Optimization of Power Consumption in Cloud Data Centers Using Green Networking Techniques

Dr. Qutaiba I. Ali — Email: qut1974@gmail.com

Alnawar J. Mohammed — Email: alnawar_j25758@yahoo.com

Computer Engineering Dept., College of Engineering, University of Mosul, Mosul, Iraq.

Abstract

In this paper, a neuro-based predictor is proposed with a prediction algorithm to estimate the required number of active servers simulating the Green Networking objectives. The inputs of such predictor are the CPU utilization of the servers in the data center and the variations of the incoming demands with the number of users' variation. During the work, different demand profiles of ClarkNet traffic traces are simulated on OPNET14.5 Modeler to obtain the required training values of servers' CPU utilization and clients' throughput. Also, Green Networking objectives are defined to maintain the Power Management Criteria (PMC) which guaranteed that all CPU utilization must be greater than 30%. Taking into account that a maximum number of 100 servers are used in such local data center, an ON/OFF control algorithm is then suggested for the power management of different servers in data center to fulfill the previous Green objectives. The Power saving is finally evaluated since it has been noticed that the power saving percentage can be increased from 17.33% to 85.33% of a total power of 75 k watts when the number of the operating servers is decreased from 80% to 5% of the overall servers.

Keywords: Artificial Neural Network (ANN), Cloud Computing, Data Centers, Green Networking, Power Consumption.

الإستهلاك الأمثل للقدرة في مراكز البيانات السحابيه بإستخدام تقنيات الشبكات الخضراء

د. قتيبة ابراهيم علي — Email: qut1974@gmail.com

النوار جاسم محمد — Email: alnawar_j25758@yahoo.com

قسم هندسة الحاسوب، كلية الهندسة، جامعة الموصل،الموصل، العراق.

الخلاصة

في هذا البحث، تم اقتراح متنبيء قائم على استخدام الشبكة العصبية مع خوارزمية لتقدير العدد المطلوب من الخوادم النشطة محاكية أهداف الشبكات الصديقة للبيئة (الخضراء). مدخلات هذا المتنبيء هي نسبة استغلال وحدة المعالجة المركزية للخوادم في مركز البيانات، بالاضافة الى التغير في مقادير الطلبات الواردة من عدد من المستخدمين. أثناء العمل، تم تسليط نماذج مختلفة من الطلبات الواردة الى خوادم ClarkNet، و تم بناء نماذج مختلفة لشبكة مراكز البيانات باستخدام برنامج المحاكاة OPNET Modeler 14.5 للحصول على القيم المطلوبة للتدريب وهي نسبة الاستغلال لوحدة المعالجة المركزية للخودام ونسبة الحمل المسلط من قبل العملاء. أيضا تم تعريف أهداف الشبكات الخضراء بالحفاظ على معايير ادارة الطاقة المتمثلة بأن تكون قيم نسبة استغلال وحدة المعالجة المركزية للخوادم أكبر من 30٪. بعد الأخذ بعين الاعتبار أن الحد الأقصى لعدد الخوادم في مركز البيانات المقترح في العمل هو 100 خادم ، اقترحت خوارزمية ON/OFF لإدارة استخدام الطاقة من قبل خوادم مختلفة في مركز البيانات لتحقيق أهداف الشبكة الخضراء السابقة الذكر. أخيراً تم تقييم معدل تقليل صرف الطاقة حيث من الملاحظ أن نسبة تقليل صرف الطاقة ازدادت من 17.33% الى 85.33% من مقدار صرف الطاقة الكلي المقدر بحوالي 75 كيلوواط عندما يقل عدد الخوادم التي تكون في حالة عمل من 80% الى 5% من عدد الخوادم الكلي.

Received: 2 – 10 - 2013

Accepted: 9 – 6 - 2013

1. Introduction

In recent years, various IT service providers (such as IBM, Microsoft, Google and other similar large organizations) have deployed data centers for the provision of Cloud Computing in addition to the hostage of hosting the internet applications and scientific researches [1]. Typically, such data centers have sizes of the order of thousands of servers and switches. Many in-depth studies related to data center's energy consumptions have been prompted by the mushroom growth of such data centers and expansion of the existing ones [2], [3]. A recent report [4] has estimated that the data centers in the US consumed approximately 1.5% of the total electricity consumption in 2006, and this number is projected to more than double nowadays.

Fig. (1) shows the power consumption distributions within a typical Data Center [5].

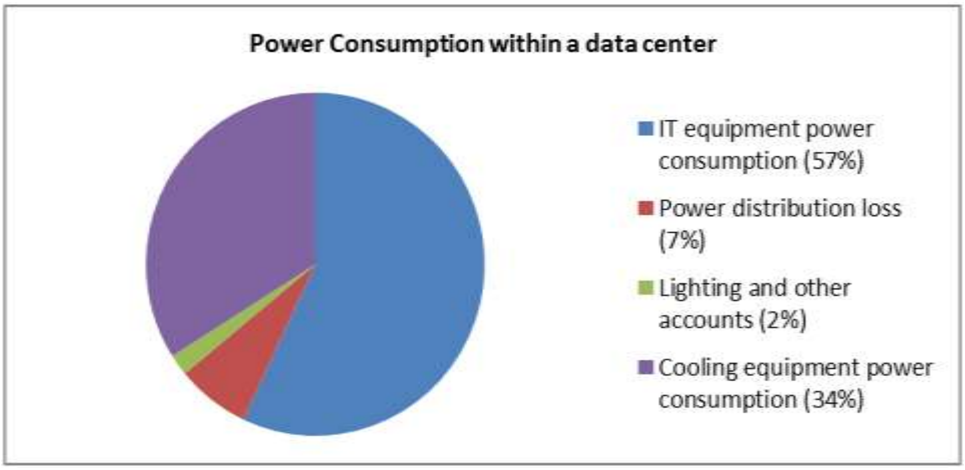


Fig. (1): Distribution of Power Consumption within a Data Center.

On other hand Cloud Computing, which refers to the concept of dynamically provisioning processing time and storage space from a ubiquitous "cloud" of computational resources, allows users to acquire and release the resources on-demand [6], [7]. It also provides access to data from processing elements, while relegating the physical location and exact parameters of the resources. From the user point of view, Cloud Computing means scalability on-demand, flexibility to meet business changes and easy to use and manage [8],[9].

