employed
[30]Adaptive Framework for Reliable Cloud Computing Environment	Replication and checkpointing based methodology	Overhead Monetary cost	Throughput increasing and decreasing overhead	Extensive space utilization increase the overall cost of implementing this literature

III. DISTANCE AWARE CHECKPOINT SYSTEM

Cloud is exposed to infinite users with distinct requirements. The job requires resource and if resource is not available then job must wait or job will fail. Distance aware checkpoint system enhances the reliability of job execution by ensuring every job although requiring more resources can be allotted to VMs with fewer resources. To accomplish this, distance aware checkpoint system uses the mechanism in which jobs or tasks are partitioned depending upon the resource availability. Peer to Peer checkpointing approach ensures regular backup of progress made by the virtual machine at backup machine. To take the backup field intervals are made. In other words after that fixed interval of time backup is initiated. Uniform environment indicates that the interval of back up is fixed. E.g. if initial back up time is t and α is the uniform time interval then t+α will be the next time interval at which back is made. Backup machine selection is on the basis of optimal resources a VM has. If the resources such as RAM, processing elements, OS, MIPS etc. associated with the virtual machines are maximum then VM is considered as optimal. Failure detection phase includes broker monitoring. Broker continues investigate the virtual machine. As the current VM deteriorates from its original metric requirements, deterioration is detected and recovery from backed up image imitated. Detailed methodology is given in this section as

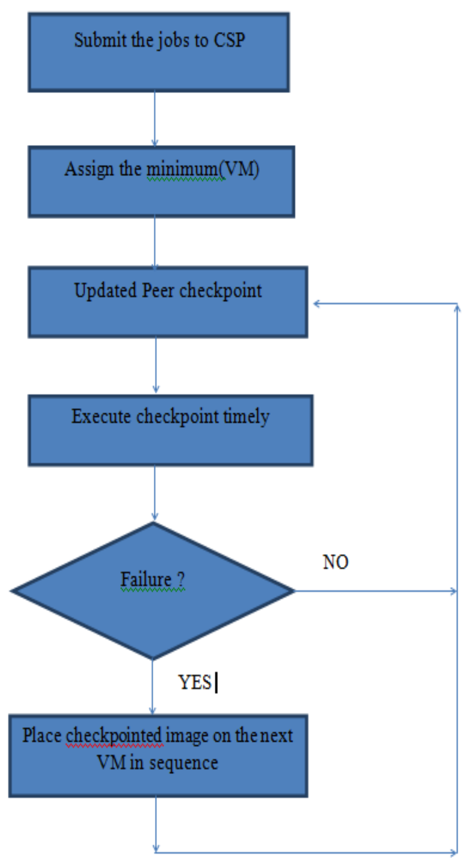

Figure 1: Distance dependent Checkpointing with Task division approach for reliability enhancement

III. RESEARCH METHODOLOGY

A. PROBLEM DEFINITION

The recuperation picture stockpiling instrument can be suited with most limited way determination methodology to store the checkpoint picture at closest server accessible. As the closest server is chosen for capacity time required for recuperation is extensively decreased. Accordingly vitality utilization additionally decreases.

In the second calculation, recuperation server determination process does not think about the separation metric. Again most brief way calculation can be utilized to diminish the time required to create reinforcement. Henceforth pleasing separation inside these two calculations can diminish execution time and vitality utilization. Since separation is straightforwardly corresponding to vitality utilization.

B. OBJECTIVE

There are certain objectives which are associated with the fault tolerance in cloud computing. The distance aware checkpoint system with distance aware resource allocation is given as under

1. To Reduce vitality Consumption to produce ideal outcomes.

2. To provide management of excess information so it very well may be diminished.

3. To improve Access time ought to be diminished.

4. To improve the disappointment time by utilizing separation based checkpointing procedure.

C. DISTANCE AWARE CHECKPOINT METHODOLOGY

Algorithm 1:

The distance aware checkpoint methodology is based on the selection of machines on the basis of distance. The machines that are selected at first place are on the basis of maximum resources they possess. The second algorithm demonstrates the selection of virtual machine for migration.

