DOI: http://dx.doi.org/10.26483/ijarcs.v8i8.4673

ISSN No. 0976-5697

Volume 8, No. 8, September-October 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

IMPROVED MIGRATION STRATEGY USING ENERGY AWARE FAULT TOLERANCE CHECKPOINTING

Tanvi
M.tech Scholar
SSIET,Dinanagar,India

AasthaMahajan
H.O.D Computer Science
SSIET,Dinanagar,India

Abstract: Server co-solidification and fault tolerance is one of the strategies to decrease the general resource utilization in the data centers other than different equipment and programming highlights. All such techniques provide energy efficient mechanism during allocation and reallocation of cloudlets to virtual machines. Virtualization techniques (VT) have made this possible with the assistance of different hypervisors responsible for creating virtual machines (VMs). Combining approaches for energy optimization along with fault tolerance, there are different parameters that should be tossed light into including execution, SLA (Service Level Agreement), CPU-I/O connection and so forth. Live migration of VM is the key to combining the servers without ceasing them in this manner with close to zero downtime for the frameworks. Unreasonable combination causes execution corruption which has extreme effect on the QoS (Quality of Service) of the application that are running in that condition. VM allotment and position are the principle issues in consolidation. Fault tolerance in proposed methodology is achieved through checkpoint with write through approach. The execution and energy consumption tradeoffs have been talked about in view of Greedy heuristics. At last the results are given a throw in terms of temperature, energy consumption, downtime and migration time.

Keywords:Server Consolidation, Energy Optimization, SLA, CPU-I/O, Live VM migration, temperature, Energy Consumption,Checkpoint, Downtime, Migration time

1. INTRODUCTION

Cloud computing provide anything as a service (XaaS) to the users on the basis of pay per use. [1]Describes cloud infrastructure and mode of operation. [2]Propose data access on various levels through cloud computing. [3]Cloud provides shared resources in terms of data deduplication, platform and other information to computer and other devices that need it. [4]Suggests distributed computing is an advertising or marketing term for different technologies that give computation, programming and capacity services that don't require end-client learning of the physical area and arrangement of the framework that conveys the services.
[5]VMs(Virtual Machines) refers to one instance of operating system that consumes certain resources as per need and load over the operating system. [6]VMs executes on isolated partition over the computer system. There could be multiple virtual machines running over single physical machine.

As the workload increases, [7]host may decide to distribute load to some other machine with optimal energy consumption or other parameters such as temperature, response time etc. The process of distributing load onto some other optimal machine from current machine is known as migration.
In the past, in order to migrate VM, it is necessary to shut down the host and then move the VM files and start the host again causing enhancement in downtime. [8]proposes a solution to this problem, which is live VM migration in which downtime is considerably reduced. This transfer process does not really mean shutting VM down permanently rather state of the VM is shifted. Hence VM is not rendered unusable. Transfer of state includes tasks, memory and internal state of the device.
Two parameters are considered for evaluation during live VM migration.

1. Downtime
2. Migration time

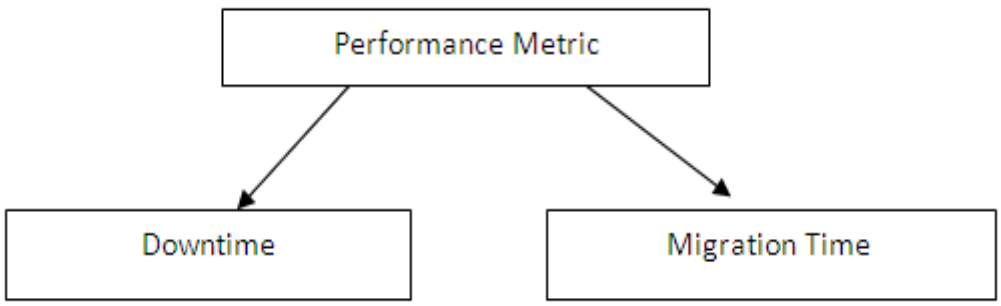

Figure 1: Performance metric in Live VM migration

Downtime:[9]Downtime is the time during which host is disabled and unable to perform any task. This is accomplished by switching it off during task and memory migration to some other optimal VM. There exists a tradeoff between performance and Downtime.

