Review of Various Load Balancing and Green Computing Techniques in Cloud

Article *in* Journal of Basic and Applied Engineering Research · January 2015

Journal of Basic and Applied Engineering Research
Print ISSN: 2350-0077; Online ISSN: 2350-0255; Volume 2, Number 2; January-March, 2015, pp. 122-127

http://www.krishisanskriti.org/jbaer.html

Review of Various Load Balancing and Green Computing Techniques in Cloud

Ritu Kapur[1] and Maitreyee Dutta[2]

[1]*ME Student, Computer Science Department NITTTR, Sector 26 Chandigarh (India)*
[2]*Professor and Head of Department Computer Science Department NITTTR, Sector 26, Chandigarh (India)*
E-mail: [1]ritukapur1591@gmail.com, [2]d_maitreyee@yahoo.co.in

Abstract—*With huge amount of energy being wasted in the form of heat during the processing of various tasks on cloud, Green Computing has become the need of hour. Huge amount of energy is wasted at various Data Centres and Data Servers in Cloud. Techniques like Load Balancing, Server Consolidation, VM Migration etc. can be used to employ a more sustainable use of energy. Algorithms employing these techniques proposed by different authors are presented in this paper. Various Green Computing Techniques like air cooling etc. and Various Meta-Heuristic Algorithms performing Load Balancing like Genetic Algorithms, Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) are also discussed in this paper. A greener use of energy will lead to a better society in the long run.*

Keywords: *Green Computing, Cloud, Load Balancing, Server Consolidation, VM Migration, Meta-Heuristic Techniques*

1. INTRODUCTION

Cloud Computing has emerged as a popular area of research and development in the past few years. Although no unique definition of Cloud Computing exists many standard organizations continue to give various definitions for it [1]. According to National Institute of Standards and Technology (NIST) definition in September 2011 in [2], Cloud computing is defined as "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction". Cloud computing as utility provides pay-per use and pay-as-you-go service [3] i.e. a client/user has to only pay for the amount of services used by him [4].

According to [5] and [6], International Standards Organization (ISO) on 24th October 2014 defined cloud computing as "an evolving paradigm" and identified and described its "key characteristics" including broad network access, measured service, multi-tenancy, on-demand self-service, rapid elasticity and scalability, and resource pooling. Cloud computing basically provides three service models: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) as shown in [2] and [7]. According to [8], [9] and [3], IaaS provides the access to various resources which provide processing, storage, networks, operating systems and various other capabilities of scaling up and down the infrastructure dynamically. The services or resources required by a particular user are listed in the Service Level Agreement (SLA) established as a contract between the cloud service provider and the cloud consumer respectively [10].

According to D. Linthium in [11], Amazon is providing lowest priced instances and thus most of the researchers in [14-17] have deployed Amazon Cloud in [12] and Azure Cloud in [13] for their simulations. CloudSim, being an open source tool can also be used for simulations as used in [18]. The detailed implementation and basic design of CloudSim is explained in [19].

International Federation of Green ICT and IFG Standard [20] define Green computing or Green IT or ICT sustainability as the study and practice of environmentally sustainable computing or IT. According to S. Murugesan [21, 22], Green Computing can include "designing, manufacturing, using and disposing of computers, servers and associated subsystems such as monitors, printers, storage devices and networking and communicational systems – efficiently and effectively with minimal or no impact on the environment." S. K. Garg et al. [23] describe the various components of cloud and explain the manner in which they consume energy.

Data Centers store large amount of data in cloud. Huge amount of energy is wasted in the form of heat on cloud. Cloud thus has lately become one of the major sources to global warming [22]. Also, one of the major causes of climatic change is Electricity. This is because, many global warming gases like carbon dioxide, carbon monoxide etc. are released during the electricity generation process [22]. According to Department of Energy (DOE) report [24], Data Centers consumed 1.5% of all electricity in the U.S. in 2006 and their power demand is growing at the rate of 12% an year.

