www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 7 Issue 7 July 2018, Page No. 24130-24135
Index Copernicus Value (2015): 58.10, 76.25 (2016) DOI: 10.18535/ijecs/v7i7.04

Achieving Energy and Storage Efficiency during Computational Offloading Using EECOC

Mr. Harpreet Singh, Dr. Sandeep Sharma

Mtech Student with department of computer Engineering and Technology Guru Nanak Dev University Amritsar india
Professor and Head in Department of Computer Engineering and Technology at Guru Nanak Dev University Amritsar India

Abstract
Achieving energy efficiency during computational offloading is the critical and is the prime objective of this work. Energy efficient Computational offloading with CO2 reduction (EECOC) is proposed for achieving energy efficiency. Load balancing and CO2 minimization are the control parameters considered during the work. Green computing is accomplished when CO2 is minimized. Hence load balancing and computational offloading leads to the desired objective. Run time migration is considered to reduce the overhead in terms of idle machines. CO2 is emitted by VMs in case they are over burdened. Computation offloading is accomplished to reduce load on VMs hence reducing levels of CO2. Load distribution considers the configuration of VMs. Machines with higher configuration is selected at first place for allocation. Load is further reduced by removing redundancy in given job. Results indicate that Energy efficient Computational offloading with CO2 reduction reduces energy consumption by 23%.

Keywords—EECOC, Energy Efficiency, Load Balancing, CO2, Green Computing, Redundancy.

Introduction

[1]**Cloud** computing is used to provide user with the resources more than capability of their physical machines. Cost is encountered on the basis of pay per use. [2], [3]As load on VMs in cloud increases, performance degrades. This degradation in performance also cause VMs to emit CO2 . Load on VMs required to be balanced to reduce CO2. Computational offloading is used to reduce load and CO2 levels. Energy Efficiency is achieved by considering [4]live VM migration in case of deterioration occurs in VMs. [5], [6]Energy efficient scheduling is also used for achieving optimization during migration. load can be further reduced by considering redundancy factor. The jobs containing similar work can be eliminated from the job to reduce load on the VM. This reduces the load and subsequently emission of CO2 levels. [18]Proposes energy efficient approach for reducing CO2 out of VMs. The technique uses temperature as a base mechanism for reducing CO2 and enhancing energy efficiency. Temperature levels are considered as threshold with which temperature of VMs are compared. In case temperature of VMs enhances beyond threshold values than workload from current VM is offloaded to next VM in sequence. Proposed methodology is based on enhancing performance and reducing load, hence reducing emission of CO2 levels. Rest of the paper is organised as under.

Section 2 describes scheduling and its objectives in advanced computing. Section 3 describes the existing work which is done in order to achieve energy efficiency. Section 4 describes the proposed work. Section 5 gives the result and performance analysis and next section gives conclusion and future work.

Scheduling And Its Objectives

Scheduling in multi cluster environment is critical area of research. Legion of scheduling algorithms are playing a part to schedule resources in multi cluster environment. Task can arrive from geographical large area and this present a challenge that which task must be given a resource at first place. In multi cluster environment Computer, data, and other resources are shared[7]. The service provider and consumers must agree upon what to be shared within given environment. The global nature of resource sharing is unique facility provided within multi cluster environment. The resource may be present within different administrative domain and demanded by node belonging to some other domain[8]. Scheduling henceforth becomes important for optimal and deadlock free system.

Scheduling is the process of assigning resources to jobs based on objective functions defined. Type of scheduling depends upon the objective function associated with the resource. Scheduling resources has following phases associated with it.

- Resource Discovery
- Resource Filtering
- Resource Selection
- Resource Scheduling Policy

Before allocation within multi cluster environment, resources must be discovered. Resource may or may not be available[7]. Hence this phase becomes critical for monitoring of resource within the system[9]. The available resources must be checked to determine whether they satisfy the requirements or not. Hence filtering is compulsory. Resource selection out of available resources is next phase. This phase is critical since out of available resources of same type resource with optimal condition is selected for allocation. This is required so that job can be completed well within time. Healthier resource selection is the target of this phase. Scheduling policy decides the resource allocation is primitive or not. Resource allocation is said to be primitive if once allocation resource can be prompted from task even if it is not yet fully completed.

Study Of Literature

Energy efficiency is critical while resources are allocated to VMs. Work has been done towards this aspect. This section describes the techniques used to achieve energy efficiency by reducing load using computational offloading mechanism.