Migration Time:[10]Migration time is the time required to shift the load on some other VM.tradeoff as existed in downtime also exists in Migration time and performance.

Proposed literature deals with enhancing performance by decreasing both of these metrics. There exists phases in the proposed literature that are used for enhancing performance considering energy consumption, rise in temperature, Memory and fan speed. The flow of proposed system is listed as under

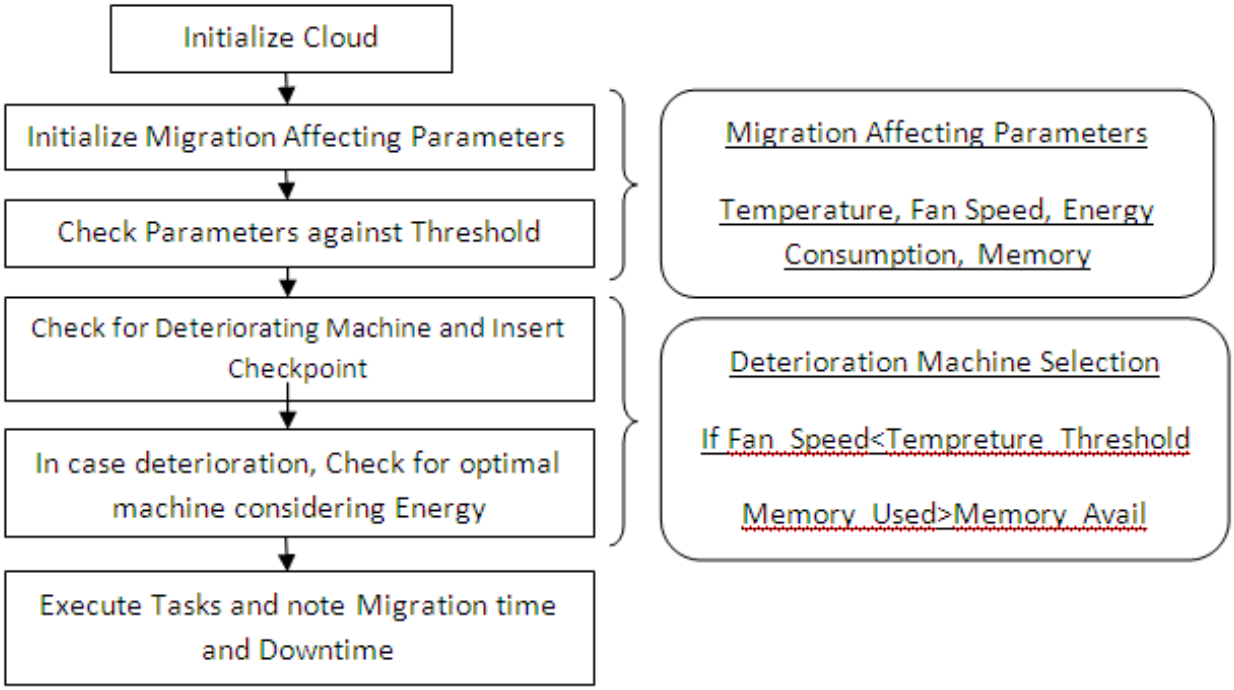

Figure 2: Flow of proposed system

To prove worth of study existing literature is studied. This study of literature is described as under

2. BACKGROUND

possible to perform computation in wearable devices. Set of policies are formulated for controlling CPU cycles in case of local computing and offloading for other mode of computing.

[11]suggests energy constraint mechanism to ensure job execution efficiently. Code migration is suggested to optimize energy efficiency. Pre-copy with remote execution takes place. With remote execution, job executes from the remote server. In case of deterioration, job is migrated through code and hence progress of job is saved and it is executed again from the place it is stopped on previous machine. Results show considerable improvement in terms of downtime and migration time.

[12]researched a task computing and cost of file offloading to minimize energy consumption. Radio resource allocation is primarily considered in this literature. Energy efficient computational offloading(EECO) on 5G network is proposed in this paper. Uplink and Downlink transmission rate is considered through the following equations.