Since, data centers are often designed to sustain peak load and low conditions, they will become under-utilized in normal operation, leaving a large room for power savings. A worthy research effort have been dedicated to reduce unnecessary power expenses. This is usually called as Green Networking technologies and protocols [10]. Green Networking is the practice of choosing energy-efficient products and networking technologies, with possible minimization of the use of energy resources [11]. Green Networking can cover all aspects of the network (such as personal computers, peripherals, switches, routers, and communication media). The optimization of power efficiencies of all network components shares a significant impact on the overall power consumption by these components. Consequently, such optimized efficiencies gained by having a Green Networking will reduce CO2 emissions and thus will help mitigate global warming [5]. Generally speaking, the goal of Green Cloud Computing is to integrate management of computing devices and environmental for control mechanisms in order to provide quality of service, robustness, and power efficiency. So, the challenge in Green Cloud Computing is the minimization of resource usage while still satisfying quality of service requirements and robustness [12].

Among many attempts to investigate the problem of power managements for data centers, Raghavndra, et al. [13] in 2008, proposed a method for solving such problem by combining and coordinating five different policies. A feedback control loop was applied to coordinate the controllers' actions. They claimed that the approach is independent of the workload type. However the CPU management was the only part which was dealt within that system, such approach felt to support both strict and variable Service Level Agreements (SLAs) and couldn't be applied for Cloud Computing providers. In 2009, Kusic et al. [14] defined the same problem in virtualized heterogeneous environments as a sequential optimization. The resource providers' gain was maximized by minimizing both power consumption and SLA violation. The drawbacks of such model lie in its needs of simulation based learning for the application specific adjustment and in its complexity. Also in 2009, the problem of request scheduling for multi-tiered web applications was studied by Srikantaiah, et al. [15] in virtualized heterogeneous system in order to minimize power consumption. Their results showed the "U" shaped variation of the power consumption with the utilization highlighting the optimal utilization point. However, such approach was a workload-dependent and application-dependent, so it is not suitable for a generic Cloud environment.

Recently, in 2010 the power consumption implications of data centers' network architecture were investigated by Gyarmati and Trinh [16] with the limitation that the optimization of network architecture cannot be applied dynamically after the design time. Also in 2010, Beloglazov and Buyya [17] proposed an energy efficient resource management model for visualized Cloud data centers. Such model reduced operational cost and provided the required Quality of Service (QoS). However, the decentralized architecture of the energy aware resource management system for Cloud data centers lacked the possibility of optimization over multiple resources. Again in 2010, Duy et al. [18] designed and implemented a Green scheduling algorithm using a neural network predictor. The target was to optimize the server power consumption in Cloud Computing. Future load demands were predicted based on historical values. The algorithm succeeded in minimizing the number of running servers via some ON/OFF control to turn off unused servers and restart them. However, the limitation of such system model was its inability to deal with diverse workloads and application services. A more accurate ON/OFF algorithm based on that of [18] was also presented by Duy et al. [7] in 2011. Also in 2011, a dynamic threshold-based approach for CPU utilization for the dynamic and unpredictable workload of the Cloud was introduced by Sinha et al. [19] in an attempt to avoid unnecessary power consumption. Such approach met some energy efficiency requirement ensuring QoS to the users by minimizing the SLA violation. Nevertheless, such technique had not been investigated on real Cloud setup to evaluate its exact performance.

More recently in 2012, Beloglazob et al. [20] conducted a survey of research in energy-efficient computing and proposed architectural principles for energy-efficient management of clouds and energy-aware resource allocation polices. Although such approach put a strong trust on open challenges in order to enhance the energy-efficient management of Cloud Computing environments, but the used optimization algorithm was slow due to complex computations. Also in 2012, Kansal and Chana [21] presented a step towards Green computing by Cloud load balancing techniques to minimize the resource computations which may further reduce energy consumption. They discussed different load balancing techniques in Cloud Computing and compared them. They also addresses the need to develop an energy-efficient load balancing techniques that can improve the performance of Cloud Computing along with maximum resource utilization; the idea which is dealt with in this paper. Again in 2012, Werner et al. [12], introduced a system management model with an integrated solution for environment services and network management based on organization model of autonomous agent components. However, the use of inaccurate feedback model provided only 8% SLA error decrease.

Data centers are usually plagued with thousands of servers to perform the processing for businesses. They also serve end users to facilitate and accomplish large business goals. The need for large and complex data centers is a must because of the continuous increase in businesses especially e-businesses. The problem with most of these data centers is that almost 90% of the servers remain idle most of the time and left performing nothing but consuming huge power [22]. Such consumption is increasing day by day as the demands from end users increase. No mechanism is available to measure the performance of already installed servers in order to increase utilization ratio, resulting in low power consumption. In many countries, it is very difficult to provide enough energy resources to the ever growing data centers, resulting in overall degradation of economy. Such energy crisis causes the IT industrial output to fall precipitously. Therefore, it preferred to have some strategies or frameworks for large data centers so that they can maintain the ever growing demands from businesses and end users and thus undertake the effects of the increasing energy costs. The implementation of some mechanism that optimally or properly utilizes server resources is a strong need to reduce the power consumption by the eliminating of the idle servers. Some metrics can also be developed to benchmark the performance of data centers so that their power consumption can be measured.

In addition to this introductory section of this paper, section 2 contains the method for optimizing the power consumption in terms of CPU utilization of different servers. System modeling is described in section 3. Simulation results are given in section 4 with some discussions. Finally, section 5 concludes this paper.

2. Optimizing Power Consumption

The management of power consumption in data centers may lead to some improvements in energy efficiency [7]. Infrastructure of Cloud Computing is housed in data centers and can be benefited significantly from these advances. Although, sleep scheduling and virtualization of computing resources are some basic techniques in Cloud Computing data centers used to improve the energy efficiency of Cloud Computing, it is important to further minimize such power consumption in data centers that host Cloud Computing services. Many researches on power consumption in Cloud Computing have focused only on the energy consumed in the data center. However, to get a clear picture of the total energy consumption of a Cloud Computing service, a more comprehensive analysis is required [9]. Power consumption in data centers can be determined by different system resources (such as CPU, memory, disk storage and network interfaces). Among all system resources, the CPU consumes the main part of power, and hence this paper focuses on managing its power consumption with efficient usage. In addition, the CPU utilization is typically proportional to the overall system load [20]. It should be noted that understanding the relationship between power consumption and CPU utilization of servers is essential in designing efficient strategies for power savings [7].

