Smart Workflow Scheduling using the Hybridization of Random Weight Model with Ant Colony Optimization (RWM-ACO)

Deepak Sharma
Research Scholar,
Indo Global College of Engineering,
Abhipur, Mohali, India.

Talwinder Kaur
Asst. Professor,
Indo Global College of Engineering,
Abhipur, Mohali, India.

ABSTRACT
The cloud based platforms are designed specifically for the provision of the high performance clusters (HPC), which is realized by using the multiple techniques all together for the realization of the distributed computing environment. The cloud platforms are designed to handle the independent queries either in the groups or individually for the minimization or optimization of the response time for the rich user experience. For this, the cloud models utilize the versatile task scheduling models, which are based upon the various types of parameter either in individuality or aggregate. In this paper, the random weight based calculation for the scheduling of the tasks over the target cloud systems, which is further channelized using the ant colony optimization (ACO) based swarm intelligence. The performance of the ACO with random weights based algorithm based upon the response time and energy consumption on a primary note. The proposed model has been found efficient in the terms of the obtained performance parameters.

Keywords
Swarm intelligence, random weight computation, cloud task scheduling, workflow management.

1. INTRODUCTION
In this paper, the model based upon the task scheduling has been proposed using the ant colony optimization (ACO) algorithm along with the random weight calculator for the flexible task scheduling over the target cloud based platform. The proposed model is aimed at solving the problem of the scheduling in the cloud platforms using the CloudSim simulator. The process sequencing is a scheme of put the runtime processes in the memory in the perfect placement or sequence in order to minimize the total tasking and communication overhead in terms of time and load respectively. The task scheduling is the major part of the cloud architectures to upgrade the achievement of the cloud platform. An idle process sequencing algorithm should be aimed at reducing the overall tasking overhead, tasking time (task completion time) and communication overhead by the whole task incoming and outgoing information. The task management faces the major challenges from the bias-free dynamic resource allocation while keeping the cloud performance on the maximum in terms of execution time and computational overhead. The proposed model is balanced task scheduling algorithm with the intelligent algorithm of ant colony optimization (ACO) for the high-end fault tolerance to reduce the failure rate among the cloud platform.

2. LITERATURE SURVEY
A. Baal and I. Chana [9] proposed that fault resistance is a considerable interest to warranty availability and accuracy of central utilities as well as operations performance. Although to minimal the breakdown crash on the scheme and operationsperformance, losses should be estimated and primitively organized. Xu et. al [14] presents a load offset representation for the public cloud that relies upon the cloud partitioning idea by a switch scheme to choose distinct methods for various conditions. Liu, Zhanghui, and X.Wang [15] **the** author introduced an advanced process sequencing method in this work. In the proposed model, author enhances the process completion time to analysis the process running time and the assets usage. K. Li et.al [16] defined that the cloud is the growth of parallel computing. Task scheduling is one of the underlying issues in this environment. Various meta-inquisitive algorithms have been designed to resolve cloud process sequencing. An efficient process scheduler should work its sequencing scheme to the different surroundings and the varieties of processes. Chang, Haihua, and X.Tang [17] author proposed a method for assests scheduling beyond the cloud environment. In this paper, depends on dynamic load balance the author has proposed a resource-scheduling algorithm. S.Cavic, Vesna, and E. Kuhn [18] author discussed in this paper about the advanced methods that develop speedily and their complication is a crucial concern. The promising way to cope with improved intricacy is to enlist a self- coordinating strategy. The various distinct strategies that deal the workload offset problem but most of issues are process oriented and however, it is hard to differentiate. A.Jain and R. Singh [19] described Grid computing is allocation of non-identical resources. In these days, large amount of the resource management peer to peer grid environment is used. Load offset is crucial concern to offset the allover workload of the knobs in cloud.

3. EXPERIMENTAL DESIGN
3.1 Normal or Body Text
The subdivision of the tasks is based on the length of the task. A task is usually divided in the t slots, where t is smallest time unit available for the task length calculation in our proposed model. A task smaller than or equal to t will be processed in one round, where the tasks larger than t can be scheduled in queue or on different VMs according to the load and time calculation for the faster processing. The arbitrary proportional rule is applied to recognize the ratio of processes in processing on the given resource has been presented in the following equations:

$$P_1 = (R_1 + K)^k / (R_1 + K)^k + (R_2 + K)^k \qquad (6)$$

$$B_1 = P_1 * TR_i \qquad (7)$$

Where A_1 is the number of tasks assigned on the resource A, P_1 is the probability of the resource, R_1 is the pheromone value based on the available ratio of RAM and CPU on VM under consideration, Tr_i depicts the resource availability required to process task i. The k and h are the coefficients used for the choice of probability among the available resources for the sequencing of the processes among the accessible resources. The value of k and h is calculated on the basis of the VM load and resource availability on all of the available VMs. The variation in the values of k and h will define the variability on the basis of the current processing load on the different VMs, which inspires the task assignment decision of the ACO algorithm. The used rule for the probability calculation has been given in the following equation:

$$P_j = (R_i + K)^k \,/\, \sum_{i=1}^{n} (R_i + K)^k \qquad (8)$$

In the proposed work, the Meta tasks are used for the testing of the proposed model. The Meta tasks does not carry any dependency on the other tasks in the processing queue, which means the response time will be calculated for the each individual task by evaluating the variation between the finish time and start time. The waiting time is also considered as the response time delay, which is caused due to the waiting period spent in the queue.

Algorithm 1: Smart Task Scheduling using Ant Colony Optimization

1. Obtain the task information from the user's end
2. Obtain the Virtual Machine (VM) information from the target cloud platform, VM
3. Obtain the load information on each of the VM in the list on step (2), VM_load
4. Begin the iteration equal to the number of tasks on step (1) with iterator i
   a. Compute the load on each of the target VM, tVM ← VM_load < Thr, where Thr is the threshold for the current load
   b. Compute the percentage of the load over each of the VM and update the tVM
   c. Find the adaptively free Virtual Machines (VM) with processing capability to the target task information, denoted cVM ← tVM > Process_Length(i)
   d. Compute the compatility of the task(i) with the VMs registered with the cVM, hcVM ← cVM ~= Process_Length
   e. Assign the task(i) to the virtual machine on the top of the list in the hcVM container
   f. Update the new task scheduling entry to the scheduling management matrix (SMM)
   g. Update the VMs load and scheduling information in the form of pheromone value computed by the ACO algorithm for the target VM and task
   h. If its not the last task
      i. Go the step 4(a)
   i. Else
      i. Return the task scheduling matrix to the analytical algorithm
      ii. Return the simulation

4. RESULT ANALYSIS

The proposed model simulation has been prepared by using the Cloud Simulator for the task scheduling procedure testing over the virtual cloud environment. A scenario of multiple user online source has been assumed in this simulation, where the request are being received from the multiple users on same time as per it happens in the social networks like Facebook, Twitter or other online giants such as Google, Amazon, etc. The VM regions has been defined according to the failure rate and virtual machine load status which is transformed into the threshold value using the mathematical equations. Entire simulation is based on the single Time zone scenario, where all users in the given user base are projected as residents of one country or time zone. The simulation can be easily considered for the testing in the peak hours for the point of task scheduling of the samples tasks over the given bunch of resources. The task density has been assumed to be overwhelming during the peak hours, which delays the request response by adding the scheduling delay or processing delay. Hence, the task scheduling method should be enough faster to reduce the task scheduling delay and assigned resources should be enough vigorous to process the task as fast as possible to minimize the processing delay.

Table 4.1:Simulation Calculation of Proposed model.

Random Job ID	Status	Data center ID	VM ID	EF Time	Start time	Finish time	Depth	Energy
1	SUCCESS	2	16	13.11	0.31	13.42	1	7.7348995
9	SUCCESS	2	5	13.7	0.31	14.01	1	8.082999
19	SUCCESS	2	4	10.7	34.9	45.6	2	6.3129997
29	SUCCESS	2	19	10.82	13.73	24.55	2	6.3837996
39(1)	SUCCESS	2	14	10.6	13.73	24.33	2	6.2539997
39 (2)	SUCCESS	2	14	10.6	13.73	24.33	2	6.2539997
49	SUCCESS	2	11	10.7	35.12	45.82	2	6.3129997
59	SUCCESS	2	10	10.8	13.86	24.66	2	6.3719997
69	SUCCESS	2	13	10.66	34.7	45.36	2	6.2893996

79	SUCCESS	2	2	5.34	64.36	69.7	4	3.1506
89	SUCCESS	2	19	10.87	69.7	80.57	5	6.41329
99	SUCCESS	2	2	0.83	104.8	105.63	9	0.48969

4.1. Response Time

The response time describes the total time taken for the cloud platform to generate the reply to the user's request.

