

Improving Downtime and Migration using the application of shadow core

Anu chitra
Mtech scholar
SSGI College , Dinanagar

Sunil Dalhotra
Assistant Professor
SSGI college, Dinanagar

Abstract

As the interest for cloud computing keeps on expanding, cloud specialist organizations confront the overwhelming test to meet the arranged SLA agreements, as far as dependability and convenient .execution, while accomplishing cost and energy efficiency. This paper proposes Shadow Replication, a novel fault-tolerance mechanism for cloud computing, which flawlessly addresses fault at scale, while limiting energy utilization and lessening its effect on cost. Energy conservation is achieved by creating dynamic cores rather than static cores. Cores are created by the application of cloudlets. In other words proportionate cores are created. Core failure metrics are considered to be memory capacity, energy and power consumption. In case any of the parameter exceeded threshold value, core is supposed to be faulted and progress is maintained within shadow which is maintained 1 per host. Progress of deteriorated Core is shifted to next core within other VM. In case all the core within VM deteriorated, VM migration is performed. Results obtained through purposed system appears to better in terms of listed energy consumption, latency, cost and fault rate.

Keywords: shadow Replication; fault tolerance; Energy Conservation.

Introduction:

Cloud Computing has developed as an attraction stage for progressively various process and information escalated applications, as it considers low-passage costs, on demand asset provisioning and distribution and diminished cost of keeping up interior IT framework [1][1].Cloud computing will proceed to develop and pull in consideration from business and open market portions. Late reviews anticipate yearly development rate of 17.7 percent by 2016, making cloud computing the quickest developing portion in the product business [2][2].

In its essential shape, a cloud computing framework is a vast group of interconnected back-end servers facilitated in a datacenter and provisioned to convey on-request, "pay-as-you-go" administrations and computing assets to clients through a front-end interface [3][3]. As the interest for cloud computing quickens, cloud specialist organizations (CSPs) will be confronted with the need to extend their basic framework to guarantee the normal levels of execution, unwavering quality and cost-viability, bringing about a multifold increment in the quantity of computing, stockpiling and correspondence segments in their datacenters.

The immediate ramifications of vast datacenters is expanded administration many-sided quality of the computing framework, abnormal amounts of energy utilization and inclination to disappointment. The advantages of green cloud computing are clear. As server farms are quick turning into a noteworthy wellspring of worldwide energy utilization, the potential investment funds identified with energy utilize, CO₂ discharges and e-waste are certain. Accomplishing these reserve funds, be that as it may, calls for new calculation models intended to diminish energy and power utilization and advance ecologically neighborly cloud computing execution situations.

Another test for cloud computing, at scale, originates from its affinity to disappointment. While the probability of a server disappointment is little, the sheer number of computing, stockpiling and correspondence parts that can bomb, be that as it may, is overwhelming. At such an extensive scale, disappointment turns into the standard as opposed to a special case [4][4].

As the quantity of clients designating their computing assignments to CSPs expands, Service Level Agreements (SLAs) turn into a basic perspective for a supportable cloud computing plan of action. In its essential frame, a SLA is an agreement between the CSPs and shoppers that indicates the terms and conditions under which the administration is to be given, including expected reaction time and unwavering quality. Inability to convey the administration as determined in the SLA subjects the CSP to pay a punishment, bringing about lost income.

Notwithstanding punishments coming about because of inability to meet the SLA necessity, CSPs confront rising energy expenses of their extensive scale datacenters. It is accounted for that energy costs alone could represent 23%–50% of the costs and this bill mounts up to \$30 billion overall [5]. This brings up the issue of how fault tolerance may affect control utilization and at last effect the earth.

Current fault tolerance approaches depend upon either time or equipment excess so as to endure disappointments. The main approach, which utilizes time repetition, requires the re-execution of the fizzled assignment after the disappointment is identified [6]. Despite the fact that this can further be advanced by the utilization of checkpointing and move back recuperation, such an approach can bring about a huge postpone increment subjecting CSPs to punishments, when SLA terms are disregarded, and high energy costs because of re-execution of fizzling assignments.

The second approach abuses equipment excess and executes numerous occasions of a similar assignment in parallel to beat disappointment and assurance that no less than one errand achieves finishing. This approach, which has been utilized widely to manage disappointment in basic applications, is as of now utilized as a part of cloud-computing to give fault tolerance while concealing the postponement of re-execution [7]. This arrangement, nonetheless, expands the energy utilization for a given administration, which thusly may exceed the benefit picked up by giving the administration. The exchange off amongst benefit and fault-tolerance calls for new systems to take both SLA necessities and energy awareness in managing disappointments.