Uplink Transmission Rate

$$r_{i,_k} = Wlog_2(1 + \frac{p_i^M g_i^M}{I^S + \sigma^2})$$

Equation 1: Uplink Transmission Rate

Where 'P' is the power of mobile device, 'I' denotes the interference, 'g' indicate the channel gain, 'σ' is the noise.

Downlink Transmission Rate

$$r_{i,_k} = Wlog_2(1 + \frac{p_i^M g_i^M}{I^M + \sigma^2})$$

Equation 2: Downlink Transmission Rate

Channel for accessing used is M. Cost under the delay constraint is reduced considerably.

[4] Proposes a decentralized approach for mobile computational offloading. Decentralized approach follows multiple virtual machines on which load is distributed. The computation is considerably reduced on individual machine. The energy efficiency is achieved since priority while allocation is considered. Results indicate improved performance.

[13] Proposes duty cycling mechanism to achieve energy efficiency in scheduling of resources in wireless sensor network. Duty cycling is divided into power management and topology control mechanisms. Node redundancy is considered in topology control and power management is considered in case of sensor allocation. Sensors have limited power and energy associated with them. This work effectively manages both energy and power and hence a result obtained is better in terms of energy efficiency. Minimum load a node can tackle is given through the following equation.

$$\gamma\big(L(G)\big) \leq K_n(G) \leq K_E(G) \leq \min(\deg(G))$$

Equation 3: Load equation for nodes

'G' indicates the graph of the form G={V,E}, 'V' is the set of vertex and 'E' is the set of edges. 'n' indicates total number of nodes.
[14]consider both dynamic power as well as leakage power for energy efficiency during scheduling. Precedence constraint is employed in this case. Jobs hence are executed in terms of precedence rather than sequential. The execution time is calculated in terms of following equations.

$$C_m = (1-\beta) * CB_m + \beta * CW_m$$

Equation 4: Execution time calculation.
Jobs are executed on 1, 2 and 4 cores for checking the power consumption. Results show better scheduling as compared to other scheduling mechanisms. [15]

2.1 PRE COPY
[16], [17]In pre copy approach the transfer from memory to the destination are to be done first and then limit the iteration reaches.

Algorithm
1. In destination node the memory and VCPUs are restrained first
2. A scan on page writes is initiated and all contents from source RAM are transferred to destination when relocation is issued.
3. In next step until iteration limit is reached the pages have been transferred.
4. When all transfer has to be done then the source is stopped and current state of CPU registers. After that state of virtual device and last memory pages are transferred to destination.
5. At destination the VM is resumed. .

The number of remaining pages to be copied for a given point in time t is then determined by
f(t) = e(t) + p(t) + h(t)

2.2 POST COPY
[18], [19]In post copy the transfer of device state and VCPU is transferred first on destination and then the execution on destination starts. The steps which are performed as given below
1. The VM at source are stopped
2. The states of devices are copied and VCPU registers on the destination VM.
3. Execution at destination are resumed
4. If not yet fetched page is accessed by VM then Page Fault occurs and page is transferred to the destination.

The mathematical formula for calculating

Source Contention=

$$\sum_{i=1}^{n} \text{Rate of outgoing traffic in Mbps for VM , if migrated with pre copy + outgoing background traffic}$$

Destination Contention
=

$$\sum_{i=1}^{n} \text{Rate of incoming traffic in Mbps for VM , if migrated with post copy + Incoming background traffic}$$

2.3 CHECKPOINT
[20], [21]The checkpoint is a mechanism that is used to back up the data before the updates are done on VM in live migration. The administrator can return the virtual machine to its state prior to the update. The action that will used to return the state to checkpoint is recover action. Each virtual hard disk that is attached to each virtual machine uses checkpoint for each to save the state. The recover action is utilized after the creation of checkpoint to restore the virtual machine.
The logs are maintained in real time environment till all the memory spaces fill out. In the checkpoint mechanism all the previous logs are removed from the system and stored in storage disk permanently.