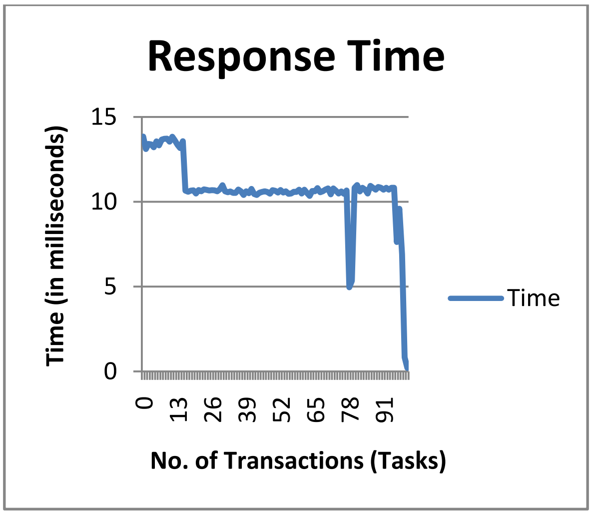

Figure 4.2: Response time per task or transaction

The cloud platforms are expected to receive the larger number of users to successfully run the installed application under the high performance clusters (HPC). The proposed model has been evaluated for the results obtained from the simulation for the response time, which has been represented graphically in the following figure. The overall response time of process 100 is 0.1 seconds. The response time for process 1 is 13.11 seconds and for process 99 the response time that is 0.83 seconds. The process 8 takes 6.91 seconds to complete the task.

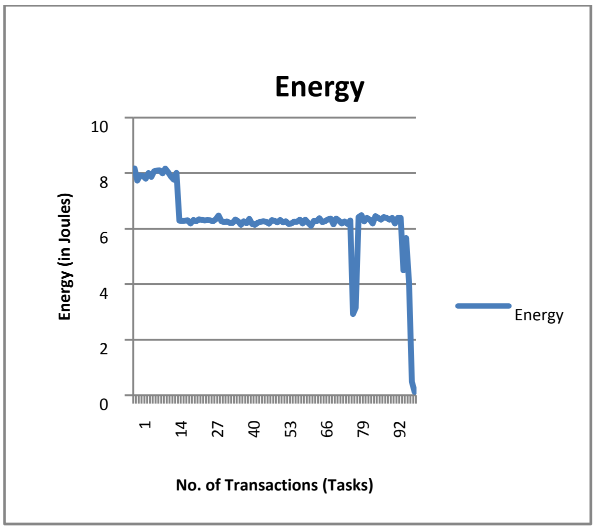

Figure 4.3: Energy consumption per task or transaction

4.2. Energy Consumption

The energy consumption has been also monitored for each of the task in order to assess the overall energy consumption during the simulation. The proposed model has been designed to schedule to task over the virtual machine with the minimum energy and cost indices. The following figure shows the overall energy consumption for all of the tasks in the simulation. The process 100 consumes 0.12389999 joules of energy to complete the execution. The process 99 consumes 0.4897 joules of energy to complete the execution. The following figure shows the energy consumed by every individual process to complete the execution in the proposed system.

5. ACKNOWLEDGMENTS

The proposed model is based upon the ant colony optimization (ACO) based smart and balanced task scheduling model. The proposed model has been designed with the active random weight computation in order to schedule the task data over the target cloud model with the assisted computing for the ACO based scheduling engine. The proposed model's performance has been analyzed using the performance parameters of energy consumption over the cloud cluster or individual virtual machine (IVM) and overall response time (response delay). The proposed model has been recorded with slihgly higher value than 6 joules on an average for the scheduling of the 100 tasks in the given simulation, which represents the robust performance by the proposed model in handling the heavy computational enabled workflows (specifically scientific workflows). In the proposed model simulation, the proposed model has been recorded with the nearly 10 milli-seconds for the scheduling of each of the task, which similarly represents the flexibility and robustness of the proposed model in handling the scientific workflows consisting of nearly 100 tasks, which are also further subdivided in the sub-tasks as per shown for the task 39 in the table 4.1.

6. REFERENCES

[1] "Gartner Highlights Five Attributes of Cloud Computing" [on-line] available from: http://www.gartner.com/newsroom/id/1035013 [0n13 Dec 2013].

[2] Z. Tari, "Security and Privacy in Cloud Computing", IEEE Cloud Computing, vol. 1, no. 1, pp. 54-57, 2014.

[3] Y. Xianfeng and L. Tao, "Load Balancing of Virtual Machines in Cloud Computing Environment Using Improved Ant Colony Algorithm", International Journal of Grid and Distributed Computing, vol. 8, no. 6, pp. 19-30, 2012.