Recently, several researches have shown that, on average, an idle server consumes about 70% of the power consumed by the server running at the full CPU speed [13],[14]. That is why the technique of switching idle servers to the OFF state is adopted in this paper to reduce the total power consumption. The power model is defined by the following equation:

$$P(u) = k \cdot P_{max} + (1 - k) \cdot P_{max} \cdot u, \quad (1)$$

Where *Pmax* is the maximum power consumed when the server is fully utilized; *k* is the fraction of power consumed by the idle server (i.e. 70%); and u is the CPU utilization. For our experiments the maximum power consumed by a server is set to 250 W, which is a usual value for modern servers [20].

3. System Modeling

The general framework of the system design of this work consist of the following parts that shown in Fig. (2).

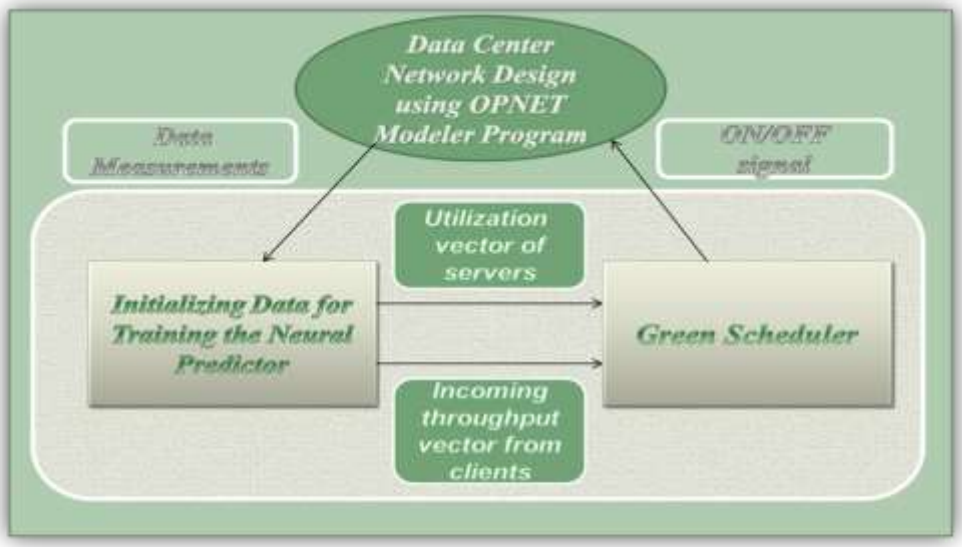

Data Center Network Design using OPNET Modeler Program
Data Measurements
ON/OFF signal
Utilization vector of servers
Initializing Data for Training the Neural Predictor
Green Scheduler
Incoming throughput vector from clients

Fig. (2): The General Framework.

In the simulation model, a local data center is assumed with 100 servers and a maximum number of end users in a LAN network that send requests to the servers of data centers are 300 users. At the beginning, it has been assumed that a normal range for CPU utilization to operate in a Green mode is to be greater than 30%, in this range different number of users and different numbers of servers are examined with the data center topology shown in Fig. (3).

In the proposed Data Center network many components and devices are included as follows:

1- LAN Network: contains the facility to configure different number of clients, therefore different values of incoming throughput or load requests can be obtained.

2- Load Balancer: One of the central issues in Cloud Computing is load balancing. It is a mechanism that distributes the dynamic local workload evenly across all the servers in the whole Cloud. Such distribution can avoid a situation where some servers are heavily loaded while others are idle or even doing little work. High user satisfaction and perfect resource utilization can then be achieved to improve the overall performance of the system, thereby minimizing the resource consumption. Thus load balancer can help in implementing fail-over, enabling scalability, avoiding bottlenecks and over-provisioning, reducing response time,
etc … [21].

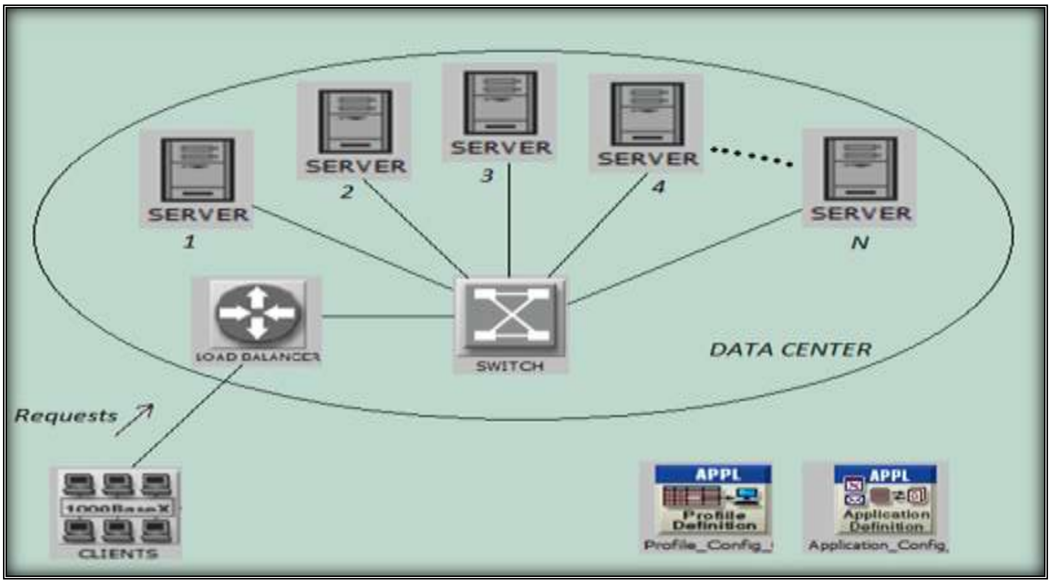


Fig. (3): Data Center Network.

