

# Nature Inspired Algorithms for Load Balancing in Cloud Computing

Sebagenzi Jason  
Department of Computer Science  
Jain University

Suchithra R., PhD  
Department MSc IT  
Jain University

## ABSTRACT

Load balancing and Consolidation of Virtual Machines is a way which is effective to improve the utilization of resources and energy efficiency in Cloud data centers. Determining when it is best to reallocate Virtual Machines from an overloaded host is an aspect of dynamic Virtual Machine consolidation that directly influences the utilization of resource and Quality of Service which the system is delivering [1]. The influence on the Quality of Service is explained by the fact that server overloads cause the shortage of resources and the degradation of applications performance. The current solutions to the problem of host overload detection are generally relying on statistical analysis guided by nature inspired in order to find the optimal solution. The limitations of these techniques are that they lead to sub-optimal results and do not allow explicit specification of a Quality of Service goal. We propose a new approach that for any stationary workload which is known and a given state configuration solves the problem of detection of host overload by maximizing the mean inter-migration time under the specified Quality of Service goal optimally [2]. Through simulations with real-world workload traces from more than a thousand Virtual Machines, we show that our approach outperforms the best benchmark algorithm and provides almost 88% of the performance of the optimal offline algorithm.

## Keywords

Cloud computing, Distributed systems, dynamic consolidation, virtualization, host overload detection and energy efficiency.

## 1. INTRODUCTION

Cloud computing has developed the Information Communication Technology industry by enabling on-demand provisioning, based on model pay-as-you-go of computing resources. To avoiding high payment in a private computing infrastructure and consequent maintenance high prices, any company can outsource the computational to the cloud.

However, the problem of data center is high energy consumption, which has increased in back days by 56 percent from 2005 to 2010, and in 2010 was approximatively between 1.1 and 1.5 percent by considering the global electricity use.

Apart from high prices, there is a product of carbon dioxide (CO<sub>2</sub>) emissions, which are predicted to be 2 percent of the global emissions [12]. This problem has been solved not completely by the physical infrastructure in modern data centers [13]. As shown by the Open Compute Project, data center used a Power Usage Effectiveness (PUE) of 1.08, which explains that almost 93 percent of the energy consumed by data centers is consumed by the computing resources.

For that reason, it is important to pay attention on the computer resource management aspect in order to be sure that

the resources of computer are used efficiently to serve applications [18], [4], [15] and [8].

However, one method for fixing the use of resources of data center, which has been demonstrated to be efficient, is consolidation of Virtual Machines.

This area leverages the nature of cloud workloads: Using live migration, the Virtual Machines are reallocated periodically according to the demand of resource for minimizing the number of physical servers, referred to as hosts, required to handle the workload.

The idle machines are switched to low power, in order to reduce the overall energy consumption and eliminate the static power. The hosts are reactivated when the demand of the resource increases.

By this we have two objectives by using this technique: the minimization of the consumption of the energy and the maximization of the quality of service delivered by the system, which form an energy-performance trade off.

The definition of the Quality of Service requirements is formalized in of the service level agreements in terms of a variety of metrics. For specification of the Quality of Service requirements, the application of a modification of the workload independent metric proposed in our previous work is mandatory.

Therefore, the problem transforms into minimization of energy consumption under Quality of Service constraints. This problem is very difficult to be solved in analytically way as a whole, as just the Virtual Machine placement, which is a part of dynamic VM consolidation, is an NP-hard problem [4], [11] and [16].

However, several types of the problem have to be used, related to the heterogeneity of physical resources and Virtual Machines; nonstationary and which the workload is unknown, as seen in Infrastructure as Service environments; performance and power costs of Virtual Machine migrations; and the largest size of cloud data center infrastructures.

Another argument for splitting the problem is decentralization of the management of resources, which is very important for fixing the resource management system for efficient handling of thousands of servers.

Flower pollination algorithm is bio-inspired algorithm which is latest one and has been proposed by Xin-She Yang in 2012. It is inspired by the process of fertilization of flowers. In Flower Pollination Algorithm, self-pollination and abiotic are considered as local pollination when cross pollination biotic is considered for the global pollination between the flower plants.

The algorithm must maintain a balance between local and global pollination. In order to optimizing the benchmark functions, Yang assumed that each plant can have only one

flower and each flower can have only one pollen grain. The action of pollination is done by pollinators like wind, flies, and insects. Thus, each flower (or pollen) can be considered as a solution of an objective function. The objective function finds the best flower, which is able of doing maximum pollination. This process continues unless stopping criteria is found.

