Proactive Job Scheduling and Migration using Artificial Neural Networks for Volunteer Grid

Saddaf Rubab[1, *], Mohd Fadzil Hassan[2], Ahmad Kamil Mahmood[3], Syed Nasir Mehmood Shah[4]
{saddaf_g02754@utp.edu.my[1,*], mfadzil_hassan@petronas.com.my[2], kamilmh@petronas.com.my[3], dr.shah@kicsit.edu.pk[4]}

Department of Computer and Information Sciences,
Universiti Teknologi PETRONAS, Seri Iskandar, Perak Darul Ridzuan, 32610, Malaysia[1, 2, 3]
Department of Computer Sciences, Dr. A. Q. Khan Institute of Computer Sciences & Information Technology, Kahuta, Pakistan[4]

Abstract. A desktop grid is heterogeneous collections of local and volunteer resources. These resources can be assigned to heterogeneous jobs whereas these resources cannot be guaranteed to be available every time of job execution. Therefore, the resource availability and load forecast can help to minimize the job failures and job migration. In this paper, a forecast based proactive job scheduling and migration (PJS-ANN) has been proposed using artificial neural networks to make load forecasts for scheduling the jobs to reliable volunteer resources. The proposed method performance has been compared with conventional load balancing (LB) and no-migration (NM) algorithms. The performance comparisons demonstrate that the PJS-ANN has lower turnaround time per job and job failure rate has been significantly improved.

Keywords: Volunteer grid, job scheduling, job migration, artificial neural networks, job failure, resources

1 Introduction

Todays' personal computers owned by the public are very powerful and can execute millions of instructions in a much less time as compared to the past computing machines [1]. But these personal computers are left unused or not in use for a long time. These unused computing resources can be used in a desktop grid, regardless of the location and makes use of it for solving scientific problems. The desktop grid facilitates in making low-cost computing infrastructure for scientific purposes [2-6]. Desktop grid falls into two categories, i.e., enterprise and volunteer. The former one comprises of surplus local compute resources or located within an enterprise. The volunteer desktop grid (VG) constitutes of resources voluntarily contributed for solving scientific problems or executes projects like SETI@home, Folding@home etc.

The resources in volunteer desktop grid computing can be withdrawn anytime as their availability is not promised by the individuals for the full length of the scientific project . Therefore, volunteer grid is highly dynamic and doesn't guarantee for any availability time of resources. The volatility of [7] resources limits the range of jobs and applications to be executed. The availability of volunteer resources can be forecasted to increase the benefits of this cost-effective infrastructure. Resource forecast is a way of estimating the future resource availability or unavailability intervals for a given forecast horizon based on the available information about the state of the system.

The load and resource availability forecast is receiving much more attention in the recent years to make forecasts of resource availability/unavailability, job load etc. for various science platforms like grid systems, weather systems etc. There are different forecast methods like linear problems [8], exponential smoothing [9], multiple linear regression [10], Kalman filtering [11], autoregressive integrated moving average (ARIMA) [12], Support Vector Machine (SVM) [11-13] and Artificial Neural Networks (ANN) [14, 15]. It is useful for assigning the longer jobs and application runs to volunteer resources after forecasting the availability for the full length of execution. This will eventually limit the job failures and reduce the job migrations. The resource availability forecast is also helpful in planning the proactive job migrations.

For an instance, if the arrival time and required CPU time by the job are known, the resource availability forecast during that time intervals will help to improve the job execution and overall makespan i.e. the total time elapse from start to finish of a job. If resource availability is forecasted with high accuracy, the overall job-resource allocation itself results better and ultimately improves the volunteer grid performance.

The resources in volunteer grid either go temporarily unavailable or permanently leave the grid. The resources with frequent temporal unavailability will result in more job migration and communication cost. This resource behavior can also be found in the resource dataset SETI@home [16, 17] extracted from a real system.

The work presents a forecast based proactive job scheduling using artificial neural network (PJS-ANN) which considers the individual resources before scheduling any new job. The performance comparison has been made between proposed PJS-ANN with the load balancing (LB) and no migration (NM).

