Investigation and Analysis of Google Cluster Usage Traces: Facts and Real-Time Issues

[1]Ali Hussein Ali Alnooh, [2] Dhuha Basheer Abdullah
[1,2] Department of Computer Science, College of Computer Science and Mathematics,
University of Mosul, Nineveh, Iraq
Email : [1]a_alnooh@yahoo.com , [2] rtdm2005@yahoo.com

Abstract— The term "Cloud" has become ubiquitous in our life since it deals with most of our activities. In fact, we are practising cloud services in most of our social, academic, and business tasks. The technological era we are currently living in, is almost based on Internet services. Nowadays, cloud computing and cloud load balancing are considered as the core for cloud services. Thus, the fields of cloud computing and cloud load balancing have attracted various researchers to develop innovative methods aiming at enhancing cloud services and providing cloud end users with adequate services. This paper discusses the field of cloud load balancing in two important aspects. The first aspect includes taking a deep look on the tasks and servers on Google cluster usage traces, while the second aspect includes the discussion about how load balancing algorithms deal with tasks that have time constraints (e.g., deadline). The findings of this paper exhibited that when using real time concepts with the selected scheduling algorithms, such algorithms require further enhancements to satisfy the need of real time tasks. Netlogo simulator was used as a tool to execute the practical side of the work.

Keywords— Cloud Load Balancing; Cloud Computing Algorithms; Google Clusters usage traces; Real Time tasks

I. INTRODUCTION

In our daily activities we use many different applications in different fields of our life. The goal of using these applications is to make our life more easy and have our works done as fast as possible. Most of the applications we use are considered as cloud based. Each application includes many services provided to serve end users [1]. Typically, users send requests through their applications, which in turn satisfy these requests. Usually, user requests are assigned to particular servers to be executed and send back the results to the requested sender. In the context of cloud computing, user requests take the form of jobs each of which can be divided into many tasks [2]. Some of tasks have time constraints (e.g., deadline) that should be executed before. Therefore, it is needed to assign tasks with time constraints to appropriate servers and make every task meet its deadline. Performing such a procedure can be processed at load balancer side [3]. Load balancing is the process of distributing tasks to servers in a more convenient and optimal way. Figure 1 shows the position of Google load balancer in the architecture diagram [4].

Load balancing is also considered as an important and a challenging issue when it comes to dealing with real time tasks [5]. In general, real time tasks need to be served and provided with resources before their deadline [6]. In the literature, there are a few real time load balancers proposed by load balancing developers. In fact, it is not an easy task to find or to develop a real time load balancer that satisfies users' needs.

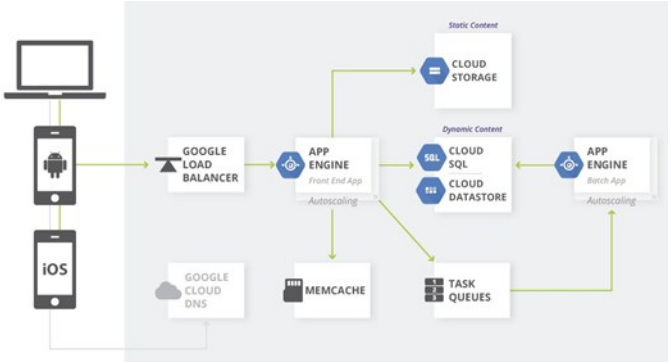

Fig. 1. The position of Google load balancer in Google App Engine Standard Environment

This paper discusses the field of cloud load balancing in terms of; 1) facts on the tasks and servers on Google cluster usage traces, and 2) how load balancing algorithms deal with tasks when having time constraints (e.g., deadline). To this end, we involve two well-known algorithms in simulating two load balancers, these algorithms are:

- Least Connection: This approach assigns tasks to the servers that have less load of tasks comparing to other servers. It is easy to code it but does not take into considerations the capacity of the selected server and tasks requirements [7], so it will suffer in case of real time tasks and overloaded servers.
- Round Robin: This is a well-known approach in the literature, it distributes the incoming tasks to servers in an even way at each round [7], but it contains some drawbacks like higher waiting time and context switching burdens.

The main contribution of this paper can be found in taking a deep look on resource requirements for jobs and tasks as well as Google machines capacities. Also investigating the real time issue for tasks when having time constraints and the suggestion of an equation to calculate the tasks deadline which yields to missed tasks.