Literature Survey

[8] display a measured usage model of the PSTR plot that can be consolidated into most business continuous working frameworks. This secluded usage model is amiable to a thorough examination of the recuperation time limits, a measure of awesome significance in complex frameworks.

[9] is a mix of the PSTR conspire and a network surveillance (NS) plot. This expansion brings about a critical change in the blame scope and recuperation time bound accomplished. The NS plot embraced is an as of late created conspire successful in an extensive variety of indicate point systems and it is known as the supervisor based NS (SNS) scheme.

[10] research about how much a dynamic load adjusting plate booking calculation in conjunction with tied declustering, a contrasting option to the traditional measuring plan, can react powerfully to varieties in a workload and circle disappointments.

[11]creates CMP memory frameworks for server combination where most sharing happens inside Virtual Machines (VMs). Our memory frameworks amplify shared memory gets to adjusted inside a VM, limit obstruction among particular VMs, encourage dynamic reassignment of VMs to processors and memory, and bolster content-based page sharing among VMs. We start with a tiled design where each of 64 tiles contains a processor, private L1 reserves, and a L2 bank.

[12]have demonstrated that the servers in the information escalated register bunches are under-used and, consequently, openings exist for better solidification of the workload on the Hot Zone. Examination of the hints of a Yahoo! Hadoop group indicated significant heterogeneity in the information's get to examples which can be utilized to guide vitality mindful information position arrangements.

[2] introduce a reenactment domain for vitality mindful distributed computing server farms. Alongside the workload dispersion, the test system is intended to catch subtle elements of the vitality devoured by server farm parts (servers, switches, and interfaces) and additionally parcel level correspondence designs in sensible setups.

[13] underlines the part of correspondence texture in server farm vitality utilization and presents a planning methodology that joins vitality efficiency and system mindfulness, named DENS. The DENS procedure adjusts the vitality utilization of a server farm, singular occupation execution, and traffic demands.

[14] we propose an Energy-Efficient Adaptive File Replication System (EAFR), which joins three parts. It is versatile to time-changing file popularities to accomplish a decent tradeoff between information accessibility and efficiency. Higher fame of a file prompts more reproductions and the other way around.

[15] we give an outline of the asset provisioning and usage designs in server farms and propose a full scale asset administration layer to arrange among digital and-physical assets. We audit some current work and arrangements in the field and clarify their constraints.

[16] address both versatility and power together, this is as opposed to a significant part of the contended work which does as such freely. Utilizing a systematic model that records for both power utilization and disappointments, we concentrate the execution of checkpoint and replication-construct methods in light of present and future frameworks and utilize control estimations from current frameworks to approve our findings

[17] demonstrates the famous impression is inaccurate; information systems are gently used contrasted with the phone organize. Indeed, even the spines of the Internet are keep running at lower portions (10% to 15%) of their ability than the exchanged voice arrange (which works at more than 30% of limit all things considered). Private line systems are used far less seriously (at 3% to 5%). Further, this circumstance is probably going to hold on.

[18] we concentrate the relative execution of three high accessibility information replication procedures, tied declustering, reflected plates, and interleaved declustering, in a common nothing database machine condition. Among the issues that we have inspected are (1) the relative execution of various procedures when no disappointments have happened, (2) the impact of a solitary hub disappointment on framework throughput and reaction time, (3) the execution effect of changing the CPU speed and additionally circle page estimate on the diverse replication systems, and (4) the tradeoff between the benefit of intra inquiry parallelism and the overhead of initiating and planning additional administrator procedures.

[11] we concentrate the connection between vitality administration, stack adjusting, and replication methodologies for information escalated bunch figuring. Specifically, we demonstrate that Chained Declustering – a replication technique proposed over 20 years back – can bolster extremely flexible vitality administration plans.

[19] we propose a versatile and power-mindful calculation, alluded to as Lazy Shadowing, as an efficient and adaptable way to deal with accomplish large amounts of strength, through forward advance, in extraordinary scale, failureprone registering conditions. Lethargic Shadowing partners with each procedure a "shadow" (process) that executes at a diminished rate, and artfully moves forward each shadow to make up for lost time with its driving procedure amid disappointment recuperation.