The outline of presented paper is as follows: Section 2 details about the related literature. Section 3 devotes to the proposed approach as well as it provides a brief introduction about ANN. Section 4 briefs the simulation results and performance comparisons. The research work presented is concluded in Section 5.

2 Related Work

The proactive job scheduling and job migration in VG has been attempted using different heuristic approaches but there is little evidence of incorporating machine learning algorithms as per the knowledge of authors. Machine learning algorithms can be used in VG to forecast the jobs and resources. However, the machine learning algorithms have been employed in distributed computing environments like cloud, and clusters etc. for other research issues as well. In this section, the brief literature about the load balancing techniques, job migration policies and machine learning algorithms' applications in distributed computing will be discussed.

A proximity based load balancing was proposed for peer-to-peer VG systems [18]. The authors manage the load among the nodes based on the communication overhead and already available load at each node. The proximity based load balancing results in less response time. A decentralized scheduling algorithm for balancing the load dynamically was presented by Di et. al.[19]. This work has been improved in [20] by the authors to minimize the conflicts to balance the load between the nodes in the desktop grid. The authors claim to reduce the conflicts using the Bernoulli probabilistic model [20], in a way to select the lightly loaded node by the heavily loaded node to migrate jobs. A dynamic load balancing technique has been presented in [21], that suggest using the variable threshold because the network size in the grid may vary and cannot be static. The authors also suggested categorizing the load into different categories. But authors didn't consider the history of grid nodes to evaluate the availability and scalability of grid nodes.

Ant Colony Optimization approach for load balancing was introduced [22] to invoke a response from the best of available network resources which can reduce the response time. Another approach [23] based on information dissemination has been proposed that can distribute the load among grid nodes without using load re-balancing by migrating jobs. The approach [23] used the nodes rate adjustment that affects the job scheduling decision, but the proposed approach is costly for large-scale heterogeneous grids.

There are hierarchical approaches proposed in the literature to schedule the grid workload. In [24], the nodes have been classified as overloaded or lightly loaded if the load is more than or less than the threshold value respectively. Another approach following the same principle has been presented in [25], it balances the load using a threshold value calculated using load deviation. A survey of hierarchical load balancing approached has been presented in [26]. The proposed load balancing approaches and policies are mostly reactive, which means they don't proactively plan the job migration to balance the load.

It has been suggested in the literature that machine learning approaches can assist in proactively scheduling jobs and balancing the load [27-29]. A resource provisioning approach based on the load prediction service has been used to manage the game users and resource allocation [29]. Using this load prediction service [29], authors claim to take timely load balancing actions and prevent servers from being over-loaded. The machine learning and sliding window concept has been combined to proactively plan the resource allocation in cloud environment [30] and suggested that machine learning algorithms can improve the prediction accuracy to a satisfactory level.

It has been concluded from the literature study that job scheduling policies can be more efficient if proactively schedule the jobs such that job migration is not required frequently. This requires forecasting the load and resource activities using the machine learning algorithms, as suggested in the literature study. The work, therefore, will aim to propose a forecast based proactive job scheduling and job migration for volunteer grid computing environment. The proposed proactive algorithm will require the prior knowledge about the resource utilization. This data has been gathered executing LCG1 job dataset [31, 32] at different intervals on SETI@home [16, 17] resource dataset in a simulated VG environment.

3 Proactive Job Scheduling using ANN (PJS-ANN)

Suppose the new jobs coming to volunteer grid and waiting for the resources have been allocated the jobs using FCFS. Later the scheduled jobs have to migrate from one resource to another due to resource temporal unavailability or withdrawal. When a resource has been selected to complete the migrated job, at that time it is available but it is possible that by the time job completely migrated the selected resource is not the best resource and the time latency adds up to the total job execution time.

In such situations, the forecasts may help to foresee which resource will be available to schedule the new job or migrating the job to alternate resource. The forecasts has been attempted in distributed computing environments using linear problems [8], exponential smoothing [9], multiple linear regression [10], Kalman filtering [11], autoregressive integrated moving average (ARIMA) [12], Support Vector Machine (SVM) [11-13] and Artificial Neural Networks (ANN) [14, 15]. The ANN is one of the most practiced forecast method in the literature. The proposed algorithm PJS-ANN integrates ANN to make the forecasts about resource availability and load profiles.