Microsoft, Google and Yahoo are doing efforts to reduce this energy wastage at their end. Microsoft has thought of shifting its electricity requirement to renewable sources of energy by established some of its data centers in Washington now running by hydroelectricity generated by dams and some to Ireland benefitting by its moderate climate and using the air cooling technique thereby reducing the energy losses by 50%. Google employs customized evaporating cooling to do the same energy saving process. Also, Google and Yahoo have gone carbon-neutral since 2007. Also many global consortiums have been organized to continue this Green IT movement.

R. Stephen [25] discusses the money spent on the power and cooling of various Data Centers and Data Servers on the Cloud. The author also discusses the concept of thin clients and virtualization software which help in reduction of electricity consumption. Thin clients reduce electricity consumption by minimizing the computations and the virtualization software helps in reducing the idle time of servers by performing the load balancing and thus also reducing the total response time and finally reducing the total time for which the servers have to run to execute the same tasks in the virtualized environment.

N. J. Kansal et al. [26] presents a Green Computing Architecture and various existing Load Balancing Techniques and a comparison between them on the basis of various QoS parameters.

Authors in [30, 31] show the mechanism of server consolidation. In this mechanism, the load from the minimal utilized servers is migrated to other servers, thus making some of the servers free and making them shut down thereby saving the power used by them.

R. Yanggratoke et al. [32] propose a Cloud Computing Architecture illustrated in figure 1, to represent a brief idea about the architecture of cloud and the role of middleware which is further explained in detail and illustrated in figure 2. The authors in [32] also propose an algorithm which reduces power consumption by doing an efficient resource scheduling.

T. Forell et al. [33] propose Open Cirrus Testbed comprising of various components which prove to provide most sustainable use of various resources and power consumption. The authors in [33, 34] also discuss about the concept of sustainability in cloud. A Cloud Sustainability Dashboard is also embedded in the Open Cirrus Testbed which accounts for the consumption of power, CO_2, water and other resources and shows the ecological impact to the user. The component also provides the alerts in case of any significant change in any sustainability component.

R. Santhosh et al. [35] propose a Pre-emptive Scheduling Algorithm to schedule resources on the basis of pre-emptive priority. In case a higher priority task arrives it can preempt the currently executing task. In order to save the processing done so far on the currently executing task, the task is migrated on other Virtual Machine and the processing on the new virtual machine resumes from the same point.

The authors in [36-42] discuss the concept of various meta-heuristic algorithms like Genetic Algorithms (GAs), Ant Colony Optimization (ACO) algorithms, Particle Swarm Optimization (PSO) algorithms etc. The authors in [36-39] discuss various genetic algorithms. GA provide a solution to complex combinational optimization problems [43], which come under the category of NP-Hard problems and thus their solution is hard to find.

The authors in [40-47] discuss various Ant Colony Optimization (ACO) algorithms. K. Li et al. [40] presents a Load Balanced version of basic ACO proposed by M. Dorigo et al. [41-42]. The authors in [40] also give a load balancing factor for the same. The authors in [48] describe an optimized version of basic PSO algorithm shown by [49].

2. ARCHITECTURE

2.1 Cloud Computing Architecture [32]

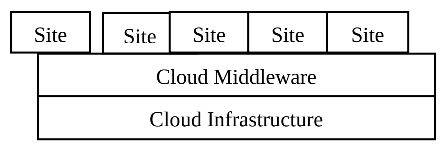

Fig. 1 Cloud Computing Architecture [32]

Cloud Service Providers or Site owners provide various services to the Cloud Consumers through the Middleware Layer from the actually executing Infrastructure Layer below which has the actual hardware resources like processors etc.