[20] propose two critical elements: (1) setting mindful transmission approach which controls how to transmit information efficiently as indicated by information need, battery level and system information rate; (2) an engineering that backings the model of portable intuitive applications in which customers and servers can cooperate with each other in an autonomic way.

PROBLEM DEFINATION

In the existing approach, shadow replication doesn't exist because whatever the work is done by the existing vm, if the faults occur, then the performance which is done by the existing vm will lost and if the new vm starts its working, then it will start work from the beginning and the entire work progress will be lost. In this, energy efficiency decreases. Energy conservation indicates utilizing more resources with minimum possible energy consumption. Energy conservation is not quite the same as proficient energy utilize, which alludes to utilizing less energy for a consistent administration. And here the need of Shadow Replication arises. By Shadow Replication, very efficiently all the work progress of the last vm, noticed and saved. So that if the fault occur, then the new vm will start performing from the

progress of the last vm. So here energy efficiency and time consumption also takes place. New vm don't need to start work from the very beginning that means all the energy and time which is used by the last vm in order to perform work doesn't waste. Because new vm will start work from the last vm progress. And here obviously utilization of time and energy consumption sustains.

1. The Fault tolerance capabilities are limited.
2. Energy-efficiency is limited in nature.
3. Time-consumption in executing cloudlets is higher.
4. Static cores of existing system cause higher energy consumption along with decremented fault-tolerance capabilities.

Objectives of study

The present reaction to fault tolerance depends upon either time or equipment excess with a specific end goal to handle faults. Time excess infers a re-execution of the fizzled calculation after the disappointment has been identified, in spite of the fact that this can further be improved by the utilization of checkpoints these arrangements still force a noteworthy deferral. In numerous mission basic frameworks equipment excess has customarily conveyed as process replication to give fault tolerance, keeping away from deferral and keeping up tight due dates. Both methodologies have downsides, re-execution requiring extra time and replication requiring extra assets, particularly vitality. This powers the frameworks specialist to pick between time or equipment excess, distributed computing conditions have to a great extent picked replication since reaction time is regularly basic. In this paper we propose another computational model called shadow computing, which gives objective based versatile flexibility using dynamic execution.

1. The Fault tolerance capabilities are enhanced by the use of proposed system.
2. The objectives of study are to enhance the energy-efficiency.
3. Time-consumption or latency has to be minimised.
4. Comparing the performance of static core based approach with dynamic core based approach.

ALGORITHM

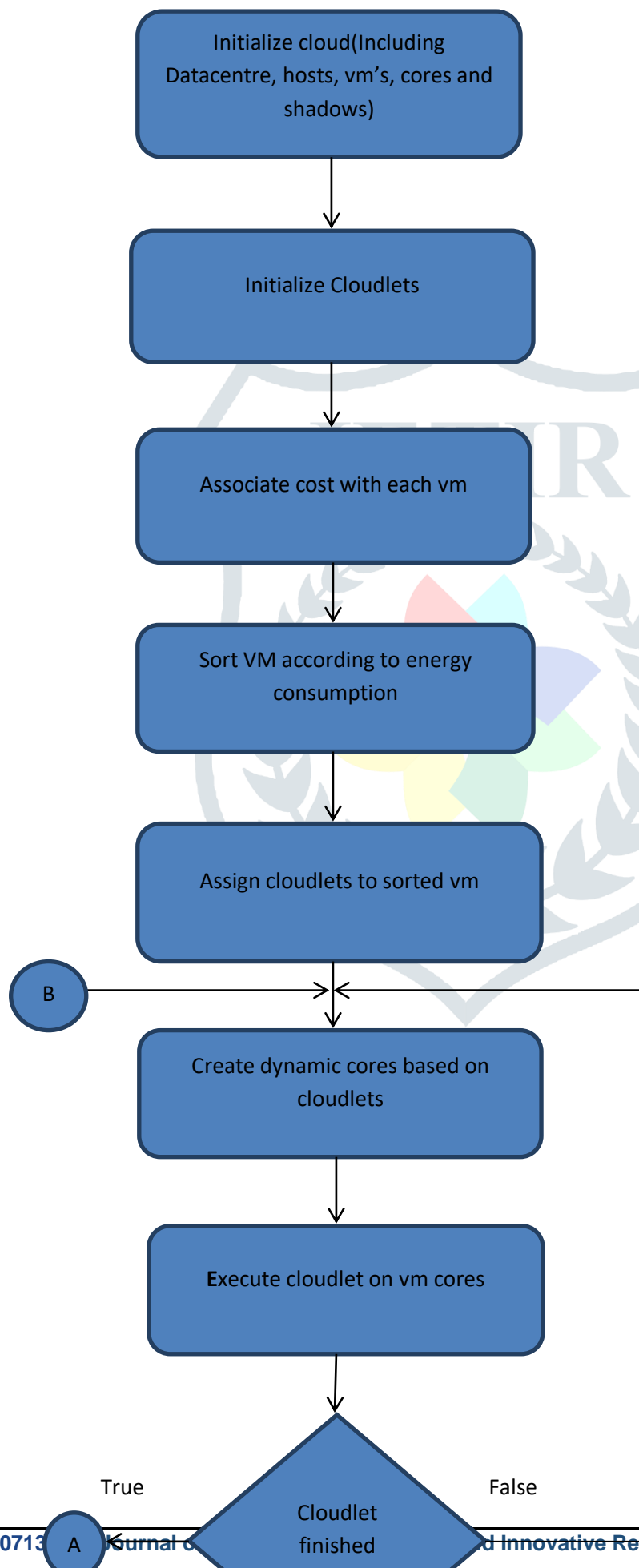
- Input : Cloudlets
- Output: Latency, Energy Consumed, Fault Rate, Cost.