3.1 Design of ANN

The Multi-Layered Perceptron (MLP) network is the most common type of ANN used for predictions and forecast models. MLPs have been applied successfully in many non-linear systems. Figure 1 illustrates a five layered ANN with three hidden layers. The input layer has five input neurons $I = \{I_1,I_2,I_3,I_4,I_5\}$ and output layer has one output neuron $O = \{O_1\}$. Each of the hidden layers has three hidden neurons $H = \{H_1,H_2,H_3\}$.

Each layer has some weight that is supplementary for the neurons of next layer. During the learning phase of ANN, the weights can be changed to properly map the actual output $Out_{actual}$with the forecasted output $Out_{forecast}$. The change in weights is depends on the error in $Out_{forecast}$ that is back propagated by using different techniques such as delta-rule method.

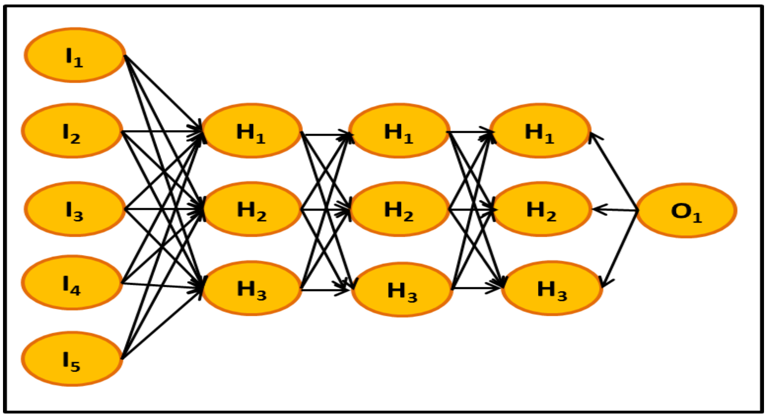

Fig. 1. Five Layered ANN

The volunteer resource data is used periodically to train the neural network and forecast the future jobs or load to be accommodated at each resource. During the training of ANN, the load patterns will be fed to the next layer. The five layered ANN used in PJS-ANN is trained at the rate of 0.8 using back propagation algorithm. The training process at each node terminates when the normalized mean square error is minimum. The error graph for one of the VG resource has been illustrated in Figure 2.