[10]propose energy efficient mobile cloud computing using wireless energy transfer. The technique combines mobile cloud computing and microwave power transfer technique. Using this technique it is 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

$$2, \quad = \quad 2^{(1 + \quad + 2)}$$

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

$$, \quad = \quad 2^{(1 + \quad + 2)}$$

Equation 2: Downlink Transmission Rate

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

[13] Proposes a decentralised approach for mobile computational offloading. Decentralised 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.

[14]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.

$$(\,(\,)) \leq \quad (\,) \leq \quad (\,) \leq \min(\deg(\,)$$

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.

[15]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.

$$= (1 -) * \quad + \; *$$

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.

Next section describes the detail proposed methodology which enhance Green computing by lowering the CO2 level and reduces energy consumption also.

Proposed System

The proposed system consider parameters Load, Energy and scheduling on individual machines within cloud environment. The overall organization of proposed methodology is listed as follows

---

Algorithm EECOC
Initialization section

Set i=0,
Max=Maximum_VM,
Threshold=Capacity_of_vm,
MaxJ=maximum_no_jobs_in_queue

---

1. Input Jobs indicated through Cloudlet(Jobsi).
2. Assign Priority with each Job(Cloudlet). Maintain Jobs according to priority in P_Queue.
3. Specify Threshold value associated with each VMi. Threshold is the maximum limit of load associated with each VMi
4. Repeat while i<MaxJ 4.a
   Assign job to VMi
   4.b if Loadi>Threshold then
   4.c I=i+1(Migrate the Load to other VM in sequence)
   End of if
   4.d Load_on_vmi=Loadi
   End of Loop
5. Load on Individual Machine=Load_on_vmi
6. CO2_level=Consumed_Loadi
7. Storage_Utilizationi=S_used/S_Total

The proposed methodology is designed to reduce energy consumption and subsequently reduce CO2 levels. The proposed system is described in detail as under

A. Priority Job Queue

[16], [17]As jobs arrive within the system, priority is assigned to them. This priority in proposed system is allocated by identifying types of jobs. The jobs arriving within the system are divided into categories as

- System Processes3
- User Processes

System processes are generally given highest preferences and ser processes are give n least preferences. The priority is assigned on the basis of following factors

- Type of operation is identified and priority is assigned as operation number.
- Client computer performing the operation.
- The type of agent through which job is originated.
- Priority of client computer in references to job order.

After priority is assigned, jobs are grouped together within the queue.

$$[\quad , \quad] = \{ _1, _2, - - - - \quad \}$$

Equation 5: x represent jobs arranged in the form of a queue.

B. Assigning Threshold Load To Virtual Machines

Threshold load is assigned to VMs on the basis of capabilities of Host machines. Host machines if has configuration of 10GB RAM, 2000GHz processor and generated VMs are 10 then configuration of each VM is listed as follows.

VM	RAM	Processor
1	1 GB	200GHz
2	1GB	200GHz
	------	-----
------	------	-----
------	------	-----
-----	------	-----
-----	-----	-----
-----	-----	------
10	1GB	200 GHz

Table 1: Configuration of VMs

As the capabilities of VMs are same hence same threshold value can be allocated to each VM. Threshold value less than 1 GB(in terms of memory) and less than 200GHz(in terms of processing requirement) are assigned to each virtual machine.

C. Job Migration For Energy Efficiency

The capabilities of VMs are checked before allocation of job. In case, job requirements are more than threshold value, next VM in sequence is checked. This process continues until all the jobs are allocated and executed. The special variable 'expire' is introduced. This variable indicates VM capacity expires or not. If, VM capacity is not expired then job can continue on current machine. Since machine capability is not used beyond its capacity hence energy consumption is reduced considerably. After executing the jobs parameter calculation process begins. This is described as under

D.Parameter Evaluation

The job execution is followed by parameter evaluation. The parameter which how worth of study are

- Load
- Energy
- CPU utilization
- Storage Utilization

The CO2 emission greatly depends upon load on current machine. In order to calculate CO2 emission by individual machine, default value of CO2 emission per unit load is assumed for each machine. As the load on VM increases, CO2 levels increases. Hence CO2 greatly depends upon the load on current machine.

Load is evaluated by the amount of time job is executed on current machine. Let $= \{ _1, _2 - - - -, \}$ indicates the jobs and V=$\{ _1, _2, - - -, \}$ are the set of VM. Load on V1 is according to burst time of Jobs. J1 if has burst time of 50 and is executed on V1 than load on V1 is 50.