- a. Initialize cloud:
 - Including Datacentre, hosts, vm's, cores and shadows.
- b. Assign Cost with Each vm.
- c. Sort the vm's according to Energy consumption.
- d. Assign cloudlets to minimum Energy Consumption VM.
- e. Create dynamic cores based on cloudlet.
- f. Execute cloudlet on vm cores.
- g. Check Energy Consumption

$$\text{Energy Consumption} = \frac{\text{CPU}\%}{P1+P2*\text{CPU}\%} * (P1 + P2)$$

- h. Check for cores failure due to memory overflow. If core fails look for another core vm or vm within same host.
- i. If Energy :-> threshold then vm fails and vm migration is required.
- j. Copy progress from Shadow failed core or vm to new machine selected.
- k. Output latency, Energy Consumed, Fault Rate, Cost.

PROPOSED FLOWCHART



A

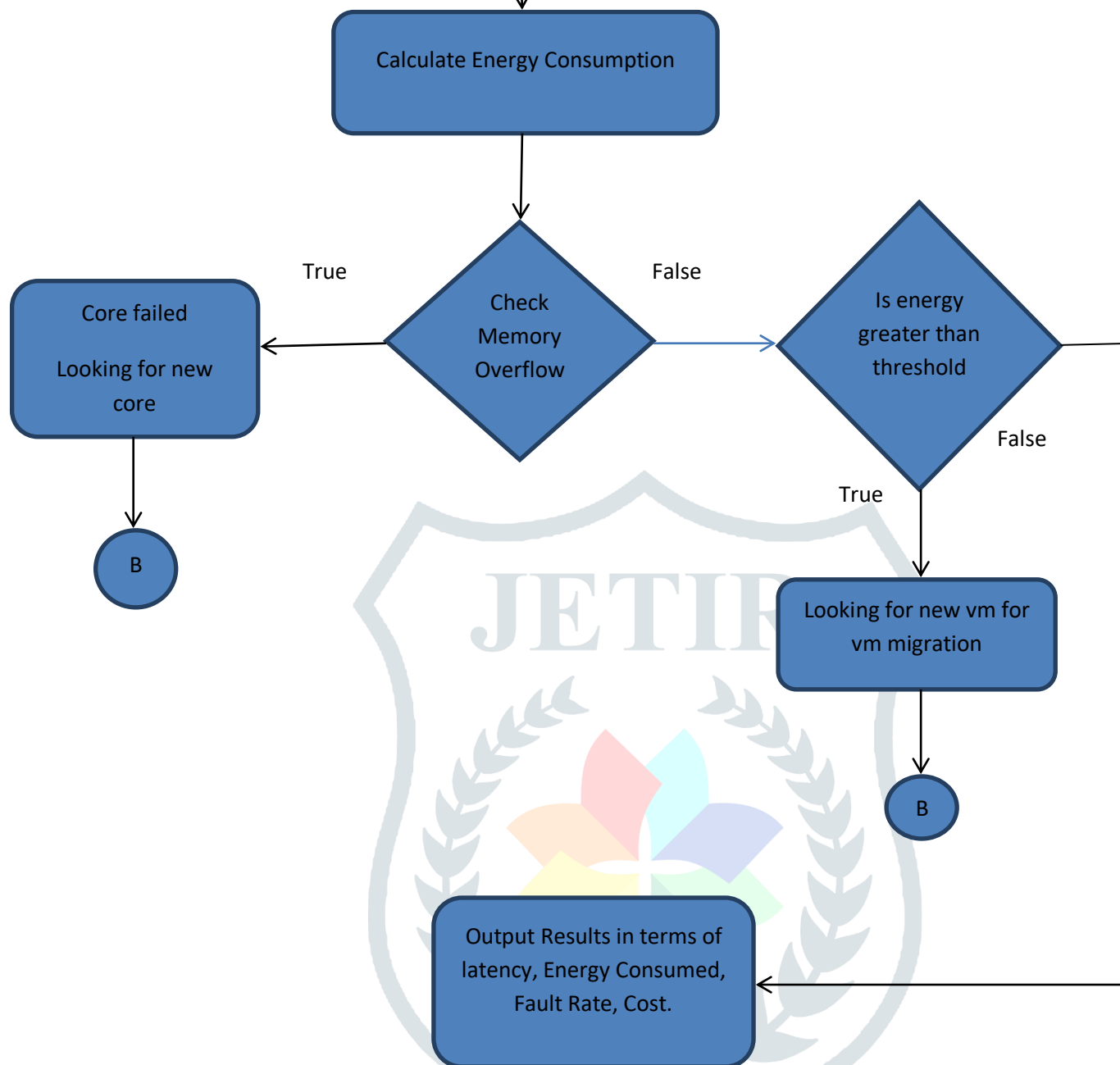


Figure 1:Proposed Flowchart

Results and performance Analysis

The results associated with the dynamic core model of the proposed system are compared against static core model and following results have been obtained.

1. Cloudlet size 90000

Parameter	existing	proposed
Average energy consumed	40.938152	18.412603
latency	14096	10755
cost encountered	328	275
fault tolerant rate	69	61

Table 1: Comparison of parameters in case of cloudlet size 90000

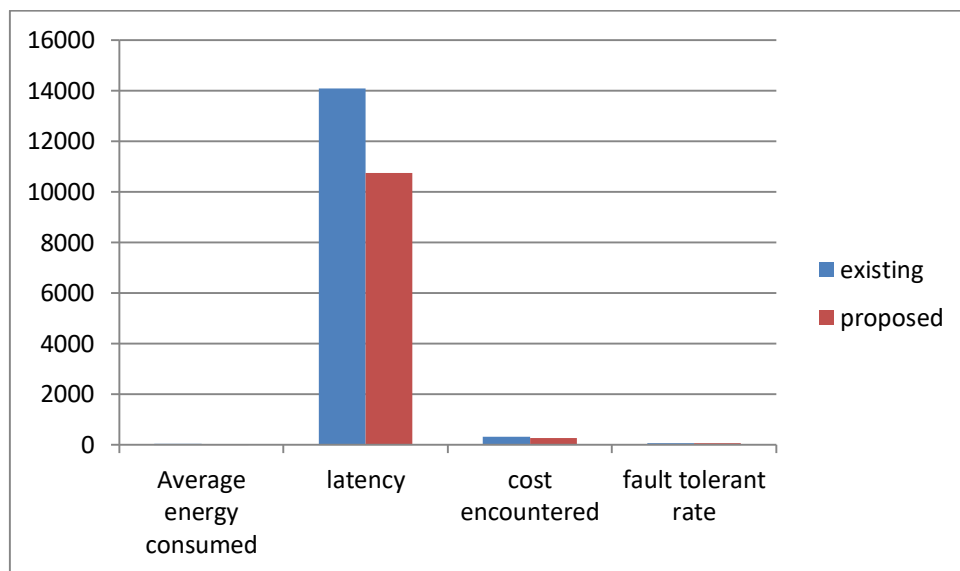


Figure 2: Comparison of parameters with cloudlet size 90000

2. Cloudlet size 85000

	existing	proposed
Average energy consumed	54.405746	14.436229
latency	13079	10503
cost encountered	898	159
fault tolerant rate	49	82