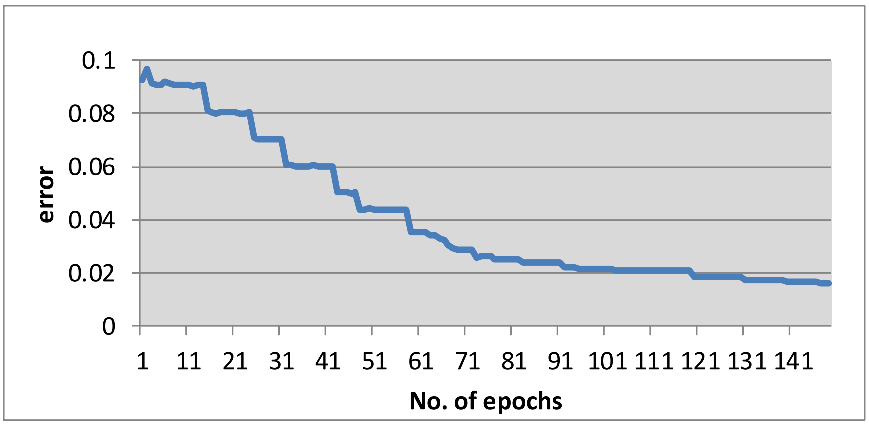

Fig. 2. ANN error plot for a random VG resource

3.2 PJS-ANN Algorithm

Each resource will forecast the future job and load that it can execute for the next job interval VG will receive the jobs. The node will forecast the jobs using trained ANN. The forecasts are then submitted to the resource broker in VG environment. When a new job will

```
Create ANN for each resource with 'i' input elements, 'h' hidden neurons and one output neuron
Train the ANN for each resource using BP algorithm from the periodically collected average job
Forecast the resource future job/load using trained ANN for next time interval set
For each resource 'i' in VG do
  For each unscheduled job 'j' do
      Access the resource availability and make a set of resources for job 'j'
      Choose a resource with near availability from VG
            Compute estimated turnaround time for job 'j' at resource 'i' from the set
            if(estimated turnaround time at resource 'i' < estimated turnaround time at resource 'i+1')
            then
                        Schedule job 'j' at resource 'i'
                        Tag job 'j' as scheduled
            else
                        Access resource availability again and make new set of resources for job 'j'
            endif
  endfor
            if (job 'j' tag as migrate)
            then
                        Schedule job 'j' to another resource from set of resources
            endif
  For each interval 't' do
            Forecast job/load for next interval and update the resource broker
  endfor
endfor
```

Fig. 3. PJS-ANN Algorithm

arrive at VG, the resource broker will allocate a resource to new job depending on the forecasts submitted to broker earlier. The selection of resource directly relates with the expected turnaround time a VG resource can offer a job. A resource expected to offer the least turnaround time using the trained ANN, will be selected. This will decrease the job waiting time and latency time. If a resource is forecasted which can offer least turnaround time, there is also less chance that any failure will occur. In case a job failure occurs, a new resource has to be selected using the updated resource job statistics and forecasts. The PJS-ANN algorithm is detailed as follows in Figure 3.

4 Simulation Results

The effect of forecast in job scheduling using ANN has been evaluated using the two sets of resources available resources, i.e., 100 and 200 resources that can execute the submitted jobs to VG. The average turnaround time of jobs scheduled using PJS-ANN, NM and LB has been illustrated in Figure 4. It can be observed that the PJS-ANN outperforms when number of available resources is either 100 or 200. The turnaround time of LB is close to PJS-ANN as the average turnaround time for NM does not include the jobs that failed and missed the deadline. The turnaround time has been calculated for the first 10,000 jobs from LCG1 dataset [31, 33].

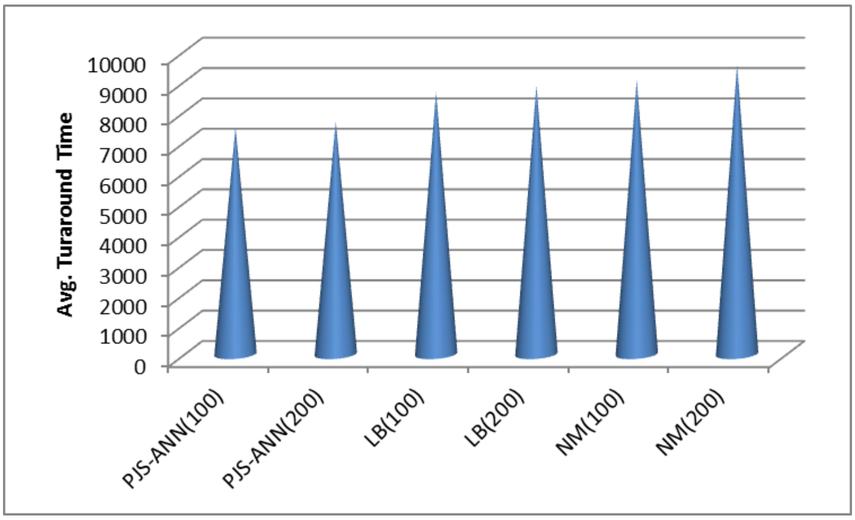

Fig. 4. Avg. Turnaround Time for 10,000 Jobs

The number of jobs arriving at VG varies to study the effect on the performance of PJS-ANN while keeping the available resources 100 and 200. Figure 5 presents the number of jobs completed. The average number of jobs completed using PJS-ANN is significantly higher than LB and NM as it involves forecast about job and estimated turnaround time offered by available resources.