1. Form the VM list on the basis of maximum resources possessed by physical machine
2. Distance based vm selection procedure is opted in the distance aware checkpoint system
3. Applying B_Sort procedure for sorting VMs according to resources possessed by VMs
    a) If(VM[i]>VM[i+1])
    b) Temp=VM[i]
    c) VM[i]=VM[i+1]
    d) VM[i+1]=Temp
4. Repeat the above steps for all the virtual machines under considerations.

Algorithm 2

The second algorithm is based on the minimum distance possessed by virtual machines and is used in case current virtual machines are failed.

1. Assign distance with each virtual machine
2. Check for minimum distance VM and add it to Optimal_VM list
3. In case current VM is failed then
4. Select the virtual machine from the optimal_VM list.
5. Output the result in terms of energy efficiency, fault tolerance rate.

Fault Injection and Recovery- In this paper reaction shortcoming will be infuse so that amid transmission of the issues that are happening will be effectively settled. At that point by applying the above calculation we will recoup the shortcomings that are happening in the framework by choosing the best server that can finish the occupations quickly that was adhered because of deficiency happening in the framework. The flowchart of the separation thought in distance aware checkpoint framework is as under

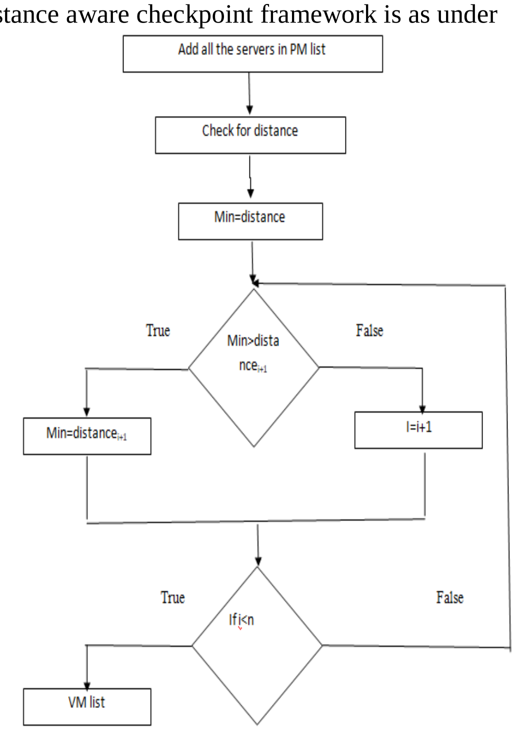

Figure 2: Flow of dynamic checkpointing approach

IV. RESULT AND PERFORMANCE ANALYSIS

The result section describes the mechanisms of checking the performance of the distance aware checkpoint system. There are parameters that are considered during the performance evaluation. The execution time is critical in the analysis process.

The finish time is estimate using the equation 1

$$Execution_{time} = finish_{time} - start_{time}$$

Equation 1: Execution time or finish time

The finish time is the difference between the submitted tasks and time during which the tasks finished execution. Table 1: gives the result in terms of finish time

Table 2: Finish time comparison

Tasks	Delay Time in milli-second(Static checkpoint)		Delay Time in milli-second(Distance aware checkpoint)	
	Before fault	After fault	Before fault	After fault
58	1989	1739	1236	1158
145	1999	1867	1336	1239
127	2079	1987	1437	1347
238	1858	1658	1268	1098
276	1868	1739	1538	1437

The plot for the finish time corresponding to cloudlets submitted to virtual machines within the datacenters is in figure 2