Table 2: Cloudlet size 85000 parameter comparison

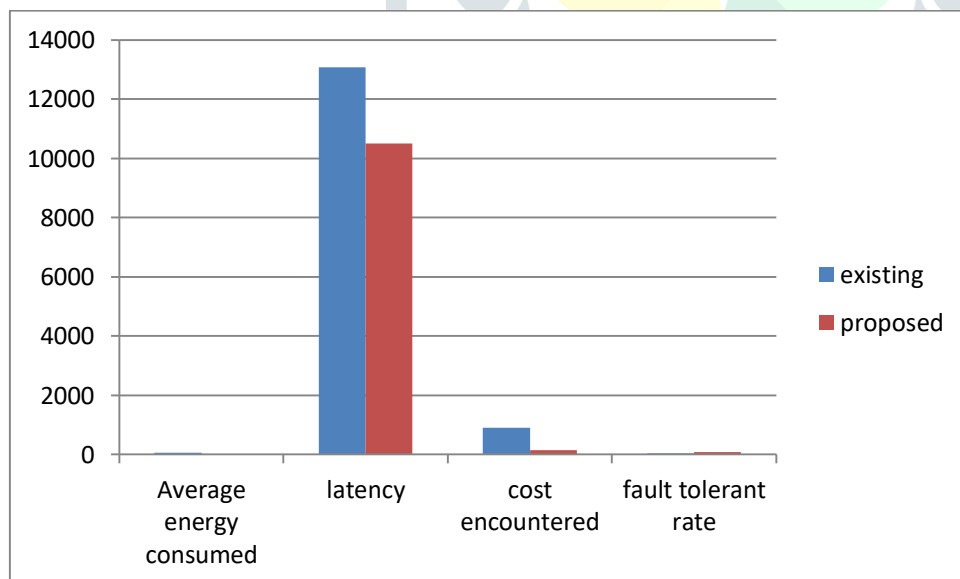


Figure 3: Comparison of parameters with cloudlet size 85000

3. Cloudlet size 80000

	existing	proposed
Average energy consumed	51.703804	14.504681
latency	11532	9038
cost encountered	793	276
fault tolerant rate	49	83

Table 3: Cloudlet size 80000 parameter comparison

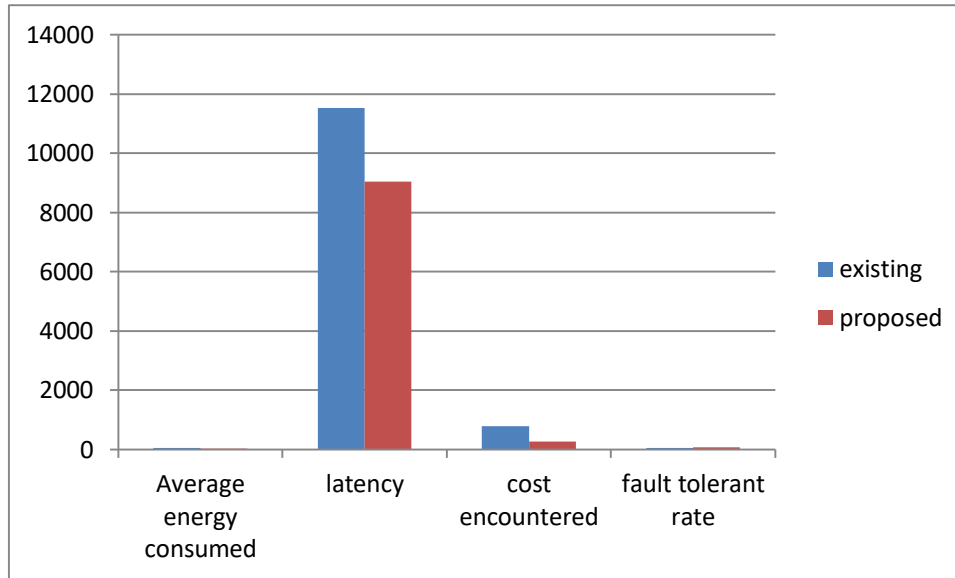


Figure 4: Comparison of parameters with cloudlet size 80000

4. Cloudlet size 75000

	existing	proposed
Average energy consumed	46.09626	25.330738
latency	14574	12533
cost encountered	637	164
fault tolerant rate	54	79