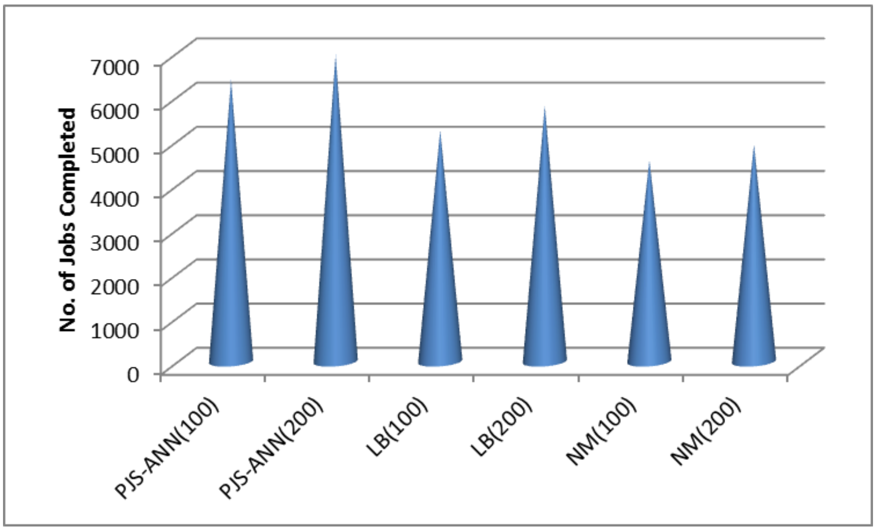

Fig. 5. No. of Jobs Completed

Figure 6 illustrates the number of jobs migrated. The comparison has been made only between PJS-ANN and LB as NM involves no job migration at all. It can be studied that forecasts decrease the number of job migrations. The number of job migrations increase as the rate of job arrival increases. If more number of resources is available to execute the jobs submitted to VG, the number migrations will decrease.

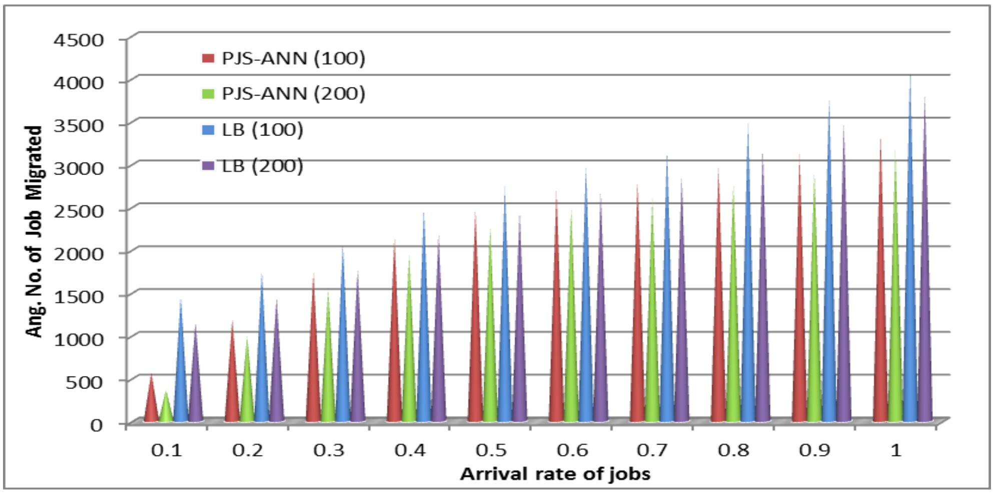

Fig. 6. No. of Jobs Migrated

If more number of resources is available to execute the submitted jobs the response time will decrease. Figure 7 presents the avg. response time for PJS-ANN, LB and NM for varying number of resources and jobs.