This paper is divided into five sections, the next section shows works related to our work. Section III explains facts and shows statistics regarding to Google cluster data traces. Section

978-1-5386-7858-9/18/$31.00 ©2018 IEEE

IV provides with the experimental results obtained. Finally, we conclude our work in Section V.

II. Related Works

Although for its importance for cloud developers, Google cluster data traces has not given much attention by the research community. Few number of research works discuss and analyse these data and came up with facts and solutions for cloud developers. In [8], the authors performed a deep study on Google cluster data traces. This study takes the features of resources usage for jobs and how these requirements vary overtime. They also discuss the correlations among job resources (e.g., memory and CPU). They propose an approach for addressing the issue of resource allocation when they are changed overtime.

In the same context, Adhikari and Amgoth [9] proposed an algorithm for Infrastructure as a Service (IaaS) cloud. Their algorithm assigns tasks to the appropriate servers in terms of the amount of resources needed aiming at utilizing resources as much as possible. Other research works focus on the issue of load balancing (our focus in this paper), which is considered as one of the main concerns in cloud computing field. Some of the works tried to integrate other fields of study with the cloud computing field. The authors in [10] used techniques from the evolutionary and swarm intelligence algorithms to develop new methods for optimizing the issue of resources utilization. Furthermore, the concepts of Genetic Algorithms (GA) can also be involved in developing new approaches for load balancing issues such as the work of Dasgupta et al. [11].

Other important issues in the current available load balancers such as the real time issue is also not given much attention in the literature. Wu and Song [12] proposed a real time load balancer for decreasing the overhead generated on particular servers. Lee et al. [6] improved VMM scheduler to be appropriate for soft-real time applications. The improved load balancer was developed to overcome the latency issue when assigning real time tasks. The technique they used was based on providing the required resources for tasks at time.

III. Dataset Facts and Statistical Analysis

In this paper, we use Google cluster usage traces dataset from Google [13]. A Google cluster is a set of machines (servers), connected by a high bandwidth network. All clusters share a common cluster management system that allocates work to machines. The loads arrive at a cluster in the form of jobs. A job consists of one or more tasks, each of which is accompanied by a set of resource requirements used for scheduling tasks onto the appropriate servers. Each task consists of multiple processes; to be executed on a single server. Tasks and jobs are scheduled onto machines according to a particular algorithm. Resource requirements and usage data for tasks are derived from information provided by the clusters' management system.

Google dataset is formed as tables, each of which describes particular information on Machines and their attributes, Tasks Events, Jobs Events, and Resource Usage. The figures in the next sections describe the dataset and more details will be provided. It should be noticed that all the data provided by Google does not reflect the real values of the traces. This is, of course, for privacy purposes, but instead they normalized the data in a way that enables researchers to use this data for analysis and experiments [13].

A. Machine Failure

As mentioned, each Google cluster has many servers for executing the incoming tasks and these servers are not always available to be utilized since a failure in machines may happen. Figure 2 depicts the availability of machines for executing tasks.

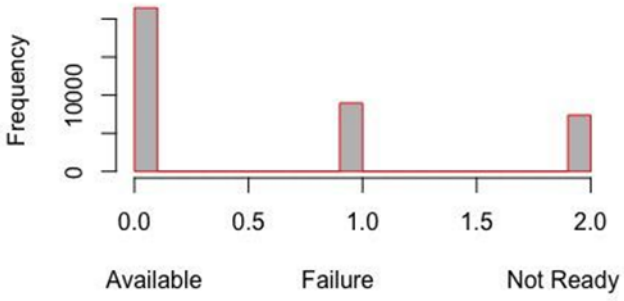

Fig. 2. Google machines availability

This figure also shows that the probability of having failure in a machine during or before the execution of particular tasks is about 0.23 and this amount varies among clusters. Therefore, the probability of having failure in a machine on a particular cluster is not fixed.

This means when the load balancer assigns a task to a cluster machine, it is not guaranteed to get the execution completed for the current task and the previously loaded tasks. For this reason, it is needed for the load balancer to consider this issue during the design phase.

B. Capacity of Machines

Figures 3 and 4 show the CPU and memory capacities for Google machines. It is clear that the CPU and Memory capacities of machines vary from a machine to another one. Furthermore, according to the information provided by the dataset, these two figures reflect the fact that Google does not provide many machines with high capacities of CPU and memory to their customers. This interprets the high price of Google hosting services for each single instance.