Consumed energy is calculated by the use of following formula

$$= * 100$$

Equation 6: Energy estimation mechanism

'Load' indicates the amount of load on current machine per unit time. It is given through following equation.

$$= \quad (\quad) * \quad (\;)$$

Equation 7: Load estimation equation

Poweri is predefined power associated with each VM. Load define burst time of job on current vm. Storage in terms of percentage is given as S(%)=(storagei/Total)*100

Equation 8: Storage in % form

Storage indicates job storage requirement and total indicates total job storage requirements. The CO2 emitted by VM is calculated using the equation.

em_factori=(em_factori*Total_Load)/100;

Equation 8: CO2 calculation em_factor indicates

the CO2 levels emitted by VM. Total_Load is the sum of burst time possessed by each job. Next section describes performance analysis and results.

Performance Analysis And Result

Comparison

Performance of EECOC is evaluated using the simulation conducted in CloudSim. Obtained results in terms of various parameters are listed as follows.

Table 2: Results in terms of Load and Storage.

Load is calculated using equation 7 and Storage (%)is calculated in terms of equation 8.

Virtual machine	Load	Storage(%)
VM 0	100	20.0
VM 1	99	19.8
VM 2	200	40.0
VM 3	110	21.0
VM 4	50	10.0
VM 5	80	15.5
VM 6	150	30.0
VM 7	40	8.0
VM 8	180	36.0
VM 9	140	28.0
VM 10	190	38.0

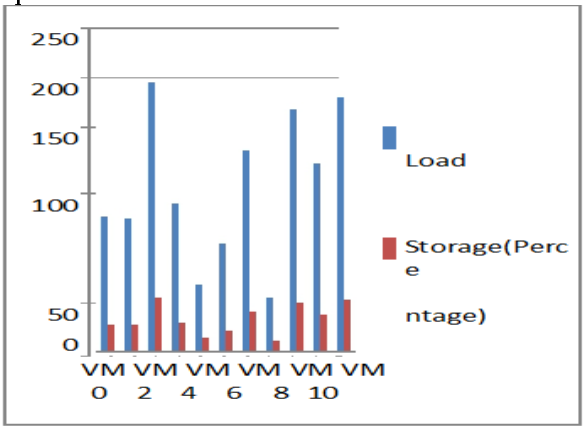


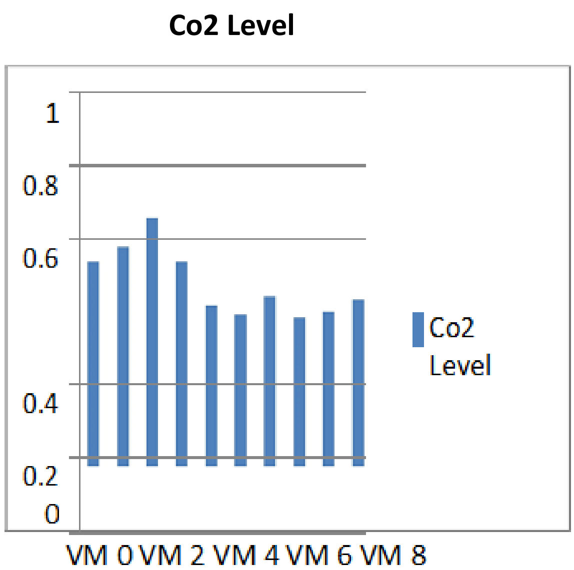
Figure 1: Plot of Load and storage present on VM.

The performance of EECOC is also evaluated in terms of CO2 levels. This is given as follows

Virtual machine	Co2 Level
VM 0	0.70
VM 3	0.70
VM 4	0.55
VM 5	0.52
VM 6	0.58
VM 7	0.51
VM 8	0.53
VM 9	0.57
VM 10	0.54

Table 3: CO2 levels associated with VMs in EECOC.

Plots corresponding to Table 3 are as follows



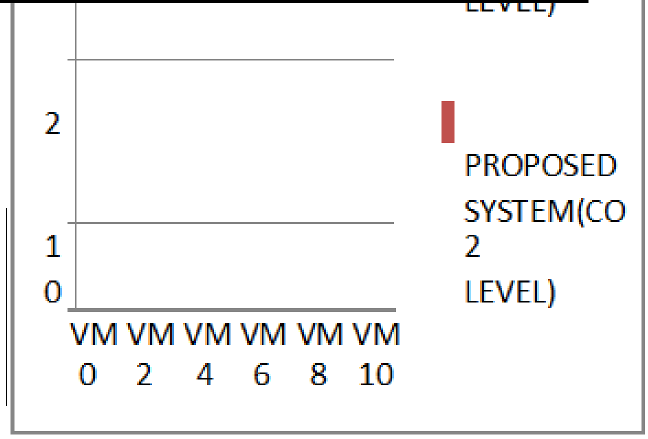


Figure 4: Comparison of EECOC with existing literature[18]

As indicated through obtained result, EECOC is performing better and result is improved by 33%.