2.2 Resource Allocation by Middleware [32]

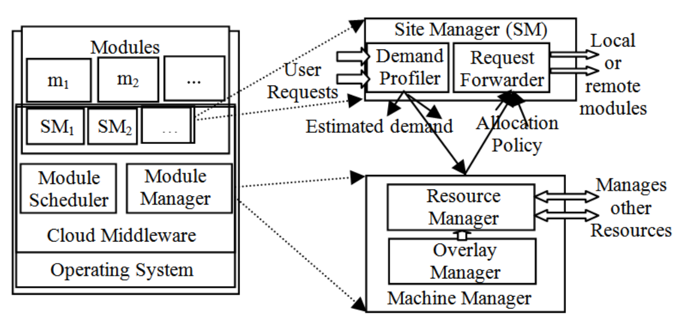

Fig. 2: Middleware Resource Allocation Architecture [32]

Middleware is the interfacing layer between the operating system and the other modules. Resource scheduling and

Resource managing is done by the module scheduler and the module manager component of the middleware, respectively. The Site Manager (SM) calculates the demand of various resources based on the user requests and these requests are then scheduled onto various resources by the Allocation Policy defined by the Machine Manager.

2.3 Green Computing Framework [25]

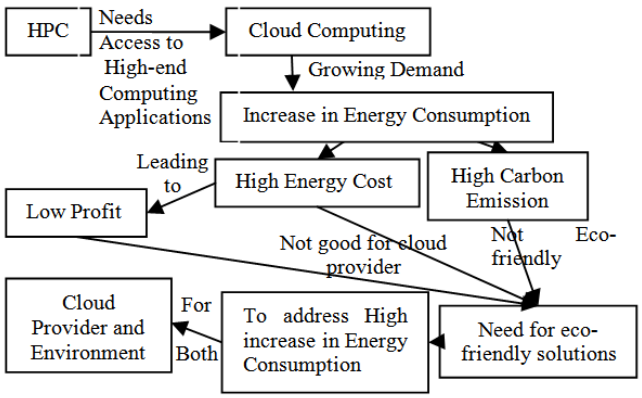

Fig. 3: Green Computing Architecture [25]

HPC i.e. High performance computing refers to any computational activity requiring more than a single computer to execute a task [26]. As shown by the scenario in figure 1, HPC users need access to high-end computing capabilities which is fulfilled by Cloud computing by providing resources as desired and when desired i.e. in the on-demand scenario to the users. But the growing demand of cloud computing has lead to an increase in energy consumption [27 - 29] by the various resources in cloud thus leading to a High Energy Cost and a High Carbon Emission. High Carbon Emission is not eco-friendly and therefore comes the need for some new eco-friendly energy efficient alternatives. High energy Cost leads to decrease in profit and thus is not desirable. New Energy Efficient solutions are needed both by the Cloud Provider and the Environment.

3. ALGORITHMS

3.1 Genetic Algorithm [36, 53]

Jingyi Ma [36] proposes a genetic algorithm named as a heuristic genetic load balancing algorithm (HGLBA) which helps in reducing the overall response time of the tasks through its load balancing phenomenon. The algorithm executes in the following manner:-

a) **Generate:** The first step is to generate the initial population, P(0). This initial population is basically some initial solutions coded in the form of chromosomes. The maximum iteration named as *MaxSize* is initialized to zero i.e. k=0;

b) **Evaluate:** In the evaluation step, the Fitness value of the Initial Population, P(k), is evaluated on the basis of some function. e.g. the evaluation function in [44] is shown below in equation 1.

$$\mathbf{T_j} = \sum_{j=1}^{m} \left(w_{ij} * Q_{ij} \right) \text{ for } \varphi = 1, 2, \ldots, n \quad (1)$$

where w_{ij} represents the weight listed to various QoS parameters (according to the user preferences), Q_{ij} represents the expected values of QoS parameters, T_j is the fitness function of the task j, m represents the total no. of parameters and n represents the total no. of resource types.

A) **Improve:** In this improve step, various operations like mutation and crossover etc. are applied in order to generate the next generation, P(k+1).

B) **Termination:** If the iteration k>MaxSize, we take the best solution and terminate the process otherwise we increment k, i.e. k=k+1, and continue the process from step B.

As this algorithm provides optimal allocation of resources, it serves the aim of the load balancing i.e. the minimization of overall execution time. The authors in [53] present a variation of genetic algorithm in an unreliable cloud scenario.