Few existing load balancing algorithms has been presented in the followings [23],[24]:

Server Load (the adopted method): Server load algorithm is load balancing of traffic among servers. This algorithm has the ability of determining the servers that should not receive any new connections because they are offline or fully utilized. The connections are then load balanced using a simple load algorithm which guaranteed that each server receives an even load (or close to) of incoming requests.

a. *Token Routing:* The minimization of the system cost by moving the tokens around the system is the main objective of this algorithm. Actually, in a scalable Cloud system, agents lack enough information of distributing the work load due to communication bottleneck. So the workload distribution among the agents is not fixed. Heuristic approach of token based load balancing helps in removing the drawback of tis algorithm. This algorithm can provide the fast and efficient routing decision.

b. *Round Robin:* In this algorithm, the processes are divided between all processors. Each process is assigned to the processor in a round robin order. The process allocation order is locally maintained independent of the allocations of remote processors. Despite that the workload distributions between processors are equal, the job processing time for different processes are different. So, some nodes may be heavily loaded and others remain idle, simultaneously. This algorithm is mostly used in web servers where similar nature HTTP requests are distributed equally.

c. *Randomized:* Randomized algorithm is of static type in nature. In this algorithm, a process can be handled by a particular node n with a probability p. The process allocation order is also maintained for each processor independent of the allocations of remote processor. This algorithm works well in case where processes are equally loaded. However, when loads are of different computational complexities the problem arises. Randomized algorithm works well when Round Robin algorithm generates overhead for process queue.

d. *Central queuing:* This algorithm works on the principal of dynamic distribution. Where, each new activity arriving at the queue manager is inserted into the queue. When a request is received by the queue manager it removes the first activity from the queue and sends it to the requester. If no ready activity is present in the queue the request is buffered, until a new activity is available. But in case new activity comes to the queue while there are unanswered requests in the queue the first such request is removed from the queue and new activity is assigned to it.

e. *Least Connection mechanism:* Least connection mechanism is a part of dynamic scheduling algorithm. It needs to count the number of connections for each server dynamically for load estimation. The load balancer records the connection number of each server. The number of connections increases when a new connection is dispatched to it, and decreases when connection finishes or timeout happens.

Load balancing is also required to achieve Green computing in clouds which can be done with the help of the following two factors:

- *Reducing Energy Consumption* - Load balancing helps in avoiding overheating by balancing the workload across all the nodes of a cloud, hence reducing the amount of power consumption.
- *Reducing Carbon Emission* - Energy consumption and carbon emission go hand in hand. The less the power consumed, the smaller is the carbon emission. This helps in achieving Green computing.

1- Servers: They represent the most important elements in the data center as a result of their significant power consumption, especially in times where they are idle. In this research, it is considered that all the servers are identical to ensure that they receive the same load from load balancer according to server load algorithm and thus the servers are equally utilized.

2- Application Definition: it is assumed that this network is loaded by HTTP requests directed to the servers' farm. During all tests the application definition simulates different load traces that are worth of all HTTP requests to the ClarkNet WWW server. ClarkNet is a full Internet access provider for the Metro Baltimore-Washington DC area. Where the trace for one day with 5-minute resolution is plotted in Fig. (4) [25].

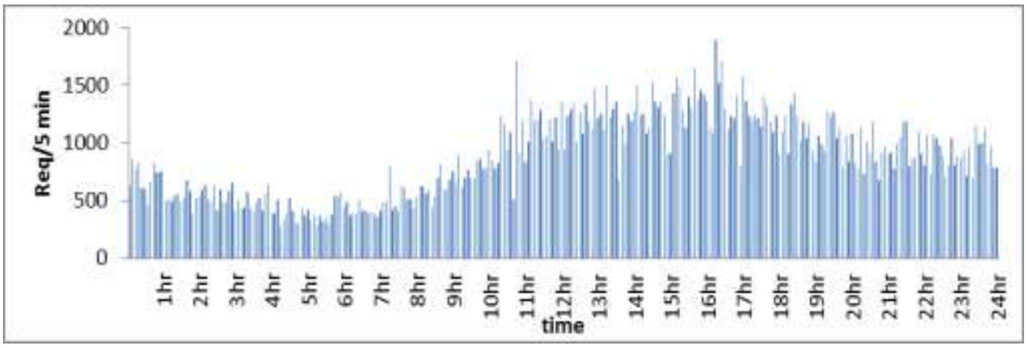


Fig. (4): ClarkNet Trace.

4. Simulation Results

Samples of the CPU utilization of servers measured from OPNET for different cases of study are plotted in Fig. (5).

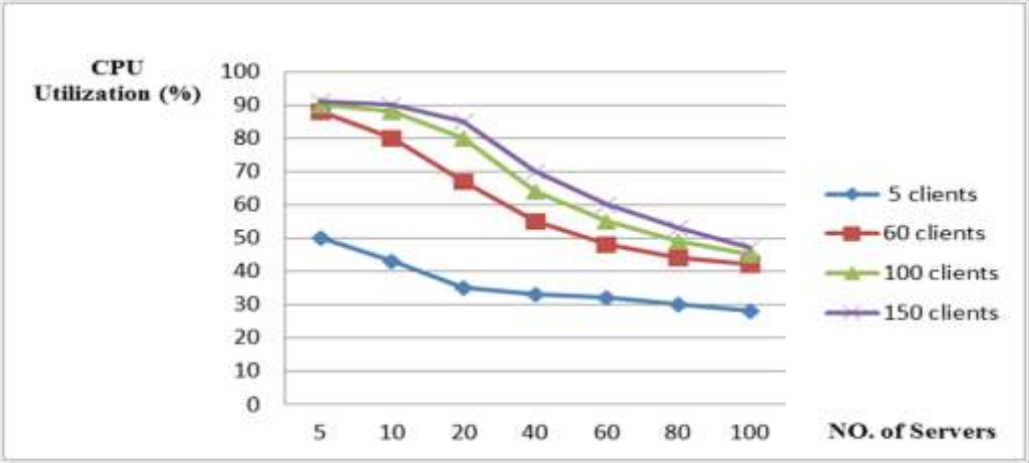


Fig. (5): The variation of CPU utilization with different number of servers for the cases of (5 clients, 60 clients, 100 clients, 150 clients).

4.1 Initializing The Testing Parameters for The Neural Network

The measured data sets from OPNET are initialized before training it with the neural network in Matlab program. The initiation method adapted in this paper is achieved by the normalization (to maximum) of each of the input data sets (CPU utilization of the servers and the incoming throughput from clients), in addition to the output data set (number of servers).

4.2 Green Scheduler

In this sub-section the structure of the green scheduler is presented, focusing primarily on its components and their main functions. As plotted in Fig. (6), the scheduler is composed of three parts in order of execution: the predictor, ON/OFF algorithm, and performance evaluation. The scheduler starts execution by running the predictor to collect the input traffic data and predict the number of active servers. Depending on the predictor output, the ON/OFF controller adjusts server allocation to minimize the power consumption. Finally, the performance is evaluated by the performance evaluation part which calculates the power consumption of the cases under study.