Therefore, to make the problem of Virtual Machine consolidation attractive and provide decentralization, in our previous work [1] we have proposed its division into four sub problems:

1. Determine when a host is considered to be overloaded, so that some Virtual Machines should be migrated from it to other hosts to meet the Quality of Service requirements.
2. Deciding when a host is considered to be under loaded, so that its Virtual Machines should be migrated, and the host will switched to a low power mode.
3. Selecting Virtual Machines to migrate from an overloaded host.
4. Allocating the Virtual Machines selected for migration to other active or reactivated hosts.

## 2. THEORY BACKGROUND

In this part, is shown that to improve the quality of Virtual Machine consolidation, it is necessary to maximize the time intervals between Virtual Machine migrations from overloaded hosts [14], [7], [8] and [9]. Since Virtual Machine consolidation is applied to reduce the number of active physical hosts, the quality of Virtual Machine consolidation is inversely proportional to B, the mean number of active hosts over n time steps:

$$B = \frac{1}{n} \sum_{i=1}^n ai$$

ai is the number of hosts which are active at the time step

i = 1, 2, 3, 4... n. A lower value of B represents a better quality of Virtual Machine consolidation.

To analyze the impact of decisions made by host overload detection algorithms on the quality of Virtual Machine consolidation, we consider an experiment, at any time where step the overload host detection algorithm can initiate a migration from a host due to an overload. Two possible consequences of a decision to migrate a Virtual Machine relevant to host overload detection:

Case 1, when a Virtual Machine to be migrated from an overloaded host cannot be placed on another active node due to insufficient resources, and therefore, a new host has to be activated to accommodate the Virtual Machine; and Case 2, when a Virtual Machine to be migrated can be placed on another active host.

### Objective of host overload detection

To analyze host overload detection in isolation, we suppose that no hosts are switched off during the experiment, that is., once a host is activated, it remains active until n.

Let  $p_r$  be the probability of Case 1, i.e., an extra node has to be activated to migrate a Virtual Machine from an overloaded host determined by the machine overload detection algorithm [13] and [14]. Then, the probability of Case 2 is  $(1 - p)$ .

Let T be a random variable showing the time between two subsequent Virtual Machine migrations initiated by the machine overload detection algorithm [15] and [16]. The expected number of Virtual Machine migrations initiated by the host overload detection algorithm over n time steps is  $m/E[T]$ , where  $E[T]$  is the expected inter-migration time.

Based on the definitions given above, we can define

$X \sim \mathbf{B}(m/E[T], p)$ , a binomially distributed random variable which denotes the number of extra hosts switched on due to Virtual Machine migrations initiated by the machine overload detection algorithm over n time steps. The expected number of extra hosts activated is  $E[X] = mp/E[T]$ . Let A be a random variable denoting the time during which an extra host is active between the time steps 1 and m.

$$B = \frac{1}{n} \sum_{i=1}^n ai$$

$$= \frac{1}{n} \sum_{i=1}^n ai + \frac{1}{n} \sum_{i=1}^n (ai - a1)$$

$$= a1 + \sum_{i=1}^n (ai - a1)$$

The first term a1 is a constant showing the number of hosts which have been initially active and remain active until the end of the experiment.

The second term

$$= \frac{1}{n} \sum_{i=1}^n ai + \frac{1}{n} \sum_{i=1}^n (ai - a1)$$

This formula shows the mean number of hosts switched on due to Virtual Machine migrations being active per unit of time over n time steps. For now, we are interested in look for the average behavior, and for that, estimating the expected value of B. It is proportional to a product of the expected number of extra hosts switched on due to Virtual Machine migrations and the expected activity time of an extra host normalized by the total time. Since the objective is to improve the quality of Virtual Machine consolidation, it is necessary to minimize  $E[H^*]$ . The variable which can be controlled directly by an overload host detection algorithm is  $E[T]$ ; therefore, to minimize the objective of a host overload detection algorithm is to maximize  $E[T]$ , that is, to maximize the mean time between migrations from overloaded hosts.

## 3. RELATED WORKS

The approaches to host overload detection for energy efficient Virtual Machine consolidation proposed in the literature, can be divide into 3 types: Virtual Machine placement (no overload detection), threshold heuristics, and taking decision based on analysis of historical data.

First work, in which Virtual Machine consolidation has been applied was to minimize energy consumption in a data center, and has been performed by Nathuji and Schwan [3]. They analyze in deep benefits of energy obtained by consolidating Virtual Machines using migration and find that the energy consumption can be reduced.