Ant Colony Optimization (ACO) Algorithm [40-47]

M. Dorigo et al. [44, 45] present the basic ACO algorithm. K. Li et al. [40] embeds the load balancing parameter into it and thus presents a Load Balanced ACO i.e. LBACO algorithm. According to the authors in [40- 45] ants have a general property of locating their path to food. In this phenomenon of locating their food, ants keep on depositing a chemical substance known as pheromone. This pheromone in turn helps new ants on their path to find the shortest route to the food. The path having largest amount of pheromone represents the shortest route. So this process thus leads the new ants to their location in minimum time through the shortest path.

The LBACO follows the following steps:

A) **Initialization:** The ants i.e. resources are randomly distributed onto various VMs initially. The pheromone is initialized as follows:

$$\tau_j(0) = pe\_num_j * pe\_mips_j + vm\_bw_j \quad (2)$$

where pe_num, pe_mips_j and vm_bw_j represent the no. of available VM processors, MIPS (Million Instructions Per Second) and the bandwidth capacity of the processor VM_j respectively.

B. Choosing the next task

Every k^{th} ant chooses the next task with the probability $p_j^k(t)$ defined as:

$$p_j^k(t)= \frac{(\tau_j)^\alpha *(EV_j)^\beta *(LB_j)^\gamma}{\Sigma(\tau_j)^\alpha *(EV_j)^\beta *(LB_j)^\gamma} \text{ if } j \ \varepsilon\ 1 \dots n \quad (3)$$

else $p_j^k(t)= 0$

where $\tau_j(t)$ represents the pheromone value at time t, EV_j represents the computing capacity of j^{th} VM defined as follows:

$$EV^j= pe\_num_j*pe\_mips_j+vm\_bw_j \quad (4)$$

LB_j represents the Load balancing factor for the j^{th} VM represented as follows:

$$LB_j= 1-\frac{res_j - \text{lastAver\_res}}{res_j + \text{lastAver\_res}} \quad (5)$$

where lastAver_res represents the average execution time of all the VMs and res_j represents the current expected execution time, defined as follows:

$$res_j= \frac{\text{total\_tasklength}}{EV_j} + \frac{\text{InputFilesize}}{vm\_bw_j} \quad (6)$$

where total_tasklength is the total length of all the tasks submitted to the VM_j and InputFilesize is the length of the task before execution and α,β,γ represent the parameters to control the relative weight of pheromone laid, the computing capacity and the load balancing factor of various VMS.

C. Pheromone Updating

$$\tau_j(t+1) = (1-\rho)^* \tau_j(t) + \Delta\tau_j \quad (7)$$

where ρ represents the pheromone decay coefficient,

$$\Delta\tau_j = \frac{1}{T_{ik}} \quad (8)$$

where T_{ik} represents the shortest path searched by k^{th} ant in the i^{th} iteration and once an optimal solution is found, the global pheromone updating is done as follows:

$$\Delta\tau_j = \frac{D}{T_{op}} \quad (9)$$

where D represents the encouragement coefficient and T_{op} represents the found optimal solution.

D. Terminating Step

If all the ants end their search i.e. the solution converges to an optimal value, current optimal solution is saved and the value of makespan variable, N_c, is incremented. If this makespan value proves to be the best value, the process ends else the ants continue to search for a better optimal solution.

Authors in [46, 47] give a variation of ACO algorithm in which the workload is distributed uniformly among all the VMs.

Particle Swarm Optimization (PSO) Algorithm [48-52]

Particle Swarm Optimization (PSO) [52] is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995 inspired by social behavior of bird flocking or fish schooling. PSO is similar to GA in the sense that the system is initialized with a population of random solutions and searches for optimal solution by updating generations. However, as compared to GA, PSO does not perform any crossover and mutation operations. In PSO, the potential solutions, called particles fly through the problem space by following the current optimum particles. Each particle keeps track of its coordinates in the problem space, which are associated with the best solution (fitness), it has achieved so far. The fitness value stored is called pbest. Another best value found by the PSO optimizer by examining the particle and its neighbors is lbest. The global best value found by examining the global population is gbest.