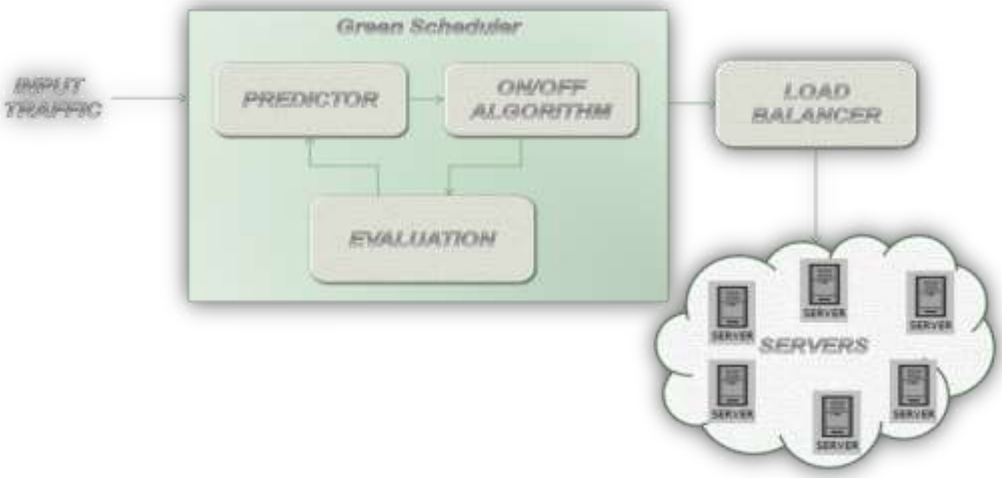


Fig. (6): Green Scheduler structure

4.2.1 ANN predictor

Artificial neural network (ANN), originally developed to simulate biological neural networks, is a computational model consisting a large number of highly interconnected neurons (or nodes). Each neuron receives input signals generated from other neurons or external inputs, processes them locally through an activation function, thus producing an output signal to other neurons or external outputs. ANN can learn the training data with a learning algorithm by forming a mapping between inputs and desired outputs. After finishing the learning process, ANN is able to understand the hidden dependencies between inputs and outputs, in addition to its generalization to a data never before seen [26].

In this paper, multi-layer feed-forward back-propagation network is chosen for predicting the number of active servers which can be considered as a target. As shown in Fig. (7) The neural network predictor in this work has 2 inputs which are the stable normalized values of both CPU utilization of servers and incoming throughput of clients (Mbps), and 1 output layer with one neuron where the result of the number of active servers is obtained. The inputs and the output layer of the neural network are separated by 3 hidden layers: hidden layer A with 2 neurons, hidden layer B with 4 neurons and hidden layer C with 7 neurons. In the hidden layer (A and B) a tan-sigmoid activation function is used, with a log-sigmoid activation function is used for hidden layer C. The connections between neurons indicate the flow of data from one neuron to the next. Weights are used to modify the connections with a D.C extra input for each neuron.

When the network is run, each neuron in the hidden layers and the output layer performs the calculation in the following equation and transfers the result the next layer:

$$O_c = h\left(\sum_{i=1}^{n} x_{c,i} w_{c,i} + b_c\right) \tag{2}$$

$$\text{where } h(x) = \left\{ \begin{array}{ll} \left(\frac{2}{1+e^{-2x}}\right) - 1 & tan - sigmoid\ activation\ function\ for\ hidden\ layers\ A\ and\ B\ neurons \\ \left(\frac{1}{1+e^{-x}}\right) & log - sigmoid\ activation\ function\ for\ hidden\ layer\ C\ neurons \\ x & for\ output\ layer\ neuron \end{array} \right\}$$

where O_c is the output of the current neuron, n is the number of neurons in the previous layer, $x_{c,i}$ is an input to the current neuron from the previous layer, $w_{c,i}$ is the weight modifying the corresponding connection from $x_{c,i}$, and b_c is the bias.

Fig.(7) shows the proposed neural predictor. While training the network, it was found that the Mean Squared Error (MSE) can converge to the value of *2.00e*$^{-6}$ as shown in Fig. (8).

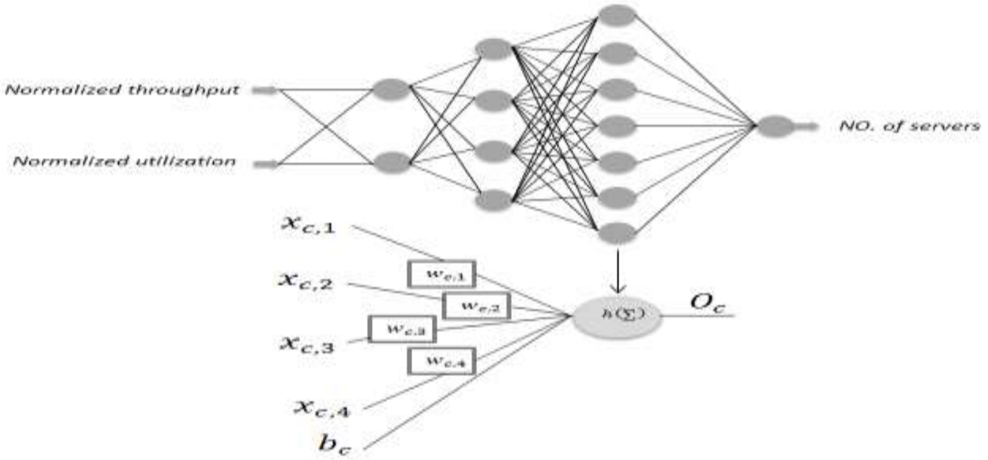


Fig. (7): 2-4-7-1 Neural Predictor.

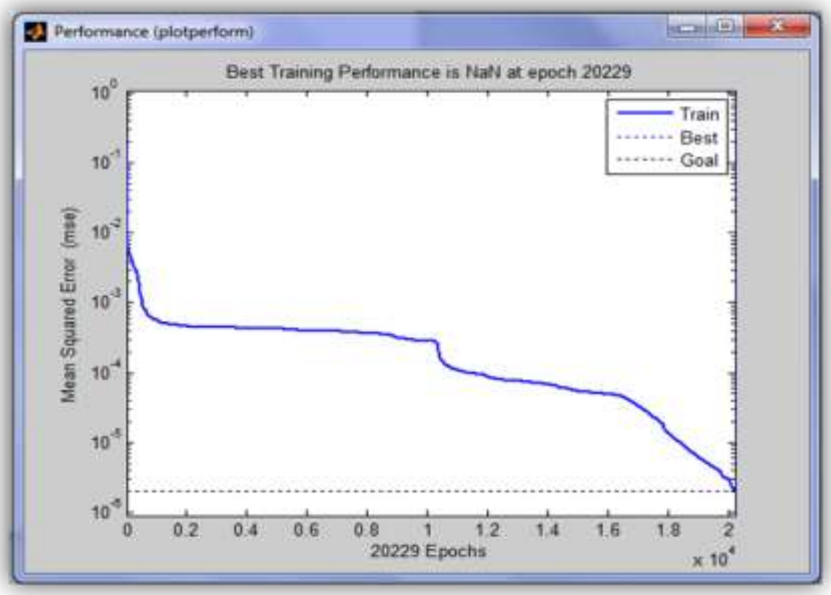


Fig. (8): MSE training performance.