3. PROPOSED SYSTEM

Proposed system uses hybrid approach of migration and check pointing. The used technique provide better performance in terms of downtime and migration time. The proposed algorithm is listed as under
A. Algorithm for energy-aware fault tolerance
Algorithm EAFT (Energy Aware Fault tolerance)

Input: load
/*Variable initialization*/
host list: all the active hosts, VM list: all vms currently available in the pool, Finalized_VM: sorted list of vms consuming minimum energy, Threshold(temp): 60, Threshold(energy): 250

1. Initialize Fan speed and temperature to each VM.
2. Evaluate power consumption of each VM using equation

$Power_i = P1 + P2 * CPU_{Utilization_i}$ -------Equation 1

3. Evaluate energy consumption of each VM using equation

$Energy_i = \frac{CPU_{Utilization_i}(P1+P2)}{Power_i}$----------Equation 2

4. Sort VMs on the basis of energy consumption
Finalized_VM$_i$= VM$_{Min}$
5. Assign Load to Finalized_vm
6. Check for deteriorating vm
If (Temperature>Threshold and Fan_Speed$_i$<Temperature)
Else (Energy>Threshold)
Migrate VM to optimal destination selected from Finalized_VM list
7. Use Equation 1 and 2 to check for power and energy consumption
8. **Output:** Downtime, Migration time, Average Energy Consumption and Latency

The results presented gives better result as described in next section.

4. RESULT AND PERFORMANCE ANALYSIS

We have considered three parameters within the result and performance analysis i.e energy consumption, downtime and migration time . The result in terms of their parameters is as under:

Table 1 showing energy consumption corresponding to existing and proposed approach.

SIMMULATION	Cloudlet	EXISTING SYSTEM	PROPOSED SYSTEM
Simulation 1	600	278.48314	220.6013
Simulation 2	700	290.45565	221.23
Simulation 3	800	285.42105	220.8621
Simulation 4	900	270.09743	232.5
Simulation 5	1000	262.55665	222

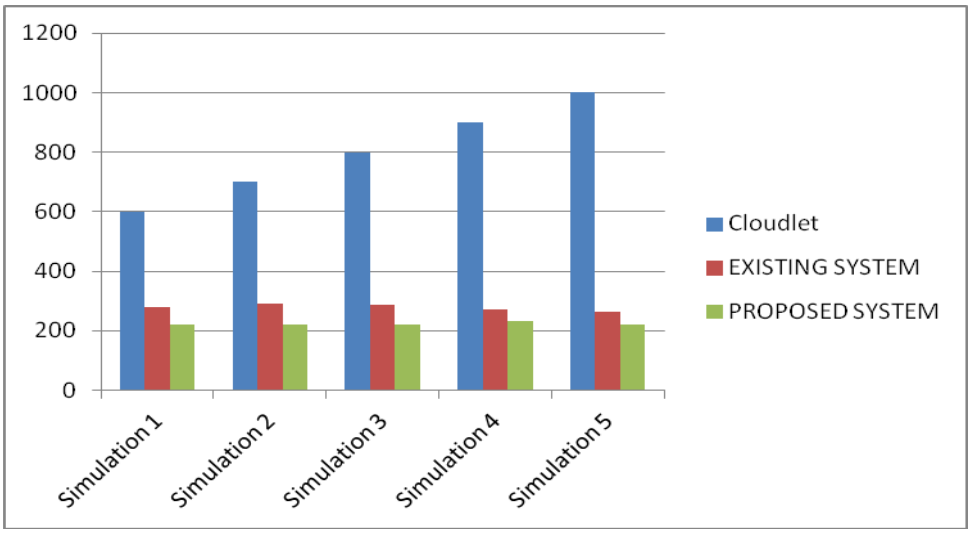

Figure 3: Results in terms of energy consumptionDowntime

Table 2: Downtime comparison of existing and proposed literature

SIMMULATION	Cloudlet	EXISTING SYSTEM	PROPOSED SYSTEM
Simulation 1	600	312	196
Simulation 2	700	325	201
Simulation 3	800	326	205
Simulation 4	900	329	206
Simulation 5	1000	330	207