The PSO works by varying the velocity per unit time, i.e. acceleration, of each particle towards its pbest and lbest locations. The change known as acceleration is brought about by generating random numbers by the random number generation method. The two basic equations used in PSO for calculation purpose are:

$$V_{i+1}=w^*V_i + c_1r_1(X_{lb}-X_i) + c_2r_2(X_{gb}-X_i) \quad (10)$$

$$X_{i+1}=X_i + V_{i+1} \quad (11)$$

where V_{i+1} is new calculated velocity, w is inertia weight, V_i is current velocity of particle, c_1 is first acceleration coefficient , r_1 is first random variable, X_{lb} is lbest, X_i is current position, c_2 is second acceleration coefficient, r_2 is second random variable, X_{gb} is gbest, X_{i+1} is next calculated position value of variable.

The algorithm proposed by [48] has the following steps:

A) **Initialize:** The population of particles, i.e. tasks, is initialized by initializing candidate solution, Inertia weight, pbest, lbest, bgest and other PSO variables.
B) **Generate:** Particle mapping are generated with machines using smallest position value heuristic.
C) **Evaluate:** Next, the fitness parameter is evaluated by using the make span parameter. If the current candidate solution has best fitness value, the new mapping is updated with its value; else the previous candidate solution is retailed.
D) **Update:** On the basis of the previous evaluate step, pbest and gbest value is evaluated

E) Calculate: New velocity value and new particle position is calculated based on the equation 10 and 11.

F) Termination: If the maximum no. of iterations are reached, candidate mapping's expected time to complete matrix is calculated and the control is passed onto the load balancing component else the process iterates to step B.

G) Load Balancing: Next we check if load balancing is required or not. If it is required, an appropriate max-max or min-min algorithm concept is chosen for shuffling the task mappings. Tasks are shuffled and reordered and mapping is reevaluated and the load balancing is rechecked. If the new mapping is found to be load balanced the process is terminated and the final task machine is ready for execution. If the load is not found to be balanced, the tasks are reshuffled and the process continues till a perfectly load balanced mapping is found.

The simulations by the authors in [48] prove that the proposed algorithm outperforms the basic algorithm.

4. CONCLUSION

The paper starts by introducing the concept of cloud computing, green computing and then the various techniques by which IT is moving towards Green IT and finally the optimized load balanced resources allocation techniques in the form of meta-heuristic techniques like GA, ACO, LBACO, PSO etc. achieving the aim of Green IT of reducing the power consumption by reducing the time of execution of same processes. Thus I conclude that the principles of Green IT can be combined with the load balanced resource allocation in order to achieve a more optimized performance and a more sustainable use of resources at the same time.

REFERENCES

[1] D. Linthincum (2014), "It's pointless to define cloud computing", http://www.infoworld.com

[2] "Cloud Computing Bible", by Barrie Sosinsky, Chapter 2

[3] A. Aggarwal and S. Jain, "Efficient Optimal Algorithm and Task Scheduling in Cloud Computing Environment" , International Journalal of Computer Trends and Technology (IJCTT), vol. 9, pp. 344-349, Mar 2014

[4] E. D. Jensen, C. D. Locke, and H. Toluca (1985), A time-driven scheduling model for real-time systems, In IEEE Real-Time Systems Symposium.

[5] J. Borne (20th October 2014), "ISO publishes new cloud computing standards and definitions", www.cloudcomputing-news.net

[6] T. Roughton (17th October 2014), "Cloud Computing terms defined in new ISO standard", www.out-law.com

[7] "Cloud Computing, A Practical Approach" by Anthony T. Velte, Toby J. Velte and Robert Elsenpeter, 2010 edition, Chapter 2.

[8] "Real time task scheduling", last chapter, Version 2, IIT Kharagpur.

[9] F. Casati and M. Shan (2001)., Definition, execution, analysis and optimization of composite e-service, IEEE Data Engineering.

[10] R Buyya, C. S. Yeo and Venugopal, "Market oriented cloud computing: Vision, hype, and reality for delivering IT services as computing utilities" in *Proc. of the 10th IEEE International Conference on the High Performance Computing and Communications*, 2008.