C. Tasks Requests for Resources

Another statistic on the dataset shows that the majority of tasks executed on Google servers do not require high volume of resources, but instead they use low or medium level of resources as depicted in Figures 5 and 6. This is maybe due to the low availability of high capacity servers that Google provide for regular users

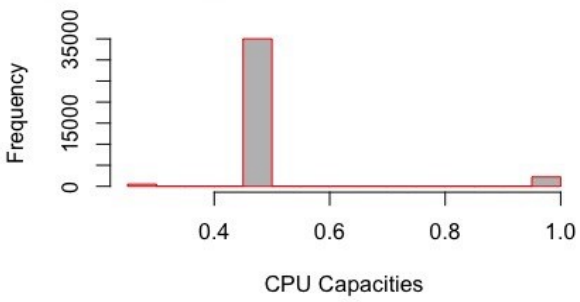

Fig. 3. Google machines CPU capacities

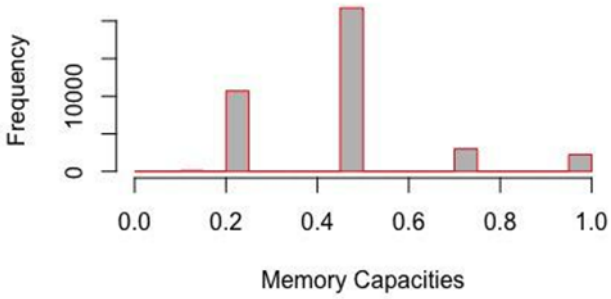

Fig. 4. Google machines memory capacities

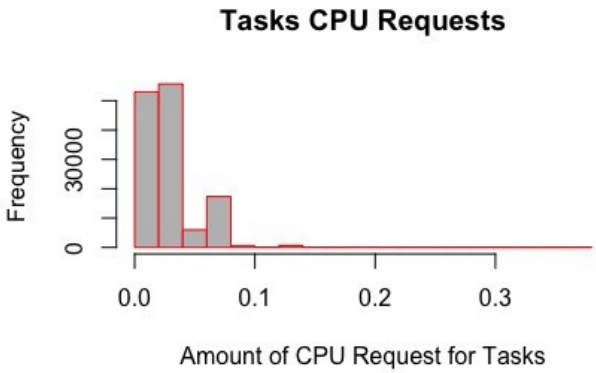

Fig. 5. Tasks CPU requests

According to the amount of CPU and memory requests, we observed that most tasks need low and medium amount of CPU and memory, and there is few requests need high CPU and memory capacities as shown in Figure 7.

Based on Figure 7, when a task needs a few amount of memory it also needs a few amount of CPU, which means there are few tasks that are considered as heavy tasks (e.g., tasks that need high capacities of CPU and memory).

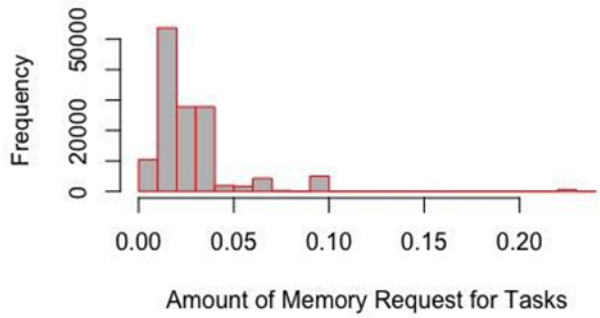

Fig 6. Tasks memory requests

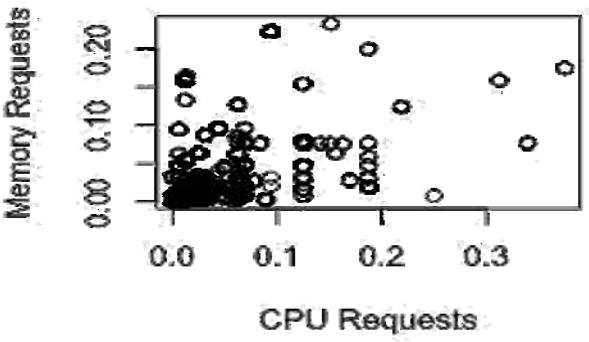

Fig. 7. Tasks CPU requests vs. tasks memory requests

D. Tasks Sensitivity to Latency

Each task in the dataset has to work under a particular scheduling class. Google does not mention in its dataset about the scheduling algorithms it uses. It only mention with a number that shows tasks sensitivity to latency. A scheduling class shows how latency sensitive a particular task is, and it takes the range 0 to 3 in the dataset. The high the value of scheduling class, the more the latency sensitive a task is.

Figure 8 depicts the distribution of scheduling classes used for tasks. This feature is useful for executing tasks in their real time, which is our concern in this paper.