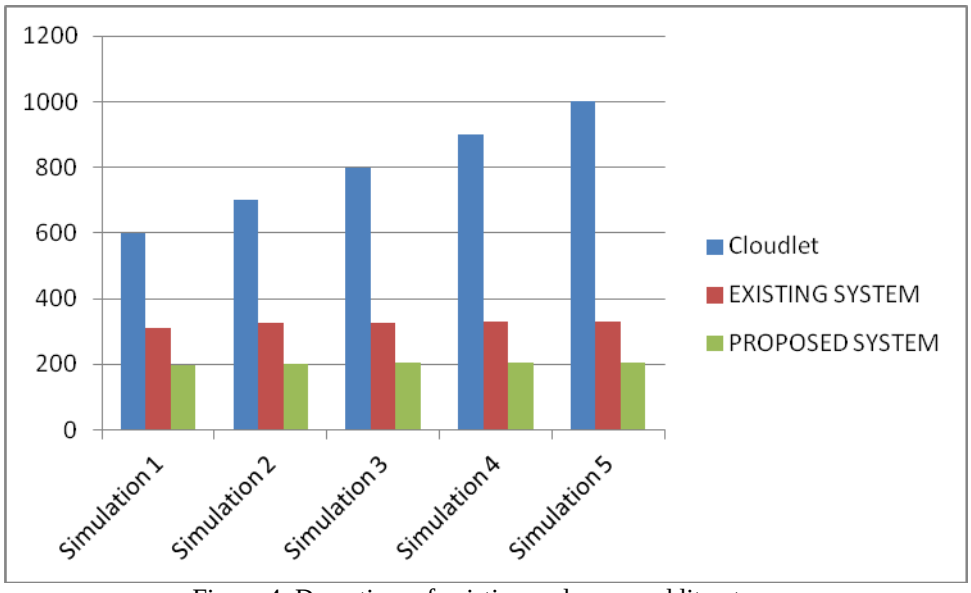

Figure 4 :Downtime of existing and proposed literature

Migration Time

Table 3: Migration time comparison of existing and proposed literature

SIMMULATION	Cloudlet	EXISTING SYSTEM	PROPOSED SYSTEM
Simulation 1	600	6650	5910
Simulation 2	700	6852	6000
Simulation 3	800	6986	6050
Simulation 4	900	7026	6150
Simulation 5	1000	7256	6235

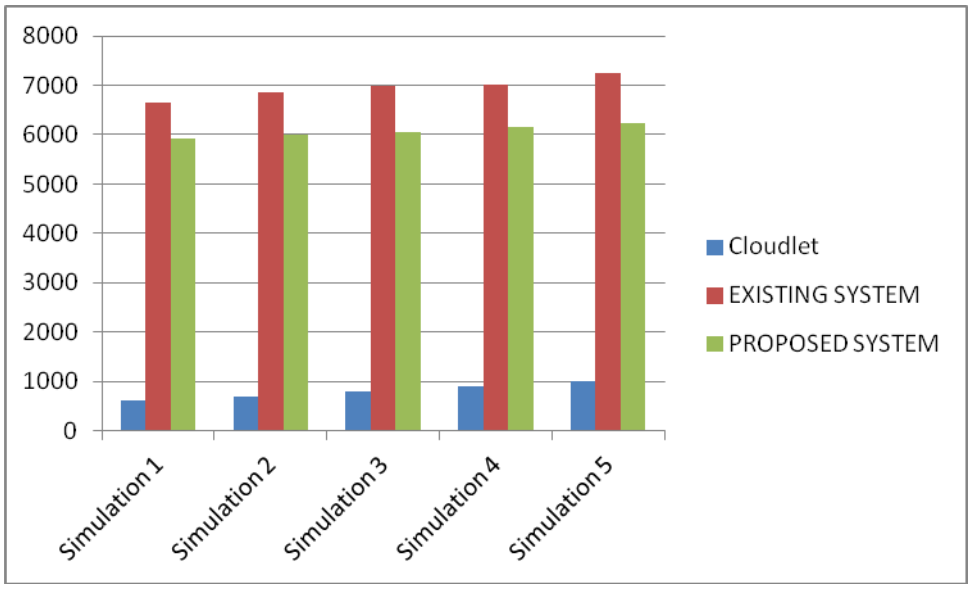

Figure 5: Migration time comparison

5. CONCLUSION

As load on data center increases, energy consumption increases. Energy consumption in case increases beyond threshold levels, normal operation of VM is disturbed. In order to overcome the problem power aware mechanism through proposed literature is suggested. Proposed work detect deteriorating machine in phases. Temperature and fan speed is used as prime parameters for initial deteriorating VM detection. In case deterioration is detected migration is performed on next optimal virtual machine sorted on the basis of energy consumption. Next parameter used to detect deterioration is energy consumption, higher the energy consumption more will be the chance of deteriorating machine. Energy consumption is dynamically varied by changing cpu utilization as load increases on VM. Dynamic relocation strategy is used for next optimal vm selection. Results in terms of energy consumption is specified. 25% conservation of energy is noted.