VI. CONCLUSION AND FUTURE SCOPE

The EECOC reduces CO2 levels and enhancing performance. Job allocation strategy is formed by considering priority of the job submitted. Once job is submitted, VMs are analysed for deterioration. As the VM is deteriorating, load is migrated from current machine to next machine in sequence. Load is reduced considerably on current machine and load is balanced. Threshold value of CO2 emitted by distinct machines is maintained. As the load is assigned on individual machine, CO2 levels emitted by VM increases. However in EECOC these levels are minimal due to migration of load.

N the future, other allocation strategies like shortest jobs first and round robin can be used along with EECOC to determine mechanism to lower CO2 levels further.

References

[1] On and Y. For, "Introduction to the Special Section on Visual Computing in the Cloud : Fundamentals and Applications," vol. 25, no. 12, pp. 1885–1887, 2015.

[2] X. Cui, B. Mills, T. Znati, and R. Melhem, "Shadow replication: An energy-aware, fault-tolerant computational model for green cloud computing,"*Energies*, vol. 7, no. 8, pp. 5151–5176, 2014.

[3] V. Yadav, P. Malik, A. Kumar, and G. Sahoo, "Energy Efficient Data Center in Cloud Computing," *2015 IEEE Int. Conf. Cloud Comput. Emerg. Mark.*, pp. 59–67, 2015.

[4] Y. Nagafuchi, Y. Teramoto, B. Hu, T. Kishi, T. Koyama, and H. Kitazume, "Routing Optimization for Live VM Migration between Datacenters," vol. 0086, no. c, pp. 64–66, 2015.

[5] X. Dai, S. Member, J. M. Wang, and S. Member, "Energy-efficient virtual machines scheduling in multi-tenant data centers," vol. 7161, no. c, pp. 1–12, 2015.

[6] V. Devasvaran, N. M. A. Latiff, and N. N. N. A. Malik, "Energy efficient protocol in wireless sensor networks using mobile base station," in 2014 IEEE 2nd International Symposium on Telecommunication Technologies (ISTT), 2014, pp. 56–60.

[7] T. Kokilavani and D. I. G. Amalarethinam, "Reduced Makespan Task Scheduling Algorithm for Grid Computing," vol. 9, no. 27, pp. 71–76, 2016.

[8] V. Hamscher, U. Schwiegelshohn, A. Streit, R. Yahyapour, R. Buyya, and M. Baker, "Evaluation of Job-Scheduling Strategies for Grid Computing," Grid Comput., vol. 1971, pp. 191–202, 2000.

[9] C. Wu, J. Li, D. Xu, P.-C. Yew, J. Li, and Z. Wang,"FPS: A Fair-Progress Process Scheduling Policy on Shared- Memory Multiprocessors," IEEE Trans. Parallel Distrib. Syst., vol. 26, no. 2, pp. 444–454, Feb. 2015.

[10] C. You, K. Huang, and H. Chae, "Energy Efficient Mobile Cloud Computing Powered by Wireless Energy Transfer," vol. 8716, no. c, pp. 1–14, 2016.

[11] D. Marculescu, N. H. Zamora, P. Stanley-marbell, and R. Marculescu, "Fault-Tolerant Techniques for Ambient Intelligent Distributed Systems-," pp. 348– 355, 2003.

[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, 2016.

[13] X. Chen, "Decentralized Computation Offloading Game for Mobile Cloud Computing," IEEE Trans Parallel Distrib. Syst., vol. 26, no. 4, pp. 974–983, Apr. 2015.

[14] 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., 2013.

[15] 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, 2011.

[16] B. Jiang, J. Wu, X. Zhu, and D. Hu, "Priority-Based Live Migration of Virtual Machine," pp. 376–385, 2013.

[17] 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.

[18] U. Wajid, C. Cappiello, P. Plebani, B. Pernici, N Mehandjiev, M. Vitali, M. Gienger, K. Kavoussanakis, D. Margery, D. G. Perez, and P. Sampaio, "On Achieving Energy Efficiency and Reducing CO 2 Footprint in Cloud Computing," vol. 7161, no. c, 2015.