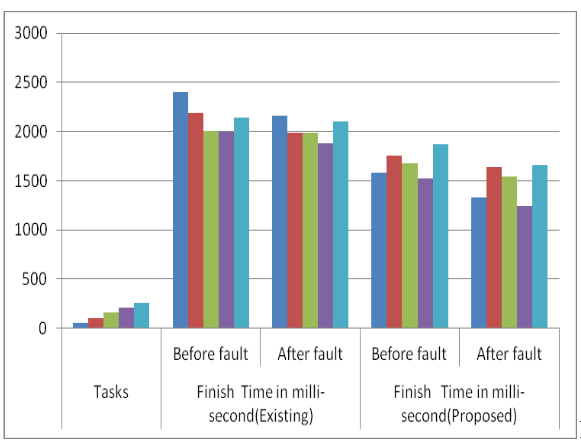

Figure 3: Plot in terms of execution time

Down Time- It is the total time during which a machine is out of action or not available for usage. The downtime is estimated by the use of total cloudlets submitted to the total execution time.

Table 3: In terms Of Down Time

Tasks	Finish Time in milli-second(Static checkpoint)		Finish Time in milli-second(Distance aware checkpoint)	
	Before fault	After fault	Before fault	After fault
59	2398	2159	1585	1328
108	2188	1989	1758	1639
158	2007	1987	1678	1547
210	1998	1879	1527	1239
259	2137	2099	1869	1660

This is given in equation 2.
Equation 2: Downtime of virtual machines

$$downtime = \frac{total_{cloulets}}{total_{execution}}$$

Downtime corresponding to the static and dynamic checkpointing approach. Is shown in the figure 3.

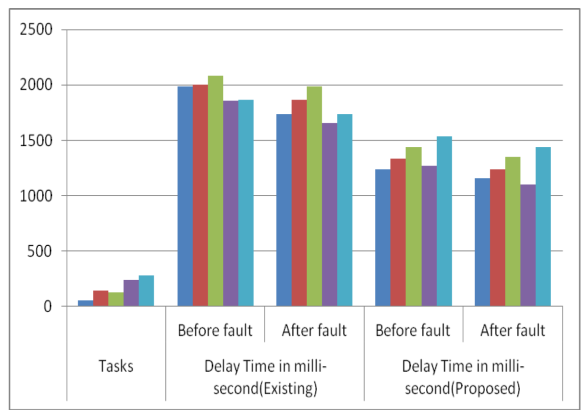

Figure 4: plot down time of distance aware checkpoint and static checkpoint paper

Disk Usage- It is the total space used by the process in the computer. The disk usage of distance aware checkpoint and static checkpoint paper will be considered in terms of Mega bytes. Disk usage is estimated by subtracting the total head movement to the start position of the head.

Table 4: in terms of disk usage

Tasks	Disk access time in MB(Static checkpoint)		Disk access time in MB(Distance aware checkpoint)	
	Before fault	After fault	Before fault	After fault
58	2657	2357	1566	1348
167	2348	2138	1857	1677
175	2268	2099	1878	1769
257	2679	2457	1989	1879
295	2179	1987	1867	1567

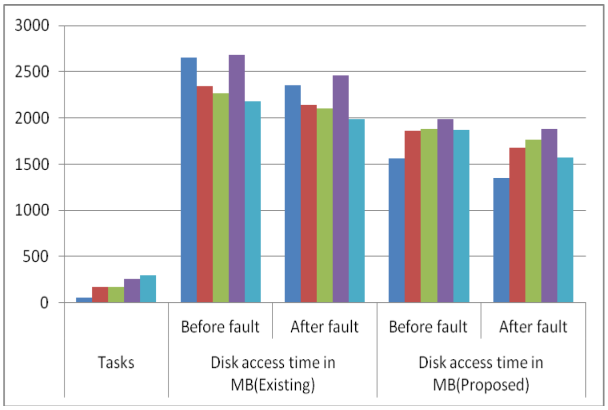

Figure 5: plot in terms of disk usage

V. CONCLUSION

In this paper we are looking at the issue of cloud administration dependability. In our distance aware checkpoint methodology checkpointing strategy is utilized in distributed computing to recuperate the information which can be lost because of framework crash or a lot more reasons. To defeat this issue we utilize most brief way remove calculation with the goal that task amid information recuperation can be relocated to the base separation servers. So it very well may be effectively accessible or open to acquire ideal outcomes as far as circle utilization, picture measure, absolute execution time, normal lost time and so forth.