Verma et al. [4] treat the problem of power of Virtual Machine consolidation as a bin packing problem and proposed an algorithm that minimizes power consumption of the data center, considering the cost of Virtual Machine migration. However, the authors did show any algorithm for determining when it is necessary to do optimization of the Virtual Machine placement. The proposed algorithm is only a

periodic invoked to adapt the placement of Virtual Machines. Zhu et al. [5] treated Virtual Machine consolidation problem and show a algorithm to set a static utilization of the CPU threshold of 85% to determine when a host is overloaded. The host is considered to be overloaded when the exceeding the threshold.

The utilization of 85% threshold has been introduced by Gmach et al. [6] based on their analysis they did on traces of workload. In their recent presentation, Gmach et al. [7] gave the benefits of combining periodic and reactive threshold invocations of the controller of migration. VMware Distributed Power Management [8] works based on the same idea with the utilization threshold fixed to 81%. But, static of threshold algorithm are not good for systems with dynamic and unknown workloads, as these algorithms cannot be adapted to the changes. We have concentrated on static threshold algorithm in our previous work [15] by adapting dynamically the threshold's value according to the analysis of statistic of the workload history. In this paper, the static and dynamic threshold algorithm is used as benchmark algorithms in the evaluation of experimentation of the proposed approach.

Jung et al. [9] analyzed the problem of consolidation of Virtual Machines running web applications to find the optimization of a global function, while meeting Service Level Agreement requirements. This approach is the workload specific, as the Service Level Agreement requirements are defined in the response time precomputed for each type of transaction of the applications. When the rate request deviates out of an allowed interval, the system converts the placement of Virtual Machines and the states of the machines.

Zheng et al. [10] proposed the efficiency of automated experimental testing a reallocation decision previous to its application, once the specified response time, in the Service Level Agreement is violated.

In the idea proposed by Kumar et al. [11], the allocation resource is converted when the application's Service Level Agreements are violated.

Wang et al. [17] put a control loops to manage allocation of resource under response time Quality of Service constraints at the server level and cluster. If the capacity of resource of a server is not sufficient to meet the applications Service Level Agreements, a Virtual Machine is migrated from the server. All works are similar to threshold-based algorithm in that they rely on instantaneous performance's value characteristics but they do not leverage the history of system to assume the future behavior of the system and the optimization of the time averaged performance metrics.

Gunter et al. [12] shows an energy-aware the consolidation of Virtual Machine system related on web applications, who's Service Level Agreements are defined in the response time of terms. The authors shown the weight of linear regression to assume the future workload and proactively optimize allocation of the resource.

This philosophy is related to the Local Regression (LR) algorithm already proposed [15]. Bobroff et al. propose an overload server technique based on analysis of historical data [13]. However, the algorithm given is too high level, which cannot allow us to implement it to compare with our philosophy.

Weng et al. [18] propose a load balancing system for clusters which are virtualized. A cluster cost of the Virtual Machine allocation is minimized periodically to detect hosts which are overloaded and under loaded, and Virtual Machines are reallocated. This approach is related to work but with the objective the Virtual Machines are deconsolidated for balancing the load across the hosts. As said above, the limitations of the previous works are that, due to their algorithm basis, they reach to sub-optimal results and don't allow the system administrator to set a Quality of Service goal.

In this work, we propose a new approach to the problem of host overload detection inspired by the work of Bernini et al. [19] also inspired by the nature on power management of electronic systems using Markov decision processes. Since most real-world systems, including Infrastructure as a Service, experience highly variable non-stationary workloads, to fix the proposed model to practical applications non-stationary environments. Even if the final approach is a heuristic, in contrast to the works, it is based on a model that allows the optimal control policy computation for any known stationary workload and a given state configuration.

## 4. PROPOSED APPROACH

### Detection of overloaded host

In order to decide the time to initiate the migration of Virtual Machines from a host, a heuristic was first proposed by Beloglazov and Buyya for setting an upper and lower utilization threshold. But due to dynamic and unpredictable workload, a fixed value of utilization threshold was not suitable [9] and [11]. Therefore, in the later work an auto adjustment technique of utilization threshold has been proposed by the authors based on previous data and statistical analysis of which was gathered during the lifetime of Virtual Machines. The idea was, by considering the deviation of CPU utilization of his heuristic, to adjust the upper bound.