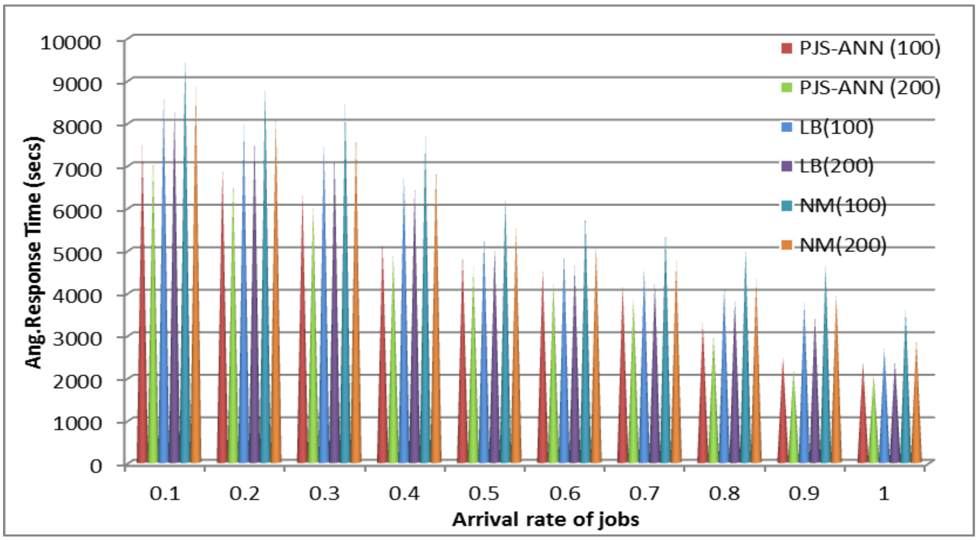

Fig. 7. Average Response Time for 100 and 200 Resources

5 Conclusion

The work presents a proactive job scheduling algorithm for volunteer grid computing environment which can reduce the job migrations. The proposed algorithm PJS-ANN involves forecasts using 5-layered ANN. The results supports that ANN is feasible solution to forecast the job/load on a resource that can further help to estimate the turnaround time for future jobs arriving at VG. PJS-ANN has been compared with NM and LB algorithms. The PJS-ANN performs well with the increasing rate of jobs while maintaining the same number of available resources. The results validate that PJS-ANN outperforms and can effectively complete the jobs with fewer deadline misses. Using the results, it can also be depicted that if the forecast is not accurate, there will be unnecessary job migrations for load balancing which will waste the VG resources. The response time has been reduced considerably using the proposed algorithm. The work can be extended to implement the optimized resource management system and can also be adopted in other distributed computing infrastructures.

References

[1] "Volunteer computing," 2013.

[2] *World Community Grid*. Available: http://www.worldcommunitygrid.org/

[3] C.-J. Huang, C.-T. Guan, H.-M. Chen, Y.-W. Wang, S.-C. Chang, C.-Y. Li, *et al.*, "An adaptive resource management scheme in cloud computing," *Engineering Applications of Artificial Intelligence,* vol. 26, pp. 382-389, 2013.

[4] I. Foster, C. Kesselman, and S. Tuecke, "The anatomy of the grid," *Berman et al.[2],* pp. 171-197, 2003.

[5] D. P. Anderson, "Volunteer computing: the ultimate cloud," *ACM Crossroads,* vol. 16, pp. 7-10, 2010.

[6] M. M. Waldrop, "Grid computing," *Technology Review (May 2002),* vol. 31, p. 37, 2002.

[7] N. P. Preve, *Grid Computing: Towards a Global Interconnected Infrastructure*: Springer-Verlag London, 2011.

[8] E. Christoforou, A. F. Anta, C. Georgiou, M. A. Mosteiro, and A. Sánchez, "Applying the dynamics of evolution to achieve reliability in master–worker computing," *Concurrency and Computation: Practice and Experience,* vol. 25, pp. 2363-2380, 2013.

[9] J. W. Taylor and R. D. Snyder, "Forecasting intraday time series with multiple seasonal cycles using parsimonious seasonal exponential smoothing," *Omega,* vol. 40, pp. 748-757, 2012.

[10] E. A. Feinberg and D. Genethliou, "Load forecasting," in *Applied mathematics for restructured electric power systems*, ed: Springer, 2005, pp. 269-285.

[11] C.-N. Ko and C.-M. Lee, "Short-term load forecasting using SVR (support vector regression)-based radial basis function neural network with dual extended Kalman filter," *Energy,* vol. 49, pp. 413-422, 2013.

[12] A. Kavousi-Fard and F. Kavousi-Fard, "A new hybrid correction method for short-term load forecasting based on ARIMA, SVR and CSA," *Journal of Experimental & Theoretical Artificial Intelligence,* vol. 25, pp. 559-574, 2013.

[13] A. Selakov, D. Cvijetinović, L. Milović, S. Mellon, and D. Bekut, "Hybrid PSO–SVM method for short-term load forecasting during periods with significant temperature variations in city of Burbank," *Applied Soft Computing,* vol. 16, pp. 80-88, 2014.