In the testing mode, the time required is 0.94 sec. Table below shows the predicted number of active servers for the corresponding input sets of incoming throughput and CPU utilization of servers.

Table (1): Predicted number of servers for each input data sets.

Clients NO. in LAN	5 Clients							60 Clients						
Incoming throughput	5.2	2.8	2.8	3.3	2.7	2.9	2.8	6	4	8.2	8.5	8.7	8.4	7.6
Utilization of servers (%)	50	43	35	33	32	30	28	88	80	67	55	48	44	42
Predicted NO. of active servers	4.9993	10.0007	19.9981	40	60.0013	79.9996	100	5.128	10.0431	20.0363	39.9997	60.0006	80.0029	100.0002
Predicted NO. of active servers after rounding	5	10	20	40	60	80	100	5	10	20	40	60	80	100
Clients NO. in LAN	100 Clients							150 Clients						
Incoming Throughput	7.5	8	12.5	12.5	11	9.4	8.2	7.5	12.8	14.5	12.9	11.7	9.5	8.7
Utilization of servers (%)	90	88	80	64	55	49	45	91	90	85	70	60	53	47
Predicted NO. of active servers	5.6134	8.4414	20.1097	39.9994	60	79.9974	99.9993	4.4746	9.9688	19.9998	39.9976	59.9997	79.9982	100.0001
Predicted NO. of active servers after rounding	6	10	20	40	60	80	100	4	10	20	40	60	80	100

4.2.2 The ON/OFF algorithm

As it was stated previously the (PMC) for the OFF decision is that the CPU utilization must be greater than 30%. In this work an ON/OFF algorithm is used to determine which servers must be turned ON/OFF according to the changes in throughput from the clients and CPU utilization of servers. A server can be in one of the following five states: OFF, RESTARTING, ON, SHUTTING and IDLE. Initially all servers are in the OFF state, which is actually a selected state to which a server is sent for power savings. Every 5 minutes the controller sends a restart signal for all servers in the data center and reads the number of active servers with the CPU utilization of servers to make the right OFF decision, after this restarting signal, the servers will move from OFF to RESTARTING. They will stay in this mode for T_{REST}, which is set to 20 seconds, before being ON. On the other hand, when a server is signaled to turn-off, it will change state and stay in the SHUTTING state for T_{SHUT}, which is set to 10 seconds, before it completely turns to OFF state. Since all servers in the data center are assumed to be identical, so any set of servers can be turned-off according to two different modes:

1- Prediction mode: based on turning OFF the idle servers after predicting the exact number of active servers that obtained from the neural predictor, therefore, an OFF signal is sent to the idle servers

2- Prediction Green mode: based on turning OFF additional 20% servers from the maximum number of servers in data center in addition to turning OFF the idle servers which result after predicting the exact number of active servers obtained from the neural predictor. In order to assure (PMC) in the case where CPU utilization of the servers is less than 30%. For example as stated in Table (1) for 5 clients in the case of 100 predicted active servers, the CPU utilization of servers value is 28% which is below the PMC, so additional 20 servers must also be turned OFF.

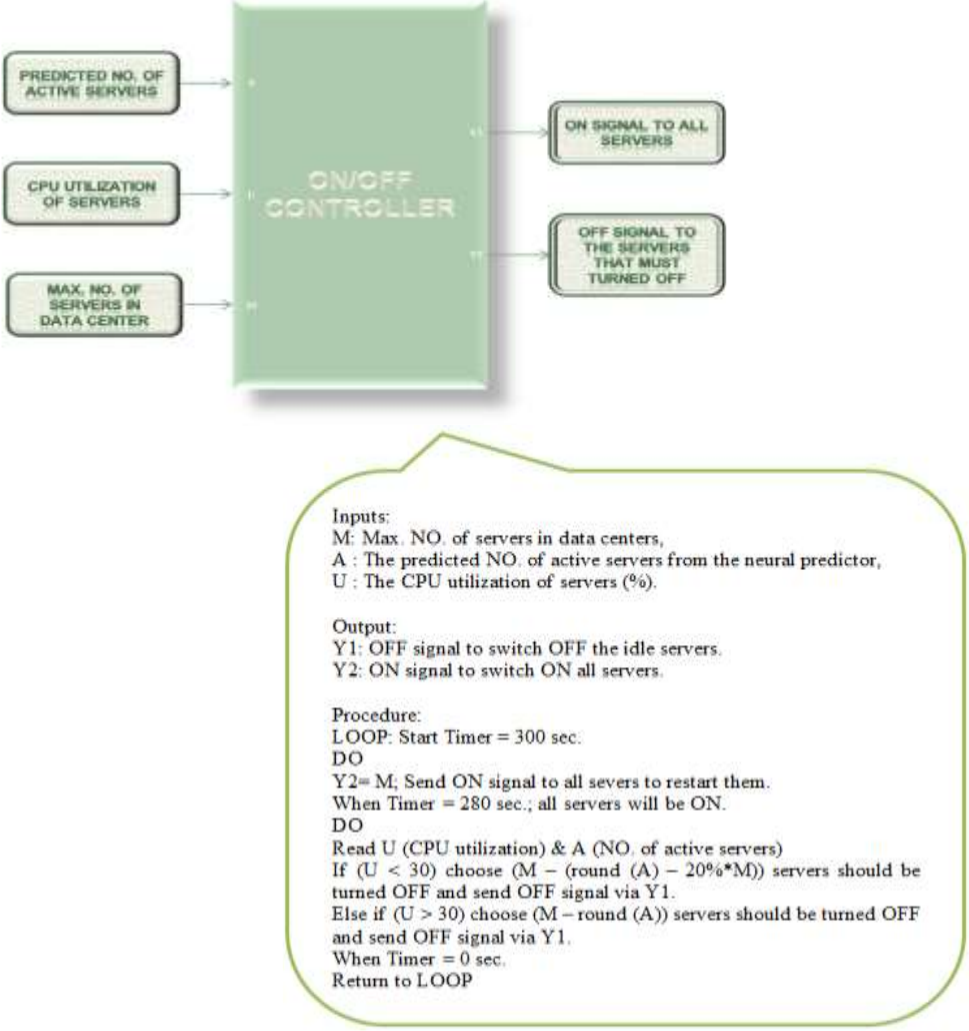
Finally, a feedback with the number of the ON/OFF signal is returned to the load balancer in OPNET which has the facility to select that number of servers to be turned OFF as shown in Fig. (9).