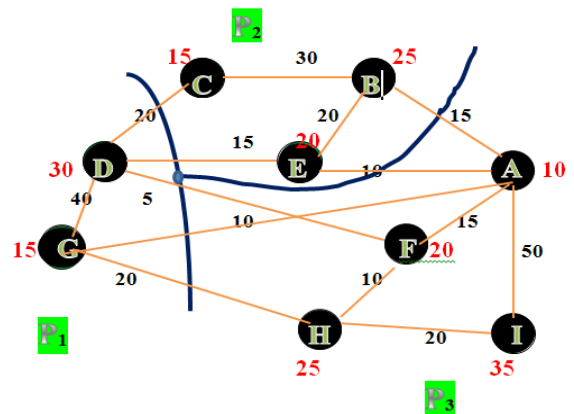


Figure 1: Tasks related to processors

Assume that the modules in the graph are divided among processors  $Pr_1$ ,  $Pr_2$  and  $Pr_3$  as shown by the heavy dark lines [Figure 1]. The total communication cost of this division is 95 units (by counting the communication costs of the arcs which are catted) when clusters are assigned to processors  $Pr_1$ ,  $Pr_2$  and  $Pr_3$  as shown on the graph [Table 1].

By doing this operation of cutting the graph in modules, we can dress a table which shows the total number of tasks assigned to each processor. By adding the total task assigned to every processor, we find the total units assigned to the number of processors as shown:

**Table 1: Total task for every Processor**

Tasks	Processor1 (Pr1)	Processor2 (Pr2)	Processor3 (Pr3)
15	45	60	195
30			
15			
25			
20			
10			
20	90		
25			
35			

**Step 1**

For each Processor Pri, determine the load level Yi

Y1= Load on processor Pr1 = 30 + 15 = 45

Y2= Load on processor Pr2 = 15 + 25 + 20 = 60

Y3= Load on processor Pr3 = 25 + 35 + 20 + 10 = 90

Hence total load = Y1 + Y2 + Y3 = 45 + 60 + 90 = 195

Hence, the average load, A = 195 / 3 = 65

**Table 2: Load Level for every processor**

Processors	Load level for every processor
Pr1	0.69
Pr2	0.92
Pr3	1.385

Zi (the load level on processor Pri) = Pri / A

Hence the load levels are:

Z1 = 45 / 65 = 0.69

Z2 = 60 / 65 = 0.92

Z3 = 90 / 65 = 1.385

**Step 2**

Given the deviation de = 0.15, compute for each processor if it is under loaded, overloaded, or balanced.

Thus the valid range for Zi is 0.85 <= Zi <= 1.15

Since Z1 = 0.69, processor Pr1 is under loaded.

Since Z2 = 0.92, processor Pr2 is balanced.

Since Z3 = 1.385, processor Pr3 is overloaded

This is the way of calculating and finding: under load, balanced and overload hosts [Figure 2]

**Algorithm1:** Detection of overload and underload

**Input:** A deviation (Threshold)

**Output:** State of the processor

If the *maximum deviation de* from equal distributions allowed in the system is given (0 < de < 1), then

Zi must satisfy the inequality 1 - de <= Zi <= 1 + de

If this inequality is satisfied for a processor, then

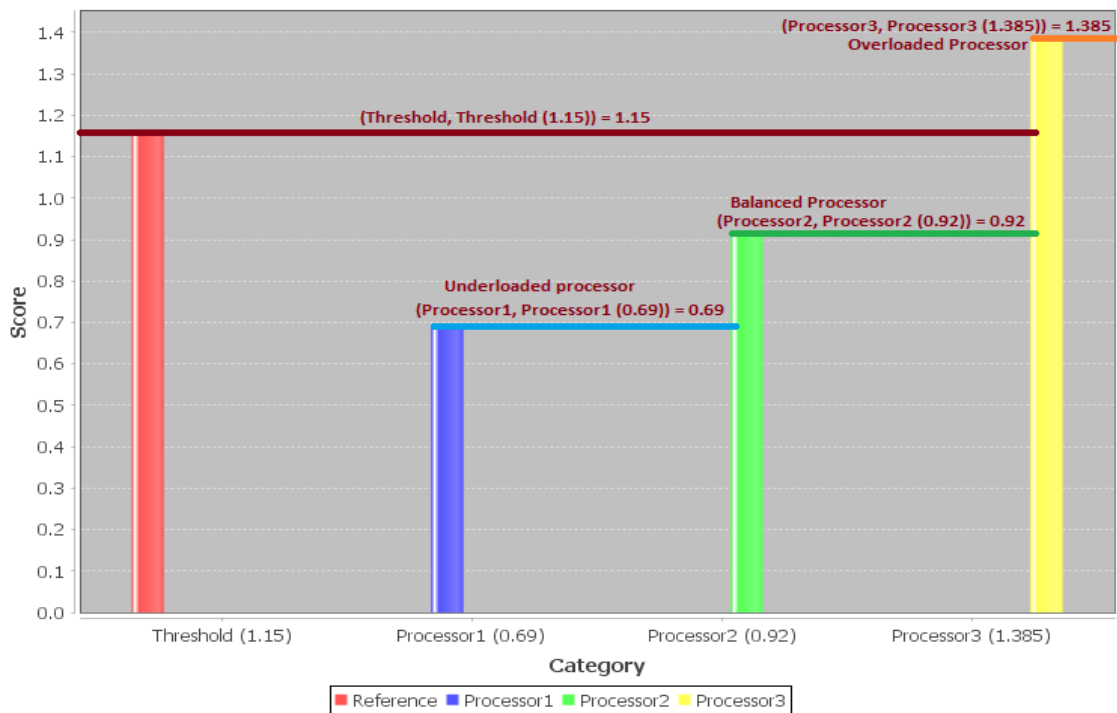
The load assigned to it is acceptable (balanced load),