Table 4: Cloudlet size 75000 parameter comparison

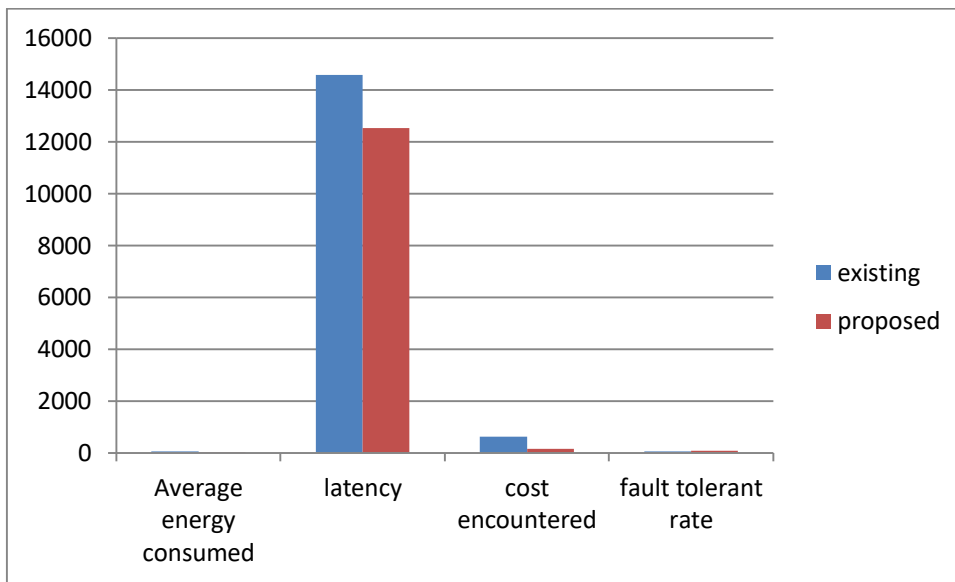


Figure 5: Comparison of parameters with cloudlet size 75000

5. Cloudlet size 70000

	existing	proposed
Average energy consumed	54.82179	5.2413588
latency	11893	18173
cost encountered	751	157
fault tolerant rate	51	76

Table 5: Cloudlet size 70000 parameter comparison

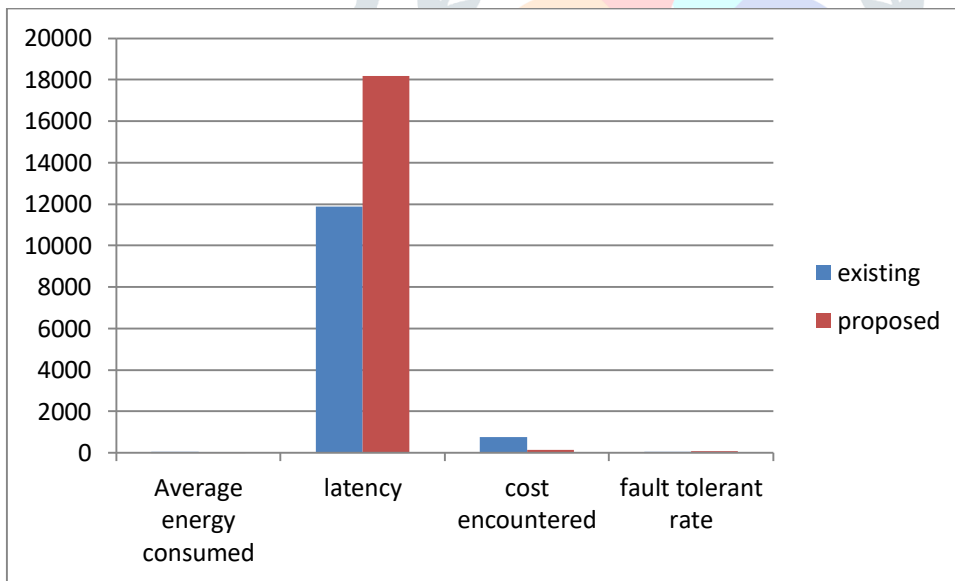


Figure 6: Comparison of parameters with cloudlet size 70000

Results in terms of proposed system shows significant improvement proving worth of study.

CONCLUSION AND FUTURE SCOPE

Static cores associated with VMs consume more energy along with cost. The cost is enhanced since core failure shift the load on another core within same VM hence enhancing overall load on the machine and hence cost associated with cloudlet execution. The problem is resolved by the use of dynamic core in which cores are created on the basis of cloudlet size. Energy efficiency is also achieved since extra core remain in sleep mode until load is not allocated on them. Fault tolerance rate is also enhanced due to the application of dynamic cores.

In future, fault tolerant strategy combined with ACO for resource allocation in cloud can be used for improving energy efficiency.