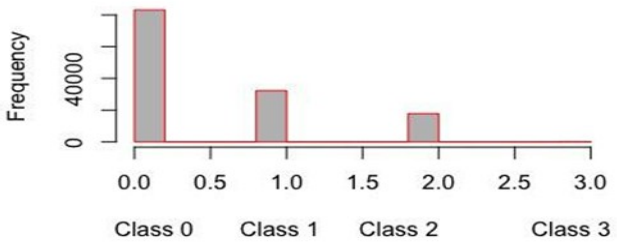

Fig. 8. Distribution of tasks scheduling classes

It is important to mention that a scheduling class is not a priority and they have different concepts. Scheduling class is related to machines policies in accessing machine resources, while priority shows whether a particular task is scheduled. This enables us to say that the scheduling algorithm used is very important when having hard time constraints for tasks. Figure 9 shows the distribution of priorities assigned to tasks.

Priorities are also useful for executing tasks in their real time. The parameter of priority is essential to be considered in the proposed approach and assigning a task with different requirements to a particular server with different capacities depends on tasks scheduling classes and priorities.

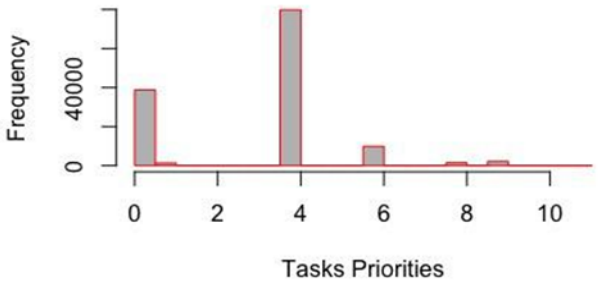

Fig. 9. Distribution of tasks priorities

IV. EXPERIMENTAL RESULTS

We simulate two load balancers, the first is based on Round Robin algorithm in assigning tasks to servers, while the second one is based on Least Connection algorithm. Both load balancers use Google cluster usage traces dataset in the simulations. The goal of these simulations is to test the time constraints for tasks and see how regular algorithms behave in terms of the number of missed deadline tasks.

For all the experiments in the simulations, we in involve approximately 127, 000 tasks and 50 servers. This data is imported from the original dataset from Google traces. The server we use for the simulations is operated under UNIX operating system.

To find the number of missed deadline tasks $\delta(T_l)$ in each algorithm, we first should calculate tasks deadlines using information provided by the dataset, which does not include tasks deadlines in its attributes. In the dataset, there are two attributes available, namely, Start Time $(T_l st)$ and End Time $(T_l et)$. These two parameters can be utilized in finding the deadlines of tasks. Before that, we should find the execution time of each task C_{T_l} which can be calculated as follows:

$$C_{T_l} = T_l et - T_l st$$

The deadline, then, of a task $\delta(T_l)$ represents the maximum response time to that task and can be calculated based on the following equation:

$$\delta(T_l) = C_{T_l} + C_{T_l} \,/\, T_l(SchC) \qquad (1)$$

where $T_l(SchC)$ denotes the scheduling class of task T_l. We form this equation to be appropriate with the dataset parameters since it makes sense when including the scheduling class of a task and express the latency sensitive concept. According to Equation 1, the high the value of sensitivity to latency, the close the deadline is. This fact is reflected in our proposed equation for deadline calculations.

A. Simulation Environment

In the practical side of our work, we designed a special purpose simulator for implementing and testing Round Robin and Least Connection algorithms as the base of the two load balancers. We simulated the data into graph complex network by assuming the servers and tasks as nodes having a relation between them depending on task latency and server capacity. The programming of the simulator is based on the concept of Multi Agent environment. We involve the NetLogo modelling, which is Java based programming environment depending on simulating the interaction between agents i.e. tasks and servers.

B. Testing Time Constraints

We used two well-known algorithms in the literature for implementing load balancing on Google cluster traces dataset. The goal of using these regular algorithms is to test the time constraints for the tasks. According to the implementation of Round Robin and Least Connection algorithms as load balancers, we first show the time consumed (tasks processing time inside each load balancer) by each algorithm.