Else if Zi > 1 + de then

The processor is overloaded.

If Zi < 1 - de then

The processor is under loaded



**Figure 2: Tasks related to under loaded processors**

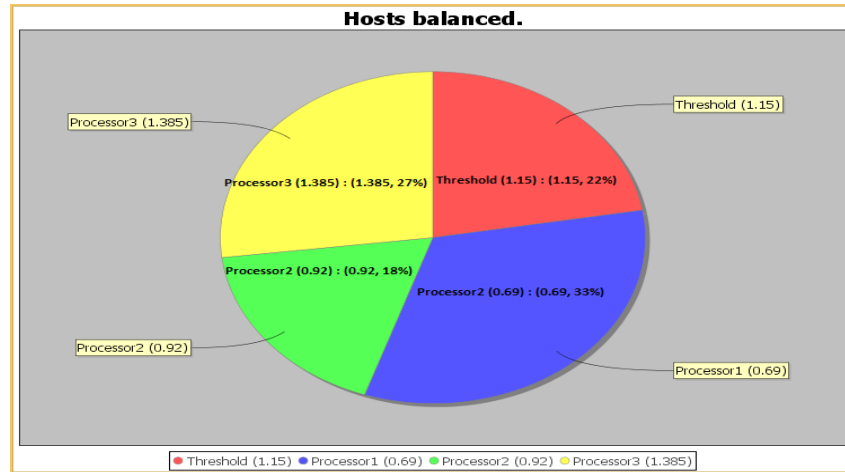


Figure 3: Tasks related to under loaded processors

**Step 3**

To generate a new process graph and modify weight on communication links for reassignment according to the following scheme:

**Algorithm 2:** Operation on Processors

**Input:** A deviation (Threshold)  
**Output:** State of the processor

- If the load of Zi is acceptable, then
- Delete all the modules assigned to Pri from the process graph.
- If Zi is overloaded,
- Copy all the modules assigned to Pri from the original process graph.
- If Zi is under loaded,
- Represent all the modules assigned to Pri by a single module.

In this graph by the help of the algorithm, assign the old Inter module communication costs as arc weights. Then modify these arc weights as follows:

For  $p = 1 \dots t$ , where t is the number of overloaded processors, Repeat the following steps:

For  $q = 1 \dots w$ , where w is the number of under loaded processors, repeats the following:

Find the number  $n_{p,q}$  of arcs which connect the cluster assigned to overloaded processor p, to the cluster assigned to the under loaded processor q. Raise the weight of each of these arcs by  $|(l_p - l_q)| / n_{p,q}$

In our example, we delete the modules of Pr2 since it is balanced and we keep the modules of Pr3 as they are since Pr2 is overloaded [Figure 4]. We reduce the modules of Pr1 to just one module since it is under loaded. The new graph is:

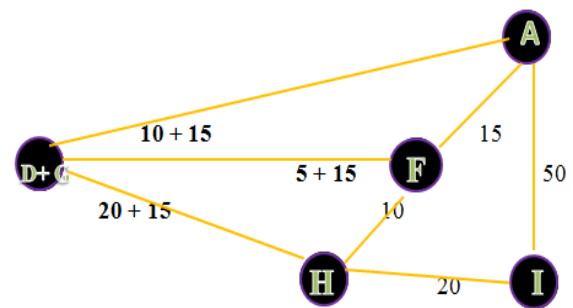


Figure 3: Tasks related to under loaded processors

**A. Virtual Machine selection**

After finding an overloaded host, the next step is to select the particular Virtual Machines to migrate from one machine to another [Figure 2] [Figure 3]. In this part, we will discuss about three Virtual Machine selections.