REFERENCES

[1] S. Ranga, "A Survey for Secure Live Migration of Virtual Machines in Cloud Computing Platform," pp. 110–115.
[2] D. Jain and V. Singh, "Feature selection and classification systems for chronic disease prediction: A review," *Egypt. Informatics J.*, 2018.
[3] C. Feng, H. Xu, and B. Li, "An Alternating Direction Method Approach to Cloud Traffic Management," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 8, pp. 2145–2158, 2017.
[4] J. Wang, W. Bao, X. Zhu, L. T. Yang, and Y. Xiang, "FESTAL: Fault-Tolerant Elastic Scheduling Algorithm for Real-Time Tasks in Virtualized Clouds," *IEEE Trans. Comput.*, vol. 64, no. 9, pp. 2545–2558, 2015.
[5] M. R. Chinnaiah and N. Niranjan, "Fault tolerant software systems using software configurations for cloud computing," *J. Cloud Comput.*, vol. 7, no. 1, 2018.
[6] A. Zhou, S. Wang, C. H. Hsu, M. H. Kim, and K. seng Wong, "Virtual machine placement with (m, n)-fault tolerance in cloud data center," *Cluster Comput.*, pp. 1–13, 2017.
[7] Y. Wang, Q. He, D. Ye, and Y. Yang, "Formulating Criticality-Based Cost-Effective Fault Tolerance Strategies for Multi-Tenant Service-Based Systems," *IEEE Trans. Softw. Eng.*, vol. 44, no. 3, pp. 291–307, 2018.
[8] G. Li, J. Wu, J. Li, K. Wang, and T. Ye, "Service Popularity-based Smart Resources Partitioning for Fog Computing-enabled Industrial Internet of Things," *IEEE Trans. Ind. Informatics*, vol. PP, no. c, p. 1, 2018.
[9] R. K. Naha, S. Garg, D. Georgakopoulos, P. P. Jayaraman, L. Gao, Y. Xiang, and R. Ranjan, "Fog computing: Survey of trends, architectures, requirements, and research directions," *IEEE Access*, vol. 6, no. c, pp. 47980–48009, 2018.
[10] A. Zhou, S. Wang, B. Cheng, Z. Zheng, F. Yang, R. N. Chang, M. R. Lyu, and R. Buyya, "Cloud service reliability enhancement via virtual machine placement optimization," *IEEE Trans. Serv. Comput.*, vol. 10, no. 6, pp. 902–913, 2017.
[11] S. M. Abdulhamid, M. S. Abd Latiff, S. H. H. Madni, and M. Abdullahi, "Fault tolerance aware scheduling technique for cloud computing environment using dynamic clustering algorithm," *Neural Comput. Appl.*, vol. 29, no. 1, pp. 279–293, 2018.
[12] A. S. Perspective, R. Jhawar, G. S. Member, and V. Piuri, "Fault Tolerance Management in Cloud Computing :," *IEEE Syst. J.*, vol. 7, no. 2, pp. 288–297, 2013.
[13] C. A. Chen, M. Won, R. Stoleru, and G. G. Xie, "Energy-efficient fault-tolerant data storage and processing in mobile cloud," *IEEE Trans. Cloud Comput.*, vol. 3, no. 1, pp. 28–41, 2015.
[14] R. Jhawar and V. Piuri, "Chapter 9 - Fault Tolerance and Resilience in Cloud Computing Environments," *Comput. Inf. Secur. Handb. (Third Ed.*, vol. 2, pp. 165–181, 2017.
[15] L. P. Saikia and Y. L. Devi, "Fault tolererance techniques and algorithms in cloud system," *Int. J. Comput. Sci. Commun. Networks*, vol. 4, no. 1, pp. 1–8, 2014.
[16] T. J. Charity and G. C. Hua, "Resource reliability using fault tolerance in cloud computing," *Proc. 2016 2nd Int. Conf. Next Gener. Comput. Technol. NGCT 2016*, no. October, pp. 65–71, 2017.
[17] J. Liu, S. Wang, S. Member, A. Zhou, S. A. P. Kumar, S. Member, and R. Buyya, "Using Proactive Fault - Tolerance Approach to Enhance Cloud Service Reliability," *IEEE Access*, pp. 1–13, 2016.