REFERENCES

- [1] B. Meroufel and G. Belalem, "Adaptive time-based coordinated checkpointing for cloud computing workflows," *Scalable Comput.*, vol. 15, no. 2, pp. 153–168, 2014.
- [2] D. Kliazovich, P. Bouvry, and S. U. Khan, "GreenCloud : A Packet-level Simulator of Energy- aware Cloud Computing Data Centers," *J. Supercomput.*, vol. 62, no. 3, pp. 1263–1283, 2012.
- [3] B. Alami Milani and N. Jafari Navimipour, "A comprehensive review of the data replication techniques in the cloud environments: Major trends and future directions," *J. Netw. Comput. Appl.*, vol. 64, pp. 229–238, 2016.
- [4] R. Balamaniandan, "Analyzing massive machine data maintaining in a cloud computing," vol. 23, no. 10, pp. 78–81, 2013.
- [5] D. Singh, J. Singh, and A. Chhabra, "High availability of clouds: Failover strategies for cloud computing using integrated checkpointing algorithms," *Proc. - Int. Conf. Commun. Syst. Netw. Technol. CSNT 2012*, pp. 698–703, 2012.
- [6] Y. Zhang, Z. Zheng, and M. R. Lyu, "BFTCloud: A Byzantine Fault Tolerance framework for voluntary-resource cloud computing," *Proc. - 2011 IEEE 4th Int. Conf. Cloud Comput. CLOUD 2011*, no. July 2011, pp. 444–451, 2011.
- [7] P. K. Szwed, D. Marques, R. M. Buels, S. A. McKee, and M. Schulz, "SimSnap: Fast-forwarding via native execution and application-level checkpointing," *Proc. - Eighth Work. Interact. between Compil. Comput. Archit. INTERACT-8 2004*, pp. 65–74, 2004.
- [8] K. H. Kim and C. Subbaraman, "A modular implementation model of the Primary-Shadow TMO replication scheme and a testing approach using a real-time environment simulator," *Softw. Reliab. Eng. 1998. Proceedings. Ninth Int. Symp.*, pp. 247–256, 1998.
- [9] K. H. Kim and C. Subbaraman, "An Integration of the Primary-Shadow TMO Replication (PSTR) Scheme with a Supervisor-based Network Surveillance Scheme and its Recovery Time Bound Analysis," *Proc. SRDS '98 (IEEE CS 17th Symp. Reliab. Distrib. Syst. 1998, pp.168-176.*, pp. 168–176, 1998.
- [10] "chain_declustering.pdf." .
- [11] M. R. Marty and M. D. Hill, "Virtual hierarchies to support server consolidation," *ACM SIGARCH Comput. Archit. News*, vol. 35, no. 2, p. 46, 2007.
- [12] R. T. Kaushik, "GreenHDFS : Towards An Energy-Conserving , Storage-Efficient , Hybrid Hadoop Compute Cluster," *HotPower*, pp. 1–9, 2010.
- [13] D. Kliazovich, P. Bouvry, and S. U. Khan, "DENS: Data center energy-efficient network-aware scheduling," *Cluster Comput.*, vol. 16, no. 1, pp. 65–75, 2013.
- [14] Y. Lin and H. Shen, "EAFR: An Energy-Efficient Adaptive File Replication System in Data-Intensive Clusters," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 4, pp. 1017–1030, 2017.
- [15] J. Liu, F. Zhao, X. Liu, and W. He, "Challenges Towards Elastic Power Management in Internet Data Centers," *2009 29th IEEE Int. Conf. Distrib. Comput. Syst. Work.*, pp. 65–72, 2009.
- [16] B. Mills, T. Znati, R. Melhem, K. B. Ferreira, and R. E. Grant, "Energy consumption of resilience mechanisms in large scale systems," *Proc. - 2014 22nd Euromicro Int. Conf. Parallel, Distrib. Network-Based Process. PDP 2014*, pp. 528–535, 2014.
- [17] A. Odlyzko, "Data Networks are Lightly Utilized, and will Stay that Way," *Rev. Netw. Econ.*, vol. 2, no. 3, pp. 210–237, 2003.
- [18] H.-I. Hsiao and D. J. DeWitt, "A performance study of three high availability data replication strategies," *[1991] Proc. First Int. Conf. Parallel Distrib. Inf. Syst.*, pp. 18–28.
- [19] X. Cui, T. Znati, and R. Melhem, "Adaptive and Power-Aware Resilience for Extreme-scale Computing."
- [20] C. S. Shih and T. K. Trieu, "Shadow phone: Context aware device replication for disaster management," *Proc. - 2012 5th IEEE Int. Conf. Serv. Comput. Appl. SOCA 2012*, 2012.