- **Minimum migration time:** There is a selection of Virtual Machine to migrate that requires minimum amount of time to finish migrating, compared to other Virtual Machines allocated to the same host.
- **Random Choice Policy:** A selection of a Virtual Machine that needs to be migrated according to a uniformly distributed discrete random variable.

In our example, we delete the modules of Pr2 since it is balanced and we keep the modules of Pr3 as they are since Pr2 is overloaded [Figure 4]. We reduce the modules of Pr1 to just one module since it is under loaded. The new graph is:

Now difference in loads between Pr1 and Pr3

$$= Y3 - Y1 = 90 - 45 = 45$$

Divide 45 by the number of two module clusters,

That is 3, Hence,  $45 / 3 = 15$ . Add this weight to each of the three arcs.

**B. Virtual Machine placement**

The Virtual Machine placement problem could be modelled as bin packing problem with variable bin sizes and prices.

The node can be represented as the bin, Virtual Machines that have to be allocated can be seen as the items, bin size can be seen as CPU, price and capacities can be seen as the power consumption of the nodes.

Among several solutions of the problem of bin packing, Beloglazov and Buyya [3] proposed a modification of popular Best Fit Decreasing algorithm that was shown to use bins, not more than  $11/9.OPT + 1$  understand that OPT is the number of bins given by the optimal solution [9] and [13].

**Step 4**

Realizations of the new system such that all the processors are balanced and the costs of communication are also minimized that is determine the optimal cut.

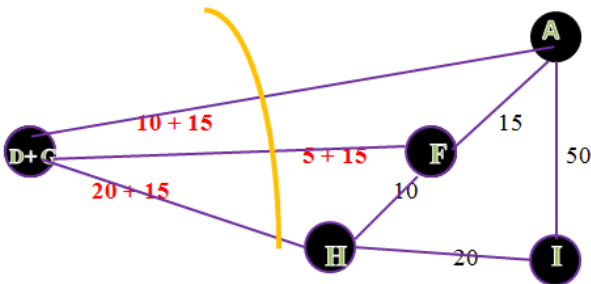
In this, consider two cuts.

Cut A:

The communication costs are  $25 + 20 + 35 = 80$

Cut B

The Cut B is where we allocate module H to processor Pr1. It looks like this:

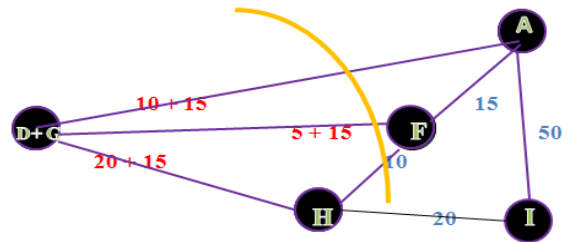


**Figure 4: Process of making Processors balanced**

**Table 3: Load Level of processors after being balanced**

Processors	Load Level of processors after balanced
Pr1	1.0769
Pr2	0.92
Pr3	1

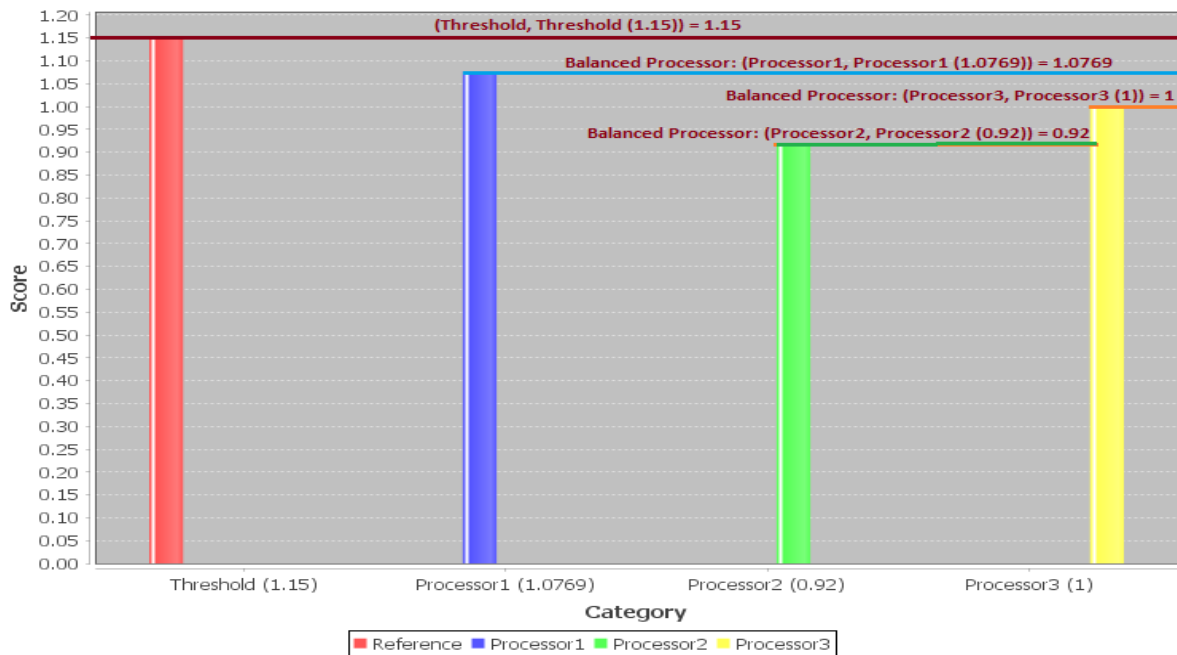
The table above shows, after making all arrangement, that all machines are balanced. Processor 1:1.0769, processor 2 has a value of 0.92 and the third one has a value of 1.