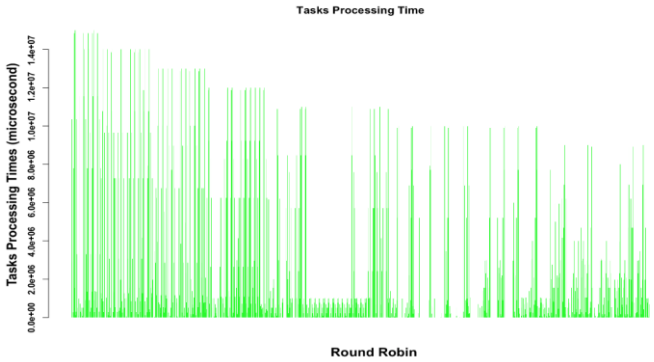

Fig. 10. The processing time of tasks using Round Robin algorithm for assigning tasks to servers

Figure 10, depicts the time consumed by Round Robin for all the tasks involved in the simulation.

Figure 11, shows the processing time of the tasks inside the load balancer when using Least Connection algorithm. It can be observed that the time consumed by this algorithm is less than from what has been obtained using Round Robin.

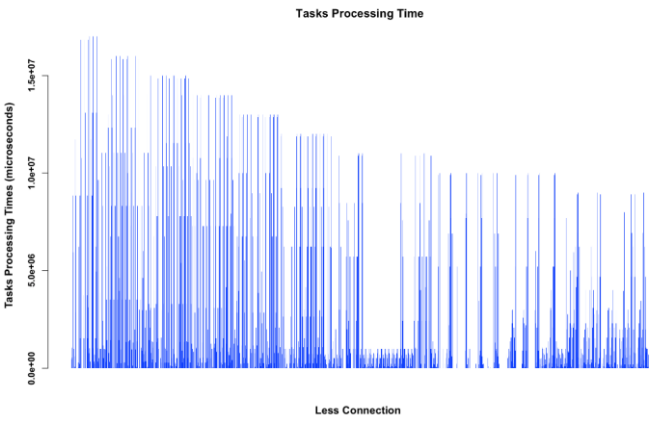

Fig. 11. The processing time of tasks using Least Connection algorithm for assigning tasks to servers

Now, we find the tasks that missed their deadlines inside the simulated load balancers. Figure 12, show the number of missed deadline tasks using Round Robin and Least Connection algorithms.

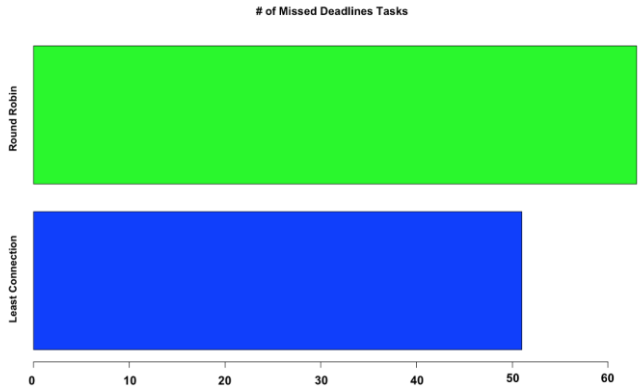

Fig. 12. The number of missed deadline tasks in Round Robin and Least Connection algorithms during the process of assigning tasks to servers

This figure reveals interesting numbers, there are many tasks missed their deadlines in both algorithms. The results show that about 63 and 51 tasks missed their deadlines in Round Robin and Least Connection respectively.

These results makes sense since these two algorithms do not take into considerations the time constraints when assigning tasks to servers (e.g., tasks deadline).

From the previous description, if we have real time tasks that need to meet their deadlines, a real time algorithm should be used. This algorithm should be designed in a way that takes into account each single time related parameter to tasks. As mentioned before, each system has critical tasks and these tasks should be given more attention than the other regular tasks in the system in terms of time constraints. Therefore, there is a need to an algorithm that handles real time tasks.

V. Conclusion and Future Work

Several studies focused on Google usage cluster traces in a statistical manner trying to abstract its features and cluster the data according to tasks behaviour. But rarely studies concentrated on the relation between tasks and servers as a real time point of view. In this paper, we discuss issues related to cloud computing and cloud load balancing. Two aspects discussed in this paper:

1) a deep look on the tasks and servers on Google traces, and some facts on Google cloud computing.

2) How load balancing algorithms deal with the time constraints of tasks such as the tasks that have hard time constraints.

The findings reveal several facts on Google cluster usage traces for tasks and servers as well as the behaviour of Round Robin and Least Connections algorithms when considering the time constraints of tasks.

In the findings i.e. according to figure 10, 11 and 12, we have seen that several tasks missed their deadlines, which is not desired when having a system with hard real time tasks.