[18] D. Boru, D. Kliazovich, F. Granelli, P. Bouvry, and A. Y. Zomaya, "Energy-efficient data replication in cloud computing datacenters," *IEEE Access*, vol. 18, no. 1, pp. 385–402, 2015.
[19] D. Ardagna, G. Casale, M. Ciavotta, J. F. Pérez, and W. Wang, "Quality-of-service in cloud computing : modeling techniques and their applications," *IEEE Access*, pp. 1–17, 2014.
[20] B. S. Taheri, "ACCFLA : Access Control in Cloud Federation using Learning Automata," vol. 107, no. 6, pp. 30–40, 2014.
[21] K. B. Ferreira, R. Riesen, P. Bridges, D. Arnold, and R. Brightwell, "Accelerating incremental checkpointing for extreme-scale computing," *Futur. Gener. Comput. Syst.*, vol. 30, no. 1, pp. 66–77, 2014.
[22] M. Salehi, M. K. Tavana, S. Rehman, S. Member, M. Shafique, and A. Ejlali, "Two-State Checkpointing for Energy-Efficient Fault Tolerance in Hard Real-Time Systems," pp. 1–12, 2016.
[23] M. V Santiago, S. E. P. Hernández, L. A. M. Rosales, and H. H. Kacem, "Checkpointing Towards Dependable Business Processes," vol. 14, no. 3, pp. 1408–1415, 2016.
[24] R. Luo, S. Member, W. Liao, S. Member, H. Zhang, and S. Member, "Fusion of Hyperspectral and LiDAR Data for Classification of Cloud-Shadow Mixed Remote," *IEEE Access*, pp. 1–14, 2017.
[25] R. M. Systems, T. Wei, P. Mishra, K. Wu, and H. Liang, "Fixed-Priority Allocation and Scheduling for Energy-Efficient Fault Tolerance in Hard," vol. 19, no. 11, pp. 1511–1526, 2008.
[26] S. Kannan and S. Rajendran, "Energy Efficient Cloud Computing," pp. 157–171.
[27] G. R. Kalanirnika and V. M. Sivagami, "Fault Tolerance in Cloud Using Reactive and Proactive Techniques," *IEEE*, vol. 3, no. 3, pp. 1159–1164, 2015.
[28] G. Yao, Y. Ding, S. Member, and K. Hao, "Using imbalance characteristic for fault - tolerant workflow scheduling in Cloud systems," vol. 9219, no. c, 2017.
[29] J. Liu, S. Wang, A. Zhou, S. Kumar, F. Yang, and R. Buyya, "Using Proactive Fault-Tolerance Approach to Enhance Cloud Service Reliability," *IEEE Trans. Cloud Comput.*, pp. 1–1, 2016.

[30] M. Amoon, C. Science, and P. O. B. R. Arabia, "Adaptive Framework for Reliable Cloud Computing Environment," *ACM*, vol. 3536, no. c, 2016.

Authors Profile

PreetKawal Kaur did Bachelor of Technology in Computer Science from Amritsar College of Engineering And Technology, Amritsar in year 2017. She is currently pursuing Masters of Technology in Computer Science from Guru Nanak Dev University, Amritsar and currently working as Research Scholar in Department of Computer Engineering and Technology. Her main area of research focuses on Fault Tolerance in Cloud Computing.

Ms. Kamaljit Kaur did Bachelor of Technology and Master of Technology in Computer Science from Guru Nanak Dev University. She is the Gold Medalist in Masters of Technology. She is currently pursuing Ph. D. and working as Assistant Professor in Computer Engineering and Technology Department. She has published more than 15 research papers in UGC Approved Journals. Her main research area focuses on Cloud Computing, Parallel Computing and Fault Tolerance.