[14] Z. Liu, W. Li, and W. Sun, "A novel method of short-term load forecasting based on multiwavelet transform and multiple neural networks," *Neural Computing and Applications,* vol. 22, pp. 271-277, 2013.

[15] R.-A. Hooshmand, H. Amooshahi, and M. Parastegari, "A hybrid intelligent algorithm based short-term load forecasting approach," *International Journal of Electrical Power & Energy Systems,* vol. 45, pp. 313-324, 2013.

[16] D. P. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer, "SETI@ home: an experiment in public-resource computing," *Communications of the ACM,* vol. 45, pp. 56-61, 2002.

[17] M. Roehrig, W. Ziegler, and P. Wieder, "Grid scheduling dictionary of terms and keywords," *GFD-I. 11, Grid Scheduling Dictionary WG (SD-WG),* 2002.

[18] T. Ghafarian, H. Deldari, B. Javadi, and R. Buyya, "A proximity-aware load balancing in peer-to-peer-based volunteer computing systems," *The Journal of Supercomputing,* vol. 65, pp. 797-822, 2013.

[19] S. Di, C.-L. Wang, and D. H. Hu, "Gossip-based dynamic load balancing in a self-organized desktop grid," *Proceedings of the 10th high-performance computing Asia, HPCAsia,* pp. 85-92, 2009.

[20] S. Di and C.-L. Wang, "Conflict-minimizing dynamic load balancing for P2P desktop grid," in *Grid Computing (GRID), 2010 11th IEEE/ACM International Conference on*, 2010, pp. 137-144.

[21] N. Rathore and I. Chana, "Variable threshold-based hierarchical load balancing technique in Grid," *Engineering with computers*, vol. 31, pp. 597-615, 2015.

[22] S. A. Ludwig and A. Moallem, "Swarm intelligence approaches for grid load balancing," *Journal of Grid Computing*, vol. 9, pp. 279-301, 2011.

[23] D. C. Erdil and M. J. Lewis, "Dynamic grid load sharing with adaptive dissemination protocols," *The Journal of Supercomputing*, vol. 59, pp. 1139-1166, 2012.

[24] S. F. El-Zoghdy, "A hierarchical load balancing policy for grid computing environment," *International Journal of Computer Network and Information Security*, vol. 4, p. 1, 2012.

[25] S. El-Zoghdy and A. Ghoneim, "A Multi-Class Task Scheduling Strategy for Heterogeneous Distributed Computing Systems," *KSII Transactions on Internet & Information Systems*, vol. 10, 2016.

[26] D. K. Patel, D. Tripathy, and C. Tripathy, "Survey of load balancing techniques for Grid," *Journal of Network and Computer Applications*, vol. 65, pp. 103-119, 2016.

[27] T. Helmy and S. A. Shahab, "Machine Learning-Based Adaptive Load Balancing Framework for Distributed Object Computing," in *Advances in Grid and Pervasive Computing: First International Conference, GPC 2006, Taichung, Taiwan, May 3-5, 2006. Proceedings*, Y.-C. Chung and J. E. Moreira, Eds., ed Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 488-497.

[28] R. Hu, J. Jiang, G. Liu, and L. Wang, "Efficient resources provisioning based on load forecasting in cloud," *The Scientific World Journal*, vol. 2014, 2014.

[29] R. Prodan and V. Nae, "Prediction-based real-time resource provisioning for massively multiplayer online games," *Future Generation Computer Systems*, vol. 25, pp. 785-793, 2009.

[30] S. Islam, J. Keung, K. Lee, and A. Liu, "Empirical prediction models for adaptive resource provisioning in the cloud," *Future Generation Computer Systems*, vol. 28, pp. 155-162, 2012.

[31] (25 December 2016). *Trace analysis report LCG*. Available: http://gwa.ewi.tudelft.nl/datasets/gwa-t-11-lcg/report/

[32] (24 Oct 2011
). *The Grid Workloads Archive*. Available: http://gwa.ewi.tudelft.nllpmwiki/

[33] A. Duarte, P. Nyczyk, A. Retico, and D. Vicinanza, "Monitoring the EGEE/WLCG grid services," in *Journal of Physics: Conference Series*, 2008, p. 052014.