**Figure 5: Processors balanced**

The costs of Communication for Cut B:

$$B = 25 + 20 + 10 + 20 = 75$$



**Figure 6: Processors balanced**

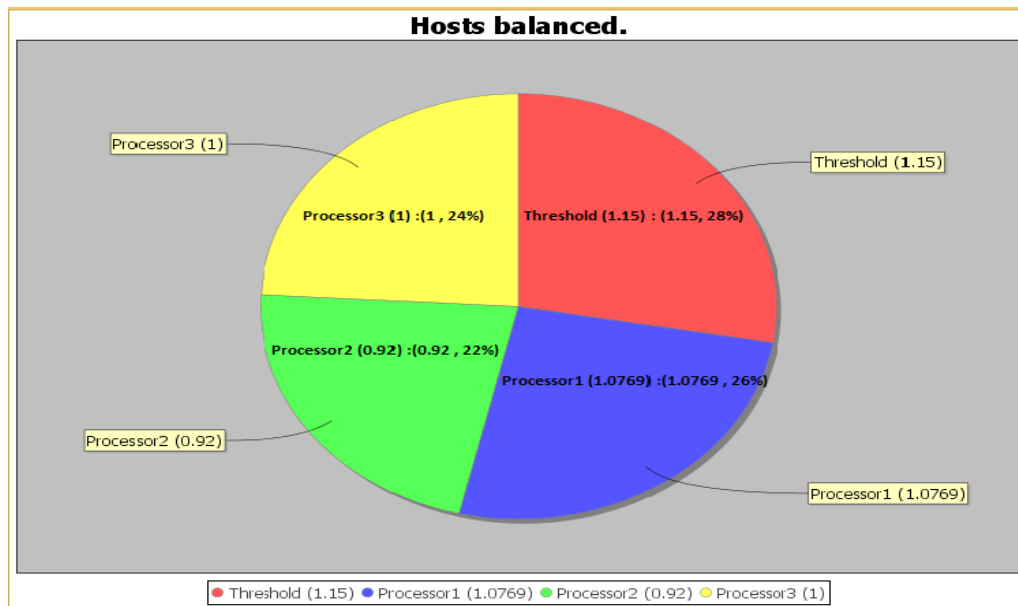


Figure 7: Processors balanced

All 3 processors are balanced.

A cut is said to be optimal when all loads are balanced [Figure 5], [Figure 6] and [Figure 7] and the cost of the communication costs minimized [Table 3].

A cut, which has the lowest cost communication but which the load does not balance, is not considered optimal.

## 5. CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, we have introduced how the nature inspired can help to find the optimization solution of host overload detection and load balancing and proposed a control algorithm for the problem Virtual Machine consolidation.

For a stationary workload which is very well known and a given configuration of state, the control policy obtained solves the problem of the host overload detection, while meeting the Quality of Service goal. We have proposed also optimal algorithms for the problem of host overload detection to evaluate the efficiency and to see if all hosts are balanced. The experimental conducted study has led to the following conclusions:

- For the simulated workload, a method for find an overloaded host in a set of several hosts has been shown and requires simpler computations.
- The Nature Inspiration algorithm helps in a simple way in finding the optimization of solution by increasing the quality of service.
- The balanced hosts has been put beside in order to make overloaded and under loaded balanced as well..
- The algorithms proposed enable explicit specification of a desired Quality of Service goal to be delivered by the system through the parameter offered, which is successfully met by the resulting value of the metric. The introduced model is based on nature inspired algorithm requiring a few fundamental assumptions. However, the experimental study which involves too many mixed workloads has shown that the algorithm is efficient

in handling them [13] and [14]. For the simulated workload the algorithm performed with a performance of the optimal offline algorithm, which is very efficient for an online algorithm.