4.2.3 Evaluation mechanism

A performance monitoring mechanism for the adaption of the workload changes can be developed. Such monitoring mechanism must balance adaptability with stability. It should be noted that the neural predictor can be trained with some updated training data to reflect the workload changes. In this context, this mechanism can be used to find the suitable training epochs during execution. There are two training policies which can be proposed; namely, static training and dynamic training. In static training, the training phase takes place at regular time intervals, making it easy to identify training epochs. In spite of load increase and decrease, the training phase can be performed on a daily or hourly basis. The drawback of this policy is that training occurs regardless of performance. In dynamic training, the solution can hold a sliding window moving average of size *S*1, and maintains an error term *MSPE* (Mean Squared Prediction Error). If *MSPE* value of the moving average of recent observations exceeds a predefined error threshold *Err*, the training process will be triggered. Otherwise, the system is said to be stable, and no training is needed. *MSPE* is identified based on equation (3).

$$MSPE = \frac{1}{S1}\sum_{i=t-S1}^{t-1}(P_i - A_i)^2 \qquad \textbf{(3)}$$

where *t* is the current time, *S*1 is the window size, and *Pi* and *Ai* are, respectively, the predicted workload and the actual workload at time *i*. Although we adopt a relatively aggressive performance monitoring mechanism in training policies, prediction errors are unavoidable. To tackle the shortage of servers in case the requested load is more than the capacity of the provided servers, a given number of servers, called additional servers are used. For example, if the predictor predicts 5 servers and 2 additional servers are used, then actually use 7 servers instead of only 5. This additional switched ON servers may be taken into account in the next evaluation.


PREDICTED NO. OF ACTIVE SERVERS
CPU UTILIZATION OF SERVERS
MAX. NO. OF SERVERS IN DATA CENTER
ON/OFF CONTROLLER
ON SIGNAL TO ALL SERVERS
OFF SIGNAL TO THE SERVERS THAT MUST TURNED OFF

```
Inputs:
M: Max. NO. of servers in data centers,
A : The predicted NO. of active servers from the neural predictor,
U : The CPU utilization of servers (%).

Output:
Y1: OFF signal to switch OFF the idle servers.
Y2: ON signal to switch ON all servers.

Procedure:
LOOP: Start Timer = 300 sec.
DO
Y2= M; Send ON signal to all severs to restart them.
When Timer = 280 sec.; all servers will be ON.
DO
Read U (CPU utilization) & A (NO. of active servers)
If (U < 30) choose (M – (round (A) – 20%*M)) servers should be turned OFF and send OFF signal via Y1.
Else if (U > 30) choose (M – round (A)) servers should be turned OFF and send OFF signal via Y1.
When Timer = 0 sec.
Return to LOOP
```

Fig. (9): ON/OFF Controller.

The proposed optimization algorithm is evaluated by calculating the amount of power saving, taking into consideration that the maximum power consumed by a server is set to 250 W, which is a usual value for modern servers [20]. Another condition is taken into account that an idle server consumes approximately 70% of the power consumed by the server running at the full CPU utilization [13],[14]. The power consumption in all servers in addition to that power consumed in the corresponding cooling systems (we assumed that we need one 2.5KW cooling unit for each 5 servers), can be written as follows:

- When leaving the idle servers in operation

$$P_{\text{idle servers in operation}} = 250 * A + 250 * 0.7 * I + \frac{M}{5} * 2.5 * 10^3 \qquad (4)$$

- After turning off the idle servers

$$P_{\text{active servers}} = 250 * A + \frac{A}{5} * 2.5 * 10^3 \qquad (5)$$

- Thus, the power savings can be given by:

$$P_{savings} = P_{\text{idle servers in operation}} - P_{\text{active servers}} \qquad (6)$$

where *A* is the number of active servers, *I* is the number of idle servers and *M=100* is the maximum number of servers in data centers. Fig. (10) shows the variations of the consumed powers of leaving the unutilized servers idle and of the case of shutting down all unutilized servers with the number of active servers in the data center. This figure also shows the amount of power saving variation with the number of such active servers.

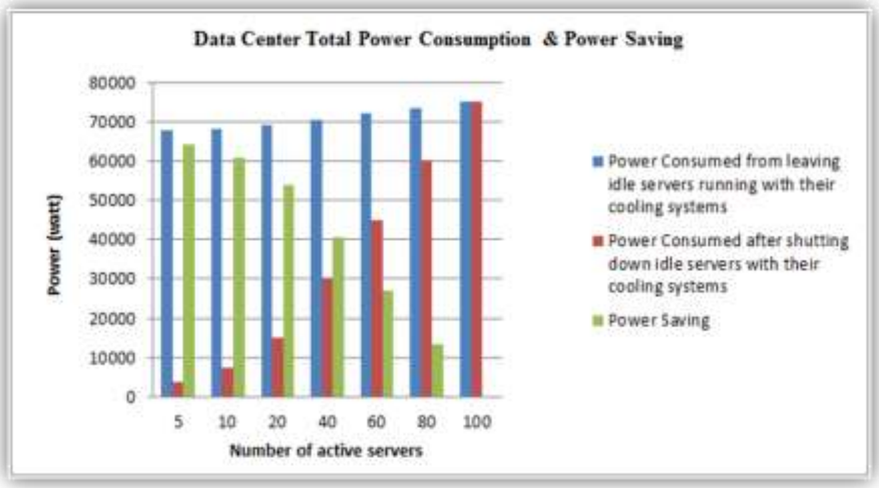


Fig. (10): Power consumption and Power saving in data center servers.