[4] L.M Vaquero., L. R Merino., J. Caceres and M. Lindner "A break in the clouds: towards a cloud definition," ACM SIGCOMM Computer Communication Review, vol.39, issue 1, pp. 50-55, 2008.

[5] Basic concept and terminology of cloud computing-[on-line] available from: http://whatiscloud.com [on 9 Jan 2015].

[6] Q. Zhao, C. Xiong, C. Yu, C. Zhang and X. Zhao, "A new energy-aware task scheduling method for data-intensive applications in the cloud", Journal of Network and Computer Applications, vol. 59, pp. 14-27, 2016.

[7] Anthony T.Velte, Toby J.Velte, Robert Elsenpeter, "Cloud Computing - A Practical Approach," Tata McGraw-Hill Edition (ISBN-13:978-0-07-068351-8).

[8] S. Bharti and K. Pattanaik, "Dynamic Distributed Flow Scheduling with Load Balancing for Data Center Networks", Procedia Computer Science, vol. 19, pp. 124-130, 2013.

[9] Y. M., "A Survey of Cloud Computing Fault Tolerance: Techniques and Implementation", International Journal of Computer Applications, vol. 138, no. 13, pp. 34-38, 2016.

[10] W. Zhao, P. M. Melliar Smith, and L.E. Moser, "Fault Tolerance Middleware for Cloud Computing," Proceedings of IEEE 3rd International Conference on Cloud Computing, USA pp 67-74, 2010.

[11] Dorigo M, Optimization, learning and natural algorithms. PhD thesis, Dipartimento di `Elettronica, Politecnico di Milano, Italy, 1992 [in Italian].

[12] M Rahman, S. Iqbal, and J.Gao "Load Balancer as a service in Cloud Computing," IEEE 8th International Symposium on Service Oriented System Engineering, pp.204-211, April 2014.

[13] "Cloud Computing: A delicate balance of risk and benefit" [0n-line] available from: http://www.zdnet.com/blog/hinchcliffe/eight-ways-that-cloud-computing-will-change- business/488 [10 Feb 2015].

[14] S. Choi, "Fault-tolerance Scheduling for Service Sharing in Mobile Social Cloud Computing", Korea Institute of Information Technology Review, vol. 11, no. 1, 2013.

[15] Gaochao Xu, Junjie Pang, and Xiaodong Fu "A Load Balancing Model Based on Cloud Partitioning for the Public Cloud," IEEE Transactions in Tsinghua Science and Technology, vol -18, issue 1, pp. 34-39, 2013.

[16] Liu, Zhanghui, and X.Wang. "A PSO-based algorithm for load balancing in virtual machines of cloud computing environment," Advances in Swarm Intelligence, Springer Berlin, pp.142-147, 2012.

[17] K.Li, G.Xu, G. Zhao, Y. Dong, and D. Wang " Cloud Task scheduling based on Load Balancing Ant Colony Optimization," 6th **IEEE** Annual China Grid Conference , pp 3-9, 2011.

[18] Chang, Haihua, and X.Tang. "A load-balance based resource-scheduling algorithm under cloud computing environment," 9th International Conference on Web-based Learning**,** Springer, Berlin, pp. 85–90, 2011.

[19] S.Cavić, Vesna, and E. Kühn, "Self-Organized Load Balancing through Swarm Intelligence," In Next Generation Data Technologies for Collective Computational Intelligence, Springer, Berlin pp. 195-224, 2011.

[20] A. Jain and R. Singh, "An Innovative Approach of Ant Colony Optimization for Load Balancing Peer to Peer Grid Environment," IEEE International Conference of Issues and Challenges in Intelligent Computing Techniques , pp. 1-5, 2014.

[21] R. Chaukwale and S.S. Kamath, "A Modified Ant Colony Optimization Algorithm with Load balancing for Job Shop Scheduling," 15th IEEE International Conference onAdvanced Computing Technologies , pp. 1–5, 2013.

[22] S. Dam, G. Mandal, K. Dasgupta, and P. Dutta "An Ant Colony Based Load Balancing Strategy in Cloud Computing," In Advanced Computing, Networking and Informatics Springer, vol 2, pp. 403-413, 2014.

[23] S. K Goyal and M. Singh "Adaptive and Dynamic Load Balancing in Grid Using Ant Colony Optimization," International Journal of Engineering and Technology, vol. 4, issue 4, pp.167-174, 2012.