[11] D. Linthincum, "Faster and Cheaper: The cloud is becoming harder to resist", http://www.infoworld.com

[12] Amazon elastic compute cloud [Online]. Available: http://aws.amazon.com/ec2

[13] Microsoft windows azure. [Online]. Available: http://code.google.com/windowsazure/

[14] X. Nan, Y. He, and L. Guan, "Optimization of Workload Scheduling for Multimedia Cloud Computing," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–4, 2013.

[15] X. Nan, Y. He, and L. Guan, "Towards Optimal Resource Allocation For Diffentiated Multimedia Services in Cloud Computing Environment," in *Proc. IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, pp. 1–5, 2014.

[16] X. Nan, Y. He, and L. Guan, "Optimal resource allocation for multimedia cloud based on queuing model," in *Proc. IEEE International Workshop on Multimedia Signal Processing (MMSP)*, pp. 1–6, 2011.

[17] X. Nan, Y. He, and L. Guan, "Optimal resource allocation for multimedia cloud in Priority Service Scheme," in *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–4, 2012.

[18] A. Aggarwal and S. Jain, "Efficient Optimal Algorithm and Task Scheduling in Cloud Computing Environment" , International Journalal of Computer Trends and Technology (IJCTT), vol. 9, pp. 344-349, Mar 2014.

[19] B. R. Raman, R. Calheiros, R.N.(2009), "Modeling and Simulation of Scalable Cloud Environment and the CloudSim Toolkit: Challenges and Opportunities'', IEEE publication , pp1-11, 2009.

[20] Green Computing definition by IFG Standard, http://en.m.wikipedia.org/wiki/Green_computing

[21] S. Murugesan, "Going Green with IT: Your Responsibility toward Environmental Sustainability," Cutter Business & mdash, IT Strategies Executive Report, vol. 10, no. 8, 2007.

[22] S. Murugesan, "Harnessing Green IT: Principles and Practices," Cutter Business & mdash, IT Strategies Executive Report, vol. 10, no. 1, , IEEE Computer Society, 2007.

[23] S. K. Garg and R. Buyya, "Green Cloud computing and Environmental Sustainability", DOI: 10.1002/9781118305393, ch16, John Wiley & Sons, Ltd, 2012.

[24] H. Andy, "Green Computing", Communication of the ACM 51.11, pp. 11-13, 2008

[25] R. Stephen, "Green it more than a three percent solution?" *Internet Computing*, IEEE 13.4 (2009): pp. 74-78, 2009.

[26] High Performance Computing- Definition, by HPC Wales, CYRU, http://www.hpcwales.co.uk/what-is-hpc

[27] U.S. EPA, "Report to congress on server and data center energy efficiency public law 109-431," 2007.

[28] The Climate Group, "Smart 2020: Enabling the low carbon economy in the information age," June 2008.

[29] Open Data Center Alliance, "Open data center alliance usage: carbon footprint values," June 2011.

[30] VMWare, "Vmware® distributed power management white paper", http://www.vmware.com/files/pdf/DPM.pdf

[31] A. Verma, G. Dasgupta, T. K. Nayak, P. De, and R. Kothari, "Server workload analysis for power minimization using consolidation," in USENIX'09. Berkeley, CA, USA: USENIX Association, pp. 28–28, 2009.

[32] R. Yanggratoke, F. Wuhib and R. Stadler, "Gossip-based Resource Allocation for Green Computing in Large clouds," in *Proc. International Conference on Network and Service Management (ICNSM)*, pp. 1 –8, 2011.

[33] T. Forell, D. Milojicic and V. Talwar, "Cloud Management: Challenges and Opportunities", in *Proc. of IEEE International Parallel & Distributed Processing Symposium*, pp. 881-889, 2011.

[34] Cullen Bash et al., "Dynamically Assessing Sustainability of Data Center and Cloud", in *Proc. of HP Internal Technical Conference*, 2011.

[35] R. Santhosh and T. Ravichandran, "Pre-emptive Scheduling of On-line Real Time Services With Task Migration for Cloud Computing", in

Proc. IEEE Conference on Pattern Recognition, Informatics and Mobile Engineering (PRIME), pp. 1-6, 2013.