The algorithms used in this work (Round Robin and Least Connections) cannot satisfy or handle real time tasks. Therefore, it can be concluded that there is a strong need to develop load balancing algorithms for real time purposes.

According to the previous conclusions, we can derive a comparison table between our work and others based on main factors like used data, number of missed tasks and real time issue. Table I depicts this comparison.

TABLE I Comparison study

Study	Google Traces	Miss tasks	Real time Issue
Reiss , Charles, et al. [14]	Yes	No	No
Travieso, G. et al [15]	No	No	No
Our work	Yes	Yes	Yes

As a future work, we currently work to propose a novel real time load balancer that will be able to deal and handle real time tasks and have them meet their deadlines as much as possible. Furthermore, we plan to benchmark the proposed algorithm with the ones in this paper and other algorithms that fit our designed simulator in terms of the parameters involved. The dataset we plan to use is the same one in this work since this dataset is well understood and explained in this paper.

References

[1] P. Bertino, Martino and Squicciarini, *Security for Web Services and Service-Oriented Architectures.* Springer, 2010.

[2] Z. Scully, G. Blelloch, M. Harchol-Balter, and A. Scheller-Wolf, "Optimally scheduling jobs with multiple tasks," *ACM SIGMETRICS Performance Evaluation Review,* vol. 45, no. 2, pp. 36–38, 2017.

[3] D. E. Eisenbud, C. Yi, C. Contavalli, C. Smith, R. Kononov, E. Mann-Hielscher, A. Cilingiroglu, B. Cheyney, W. Shang, and J. D. Hosein, "Maglev: A fast and reliable software network load balancer." in *NSDI*, 2016, pp. 523–535

[4] G. C. Platform, "Web app arch. diagram" 2018. Available: s://cloud.google.com/solutions/architecture/webapp

[5] D. Liu, N. Guan, J. Spasic, G. Chen, S. Liu, T. Stefanov, and W. Yi, "Scheduling analysis of imprecise mixed-criticality real-time tasks," *IEEE Transactions on Computers,* 2018

[6] M. Lee, A. S. Krishnakumar, P. Krishnan, N. Singh, and S. Yajnik "Supporting soft real-time tasks in the xen hypervisor," in *Proceedings of the 6th ACM SIGPLAN/SIGOPS International Conference on Virtual*

Execution Environments, ser. VEE '10. New York, NY, USA: ACM, 2010, pp. 97–108. [Online]. Available: ://doi.acm.org/10.1145/1735997.1736012

[7] K. Al Nuaimi, N. Mohamed, M. Al Nuaimi, and J. Al-Jaroodi, "A survey of load balancing in cloud computing: Challenges and algorithms," in *Network Cloud Computing and Applications (NCCA), 2012 Second Symposium on.* IEEE, 2012, pp. 137– 142.

[8] O. Beaumont, L. Eyraud-Dubois, and J.-A. Lorenzo-del Castillo, "Analyzing real cluster data for formulating allocation algorithms in cloud platforms," *Parallel Computing*, vol. 54, pp. 83–96, 2016.

[9] M. Adhikari and T. Amgoth, "Heuristic-based load-balancing algorithm for iaas cloud," *Future Generation Computer Systems,* vol. 81, pp. 156–165, 2018.

[10] A. Dave, B. Patel, and G. Bhatt, "Load balancing in cloud computing using optimization techniques: A study," *in Communication and Electronics Systems (ICCES), International Conference* on. IEEE, 2016, pp. 1–6.

[11] K. Dasgupta, B. Mandal, P. Dutta, J. K. Mandal, and S. Dam, "A genetic algorithm (ga) based load balancing strategy for cloud computing," *Procedia Technology*, vol. 10, pp. 340–347, 2013.

[12] Y. Wu, X. Song, and G. Gong, "Real-time load balancing scheduling algorithm for periodic simulation models," *Simulation Modelling Practice and Theory*, vol. 52, pp. 123–134, 2015.

[13] C. Reiss, J. Wilkes, and J. L. Hellerstein, "Google cluster-usage traces: format+ schema," Google *Inc., White Paper,* pp. 1–14, 2011.

[14] Reiss , Charles, et al. "Heterogeneity and dynamicity of clouds at scale: Google trace analysis." Proceedings of the Third ACM Symposium on Cloud Computing. ACM, 2012.

[15] Travieso, G., Ruggiero, C. A., Bruno, O. M., & da Fontoura Costa, L., "A complex network approach to cloud computing", *Journal of Statistical Mechanics: Theory and Experiment*, 2016(2), 023402.