## 6. ACKNOWLEDGMENTS

The author would like to thank everyone who contributed and gave the input and constructive comments on improvements of the paper.

## 7. REFERENCES

- [1] A, B., & R, B. (2014). *Online Optimal algorithms and heuristic adaptive for energy and performance efficient dynamic consolidation of virtual machines in cloud computing*. Concurrency and computation: Practice and Experience (CCPE),DOI: 10.1002/cpe (in press).
- [2] A, B., R, B., Y, C. L., & A, Z. (2014). *A taxonomy and survey of efficient of the energy data centers and cloud computing systems*. Advanced in computers, M.Zelkowitz (ed), vol. 82, pp. 47 -111.
- [3] A, V., G, D., T, K. N., P, D., & R, K. (2014). *Server workload analysis for power minimization using consolidation*. In proc. of the USENIX Annual Technical Conference, pp. 28 - 29.
- [4] A, V., P, A., & A, N. (2015). *Migration and Power cost aware application placement in virtualized systems*. In Proc. of the 9th ACM/IFIP/USENET International Conference on Middleware pp. 243 - 264.
- [5] B, G., N, J., & C, W. (2014). *Managing cost, performance and reliability trade-offs for energy-aware server provisioning*. in Proc. of the 30st Annual IEEE international Conference on Computer Communications (INFOCOM), pp. 1332 - 1340.
- [6] C, W., M, L., Z, W., & X, L. (2014). *Automatic performance tuning for the virtualized cluster system*. In Proc. of the 28th international conference on Distributed Data Systems (ICDCS), pp. 183 - 190.
- [7] D, G., J, R., L, C., & A, K. (2014). *Pool resource management: Reactive versus proactive or lets be*

- friends*. Computer Networks, vol. 53, no. 17, pp. 2905 - 2922.
- [8] D, G., J, R., L, C., G, B., T, T., & A, K. (2014). *An approach integrated to resource pool management: Policies, efficiency and quality metrics*. In Proc. of 38th IEE international Conference on Dependable Systems and Networks (DSN), pp. 326 - 335.
- [9] E, Y. C., L, B., A, B., Y, H. L., & G, D. M. (2015). *Dynamic management of the power for nonstationary service requests*. IEEE Transactions on computers, vol. 51, n0. 11, pp. 1345 - 1361.
- [10] G, B. (2014). *Queueing networks and Markov chains: modeling and performance evaluation with computer science applications*. Wiley Blackwell.
- [11] G, J., M, A. H., K, R. J., R, D. S., & C, P. (2014). *Mistral: Dynamically managing power, performance and adptation cost in Cloud Infrastructures*. In Proc. of the 30th international conference on Distributed Computing systems (ICDCS), pp. 62 - 73.
- [12] Gartner, I. (2015). *Gartner estimates ICT industry accounts for twoo percent of global CO2 emissions*. Gartner Press Release.
- [13] J, K. (2015). *Growth in data center electricity use 2005 to 2010*. Oakland, CA: Analytics Press.
- [14] K, M., J, F., & C, D. (2015). *Comparing VM- Placement algorithms for on-demand clouds*. In Proc. of the 3rr IEEE international Conference on cloud computing Technology and Science. PP. 91 - 98.
- [15] L, B., A, B., A, P., & G, D. M. (2014). *Policy optimization for dynamic power management*. IEEE Transactions on computer-Aided Design on Integrated Circuits and Systems, vol. 18, n0. 6, pp. 813 - 833.
- [16] N, B., A, K., & K, B. (2015). *Dynamic placement of Virtual Machines for managing Service Level Agreement violations*. In Proc. of the 10th IFIP/IEEE international Symposium on integrated Network management (IM), pp. 119 - 128.
- [17] Q, Z., & B, V. (2014). *Utilization-based pricing for power management in data centers*. Journal of Distributed Computing, vol. 72, n0. 1, pp. 27 - 34.
- [18] R, N., & K, S. (2015). *Virtual power: Coordinated power management in virtualized entreprise systems*. ACM SIGOPS Operating Systems Review, vol. 41, n0. 6, pp. 265 -278.