5. Conclusions

In this paper, a neuro-based predictor has been proposed with to find the required number of active servers in a local data center simulating the Green Networking objectives. The CPU utilization of these servers and the variations of the incoming demands have been obtained simulating different demand profiles of ClarkNet on OPNET14.5 Modular. Green Networking objectives have been defined by SLA to maintain all CPU utilization at greater than 30%. The predictor output is the number of active servers, while a maximum number of 100 servers have been adopted in such data center. An ON/OFF control algorithm has been suggested for the power management of different servers to fulfill the predefined Green objectives. The performance of the proposed predictor has been evaluated by the calculation of the power saving for different cases of operation, highlighting a promising Green field.

References

[1] W. Yang, D. Kang, F. Tong, Y.Kim, "*Performance Analysis of Energy Savings according to Traffic Patterns in Ethernet with Rate Adaptation", 12th International Conference on Computer Modelling and Simulation, 2010.

[2] S. K. Garg and R. Buyya, "*Green Cloud Computing and Environmental Sustainability",* www.**cloud**bus.org/.../**Cloud**-Env**Sustainability**2011.

[3] J. Shuja, S. A. Madani, K. Bilal, K. Hayat, S. U. Khan and S. Sarwar, *"Energy-Efficient data centers",* Computing DOI 10.1007/s00607-012-0211-2, Springer, Sept. 2, 2012.

[4] EPA Datacenter Report Congress, *http://www.energystar.gov/ia/partners/prod development/downloads/EPA Datacenter Report Congress Final1.pdf*, accessed Jun. 26, 2010.

[5] N. Chilamkurti, S. Zeadally, and F. Mentiplay, *"Green Networking for Major Components of Information Communication Technology Systems",* EURASIP Journal on Wireless Communications and Networking, Volume 2009, Article ID 656785, 7 pages.

[6] A. Orgerie and L. Lefèvre, *"When Clouds become Green: the Green Open Cloud Architecture",* Parallel Computing 19 (2010), 228-237.

[7] T.V.T. Duy, Y. Sato, and Y. Inoguchi, ”*A Prediction-Based Green Scheduler for Datacenters in Clouds”,* The Institute of Electronics, Information and Communication Engineers (IEICE Trans.), 2011.

[8] L. Liu, H. Wang, X. Liu, X. Jin, W. He, Q. Wang and Y. Chen, “*GreenCloud: A New Architecture for Green Data Center”,* ICAC-INDST’09, June 16, 2009, Barcelona, Spain.

[9] J. Baliga, R. A. Ayre, K. Hinton and R. S. Tucker, “*Green Cloud Computing: Balancing Energy in Processing, Storage, and Transport”,* Proceedings of the IEEE, Vol. 99, No. 1, Jan. 2011.

[10] A. P. Bianzino, C. Chaudet, D. Rossi and J. Rougier, “*A Survey of Green Networking Research”,* arXiv:1010.3880v1 [cs.NI], 19 Oct. 2010.

[11] E. Altman, C. Hasan, J.Gorce and L. Roullet*, “Green Networking: Downlink Considerations”,* in Proc. International conference on NETwork Games, COntrol and Optimization (NetGCoop), 2011, pp.1-4.

[12] J. Werner, G. A. Geronimo, C. B. Westphall, F. L. Koch, R. R. Freitas and C. M. Westphall, *“Environment, Services and Network Management for Green Clouds”,* CLEI Electronic Journal, Volume 15, Number 2, Paper 2, Aug. 2012.

[13] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang and X. Zhu, *“No Power Struggles: Coordinated Multi-Level Power Management for the Data Center”*, SIGARCH Computer Architecture News 36 (1) (2008) 48–59.

[14] D. Kusic, J.O. Kephart, J.E. Hanson, N. Kandasamy and G. Jiang, *“Power and Performance Management of Virtualized Computing Environments via Lookahead Control”*, Cluster Computing 12 (1) (2009) 1-15.

[15] S. Srikantaiah, A. Kansal and F. Zhao, *“Energy Aware Consolidation for Cloud Computing”*, Cluster Computing 12 (2009) 1-15.

[16] L. Gyarmati and T. Trinh, *“How Can Architecture Help to Reduce Energy Consumption in Data Center Networking?”* in: Proceedings of the 1st ACM International Conference on Energy-Efficient Computing and Networking, e-Energy 2010, Passau, Germany, 2010,
pp. 183-186.

[17] A. Beloglazov and R. Buyya, “*Energy Efficiency Resource Management in Virtualized Cloud Data Centers”*, 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, 2010.

[18] T.V.T. Duy, Y. Sato, and Y. Inoguchi, “*Performance Evaluation of a Green Scheduling Algorithm for Energy Savings in Cloud Computing”*, Proc. 24th IEEE International Parallel and Distributed Processing Symposium (The 6th Workshop on High-Performance, Power-Aware Computing), pp. 1-8, Apr. 2010.

[19] R. Sinha, N. Purohit, H. Diwanji, “*Energy Efficient Dynamic Integration of Thresholds for Migration at Cloud Data Centers”,* Special Issue of International Journal of Computer Applications (0975 – 8887) on Communication and Networks, No.11. Dec. 2011, www.ijcaonline.org.

[20] A. Beloglazov, J. Abawajy and R. Buyya, *“Energy-Aware Resource Allocation Heuristics for Efficient Management of Data Centers for Cloud Computing”*, Future Generation Computer Systems 28 (2012) 755-768, journal homepage: www.elsevier.com/locate/fgcs.

[21] N. J. Kansal and I. Chana, *“Cloud Load Balancing Techniques: A Step Towards Green Computing”,* IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No 1, January 2012, ISSN (Online): 1694-0814, www.IJCSI.org.

[22] M. Uddin and A. Abdul Rahman, *"Energy Efficiency and Low Carbon Enabler Green IT Framework for Data Centers Considering Green Metrics",* Renewable and Sustainable Energy Reviews 16 (2012) 4078-4094, journal homepage: www.elsevier.com/locate/rser.

[23] S. Ray and A. De Sarkar, *"Execution Analysis of Load Balancing Algorithms in Cloud Computing Environment",* International Journal on Cloud Computing: Services and Architecture (IJCCSA), Vol. 2, No. 5, October 2012.

[24] M. Arregoces and M. Portolani, *"Data Center Fundamentals"*, Cisco Press, USA, 2004.

[25] Traces in the Internet Traffic Archive, http://ita.ee.lbl.gov/html/traces.html.

[26] T.V.T. Duy, Y. Sato, and Y. Inoguchi, *"Improving Accuracy of Host Load Predictions on Computational Grids by Artificial Neural Networks",* International Journal of Parallel, Emergent and Distributed Systems, 26(4): 275-290, 2010.

The work was carried out at the college of Engineering. University of Mosul