[36] J. Ma, “A Novel Heuristic Genetic Load Balancing Algorithm in Grid Computing”, in *Proc. of IEEE International Conference on Intelligent Human-Machine Systems and Cybernetics*, pp. 166-169, 2010.

[37] J. Alcaraz and C. Maroto, A robust genetic algorithm for resource allocation in project scheduling, Annals of Operations Research 102 (2001) pp. 83–109.

[38] S. Hartmann, A competitive genetic algorithm for the resource-constrained project scheduling, Naval Research Logistics 456 (1998) pp. 733–750.

[39] U. Kohlmorgen, H. Schmeck and K. Haase, Experiences with fine-grained parallel genetic algorithms, Annals of Operations Research 90 (1999) pp. 203–219.

[40] K. Li, G. Xu, G. Zhao, Y. Dong and D. Wang, “Cloud Task scheduling on Load Balancing Ant Colony Optimization”, in the *Proc. of IEEE Sixth Annual ChinaGrid Conference*, 2011, pp. 3-9

[41] D. Merkle, M. Middendorf and H. Schmeck, ”Ant Colony Optimization for Resource-Constrained Project Scheduling”, in the *Proc. of IEEE transactions on Evolutionary Computation*, *Vol. 6, NO. 4, August 2002*, pp. 333-346

[42] B. Tierney, W. Johnston, J. Lee and M. Thompson, “A data intensive distributed computing architecture for grid applications”, Future Generation Computing System 2000, 16(5): pp. 473-481, 2000.

[43] G. Guo-Ning and H. Ting-Lei, “Genetic Simulated Annealing Algorithm for Task Scheduling based on Cloud Computing Environment”, in *Proc. of International Conference on Intelligent Computing and Integrated Systems*, pp. 60-63, 2010

[44] M. Dorigo and C. Blum, “Ant colony optimization theory: A survey” in Theoretical Computer Science 344 (2–3) (2005), DOI: 10.1016/j.tcs.2005.05.020, pp.243–278, 2005.

[45] M. Dorigo and L.M. Gambardella, “Ant colony system: A cooperative learning approach to the traveling salesman problem”, in IEEE Transactions on Evolutionary Computation (1997), pp. 53–66, 1997.

[46] Bagherzadeh, J., MadadyarAdeh, M., “An Improved Ant Algorithm for Grid Scheduling Problem” in 14th International CSI Computer Conference (CSICC), pp.323-328, 2009.

[47] Lorpunmanee, S., Sap, M.N, Abdul Hanan Abdullah, A.H., “An Ant Colony Optimization for Dynamic Job Scheduling in Grid Environment” in Proceedings of World Academy of Science, English and Technology Volume 23 august 2007, ISSN 1307-6884, 2007.

[48] A. Singh and M. Kalra, “PSO Based Load Balancing Algorithm for Cloud Environment”, Thesis, 2014.

[49] M. Clerc, “The Swarm and the Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization”, in IEEE *Proc. of the 1999 Congress on Evolutionary Computation, Washington, DC, vol. 3,* pp. 19-24,1999.

[50] Y. Shi and R. C. Eberhart, “Parameter selection in particle swarm optimization”, in *7th International Conference on Evolutionary Programming, San Diego, California, USA*, pp. 591-600, Springer 1998.

[51] R. C. Eberhart and Y. Shi, “Particle Swarm Optimization: Developments, Applications and Resources”, in *IEEE Proc. of the 2001Congress on Evolutionary Computation, Seoul, vol.1*,pp-81-86, 2001.

[52] J. Kenndy and R. Eberhart, “Particle Swarm Optimization”, in *IEEE Proc. of International Conference on Neural Networks, vol. 4,* pp. 1942-1948,1999.

[53] Lovejit Singh and Sarbjeet Singh, “A Genetic Algorithm for Scheduling Workflow Applications in Unreliable Cloud Environment”, Second International Conference on Security in Computer Networks and Distributed Systems (SNDS-2014), Springer Berlin Heidelberg, pp. 139-150, 2014.