6. REFERENCES

[1] Y. Charalabidis, M. Janssen, and O. Glassey, "Introduction to Cloud Infrastructures and Interoperability Minitrack," p. 7695, 2010.

[2] K. Yang and X. Jia, "Expressive, Efficient, and Revocable Data Access Control for Multi-Authority Cloud Storage," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 7, pp. 1735–1744, Jul. 2010.

[3] X. Zhang, Z. Huo, J. Ma, and D. Meng, "Exploiting Data Deduplication to Accelerate Live Virtual Machine Migration," 2012.

[4] X. Chen, "Decentralized Computation Offloading Game for Mobile Cloud Computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974–983, Apr. 2012.

[5] S. B. Rathod and V. K. Reddy, "Secure Live VM Migration in Cloud Computing : A Survey," vol. 103, no. 2, pp. 18–22, 2014.

[6] M. Zhang, H. Jin, X. Shi, and S. Wu, "VirtCFT : A Transparent VM-Level Fault-Tolerant System for Virtual Clusters," pp. 147–154, 2014.

[7] J. Xu and J. A. B. Fortes, "Multi-objective Virtual Machine Placement in Virtualized Data Center Environments," 2014.

[8] Y. Zhong, J. Xu, Q. Li, H. Zhang, and F. Liu, "Memory State Transfer Optimization for Pre-copy based Live VM Migration," pp. 290–293, 2014.

[9] S. Sharma and M. Chawla, "A three phase optimization method for precopy based VM live migration," *Springerplus*, 2014.

[10] V. Verroios and M. Roussopoulos, "Time-Constrained Live VM Migration in Share-Nothing IaaS -Clouds," 2015.

[11] D. Marculescu, N. H. Zamora, P. Stanley-marbell, and R. Marculescu, "Fault-Tolerant Techniques for Ambient Intelligent Distributed Systems-," pp. 348–355, 2015.

[12] K. Zhang, Y. Mao, S. Leng, Q. Zhao, L. Li, and X. Peng, "Energy-efficient Offloading for Mobile Edge Computing in 5G Heterogeneous Networks," vol. 3536, no. c, pp. 1–10, 2015.

[13] L. Aslanyan, H. Aslanyan, and H. Khosravi, "Optimal node scheduling for integrated connected-coverage in wireless sensor networks," *CSIT 2013 - 9th Int. Conf. Comput. Sci. Inf. Technol. Revis. Sel. Pap.*, 2015.

[14] I. Transactions, O. N. C. Design, and O. F. Integrated, "Reliability-Driven Energy-Efficient Task Scheduling for Multiprocessor Real-Time Systems," vol. 30, no. 10, pp. 1569–1573, 2015.

[15] D. Miraculine, "Efficient Data Transmission during Virtual Machine Failures Using Hybrid Copy Live Migration," pp. 119–126, 2015.

[16] D. Kapil, E. S. Pilli, and R. C. Joshi, "Live virtual machine migration techniques: Survey and research challenges," in *2013 3rd IEEE International Advance Computing Conference (IACC)*, 2015, pp. 963–969.

[17] G. J. Jeincy, "Space Prediction for Cloud Computing," 2016.

[18] W. Zhang, K. T. Lam, and C. L. Wang, "Adaptive Live VM Migration over a WAN: Modeling and Implementation," in *2014 IEEE 7th International Conference on Cloud Computing*, 2016, pp. 368–375.

[19] P. Kaur and A. Rani, "Virtual Machine Migration in Cloud Computing," vol. 8, no. 5, pp. 337–342, 2016.

[20] J. Devale, "Checkpoint / Recovery : Overview Checkpointing - Recovery," *Memory*, 2016.

[21] J. Hursey, J. M. Squyres, T. I. Mattox, and A. Lumsdaine, "The Design and Implementation of Checkpoint / Restart Process Fault Tolerance for Open MPI ∗.,2016"