

Energy Aware Load Balancing in Cloudlet Federation

DOI : 10.36909/jer.ICEPE.19553

Muhammad Inam*, Muhammad Ziad Nayyer*, Muhammad Hasanain Chaudary**

* Department of Computer Science, GIFT University, Gujranwala, 52250, Pakistan

** Department of Computer Science, COMSAT University Islamabad, Lahore Campus,
Lahore, Pakistan

** Corresponding Author: mhchaudary@cuilahore.edu.pk

ABSTRACT

With the rapid increase of compute intensive tasks and workload, the need of the hour is to manage the total workload due to resource constraints and increased cost. For complex computations where multiple computer systems are required to execute a single task such as in federated cloudlet environment, load balancing is the main challenge. Load balancing means to divide the total workload between all the member nodes to get the maximum benefits from the federated environment. A cloudlet is a resourceful computer that is coupled to the Internet and is accessible to the mobile devices in the vicinity. Cloudlet Federation is the concept of a cooperative framework to share resources and load balancing among various member cloudlets. Different tasks consume different amount of energy for their execution that results in a large amount of heat dissipation. Due to heat, the performance of the systems is decreased. The more heat, the more performance degradation, so it becomes very vital to keep the energy consumption as minimum as possible. To address this problem, this paper proposes a novel scheduling strategy that will distribute the total load between all the member nodes in a federated cloudlet environment for load balancing. The proposed methodology not only considers the energy level of the system on which the task has to be executed but also considers the energy required by the tasks in queue. The proposed methodology is tested in a Cloudlet Federation environment and the results show an improved load balancing in terms of energy consumption.

Keywords: Energy Efficiency; Task Arrival; Cloudlet Federation; Power Consumption; Load Balancing.

INTRODUCTION

The core function of a computer system is to meet the end user requirements. All computer systems have CPU as the key component for scheduling and management of different tasks on the basis of available resources (Nayyer et al., 2020 & Mhedheb et al., 2016). Different task scheduling algorithms have been reported in the literature to get the maximum output (Bose et al., 2015).

Cloudlet federation provides different type of services such as computational and storage services (Nayyer et al., 2020). In Cloudlet Federation, systems are virtualized for maximum utilization of the available resources to meet the increasing user demands. Greater energy consumption in Cloudlet Federation results in increased computational cost and releases huge amount of carbon dioxide CO₂ (Mhedheb et al., 2016 & Adhikary et al., 2013). The concept of cloudlet federation is different from cloud federation in such a way that cloudlets target closer proximity and falls in the category of edge computing, whereas cloud computing solutions are based on distant remote cloud.

Keeping in mind the significance of load balancing, various techniques and algorithms such as PALB (Power Aware Load Balancing) (Bose et al., 2015, Galloway et al., 2011 & Shree et al., 2016), e-STAB (Energy-Efficient Scheduling for Cloud Computing Application with Traffic Load Balancing) (Bose et al., 2015), DDR (Dynamic Round Robin) (Bose et al., 2015, Mhedheb et al., 2016 & Berral et al., 2010), and MAX-MIN (Bose et al., 2015) etc. have been proposed. All the techniques reported in literature balance the workload on the basis of the information about the energy consumed by the system only. (Shree et al., 2016, Bose et al., 2015 & Adhikary et al., 2013). The proposed technique observes how much energy a task will consume present in the queue of the system for its execution and also considers the energy consumed by that system and then on the basis of this information selects optimal system for task execution.

To the best of our knowledge no such technique exists, that deals with the challenge of resource optimization in Cloudlet Federation based on energy consumption.

The rest of the paper is organized as follows. Section 2. describes the proposed energy aware load balancer and optimal system selection algorithm. Performance evaluation is presented in Section 3. Section 4. consists of conclusion and future directions.

RELATED WORK

Many researches have been carried out for load balancing and scheduling of the tasks for best utilization of all the available resources. Below is the summary of the studied techniques that have been presented in the literature to meet the challenges of load balancing in Cloudlet Federation (Mhedheb et al., 2016 & Shree et al., 2016).

A technique named Ena-Cloud minimizes the number of running servers in cloud environment and also balance the total workload between all the functioning servers. The main objectives of this technique include load balancing by reducing the number of functioning servers. Another technique called e-STAB, schedule all the available tasks between all the functioning systems through load balancing for the effective utilization of resources. For efficient load balancing and job scheduling it monitors network traffic by avoiding network congestion (Shree et al., 2016 & Bose et al., 2015).

Another Power Aware Load Balancing technique (PALB) monitors all the working nodes to balance the workload in the Cloudlet Federation. After balancing the workload, it evaluates which node should remain powered up and which node should be powered down. This approach works in three steps, first, it finds the server where the VM needs to be placed, second, if required, it powers on the VM and third, if not, it will keep the VM in powered off state (Shree et al., 2016 & Berral et al., 2010).

A common load balancing algorithm called Dynamic Round Robin which is used for VM placement in Cloudlet Federation. This scheme works in two phases, first, when the VM is about to be migrated, it blocks any new task assignment keeping only the running ones. Second,

if the running tasks are taking too much time it will migrate these tasks to the other VM before this VM is forced to be shut down (Nayyer et al., 2020, Shree et al., 2016 & Adhikary et al., 2013).

Another load balancing technique named Max-Min is being proposed that creates a pool of unallocated tasks and selects the task with minimum completion time and maximum execution time. After removing the assigned task from the pool, its execution time is added to the total execution time of all the running tasks. This facilitates the server in load balancing and managing all the tasks on the basis of already available information (Shree et al., 2016 & Wang et al., 2021).

A multi-dimensional maximized balanced resource utilization abbreviated as MAX-BRU is a virtual machine placement algorithm that maximizes the resource utilization by performing load balancing of server resources. It minimizes the number of running servers that results in low energy consumption but it does not explain any scheme for analyzing the load on the server (Shree et al., 2016).

The **Table 1** shows a comparative analysis of different load balancing techniques along with their focusing parameters and benefits.

Table 1. Comparative analysis

Technique Name	Technique Type	Focus	Advantages
Ena-Cloud	Dynamic	Migration of VM, RU, Servers	Reduce the VM migrations as well as servers.
e-STAB	Dynamic	Load Balancing, task provisioning	Avoid network blockage
PALB	Dynamic	Resource Usage and Power	Facilitate load balancing in IaaS
Dynamic Round Robin	Dynamic	Virtual Machine assignment and relocation	Works in dynamic fashion as compared to traditional
MAX-MIN	Static	Execution Time	Effective resource provision and usage
MAX-BRU	Dynamic	Resource consumption and balancing	Minimize running server and reduce energy consumption.

It can be concluded from the above discussion that all the load balancing techniques presented in past are focused on different parameters such as VM migration, resource usage, task-based load balancing, VM relocation and task's execution time. However, there exist no technique

that focuses on the collective view of energy i.e., task and system's energy state combined.

SYSTEM OVERVIEW AND PROBLEM FORMULATION

A. SYSTEM MODEL

In this study a client-server model is used where a broker acts as a server and provides services to the clients on-demand in the closer proximity from adjacent cloudlets. The broker manages all the information about member cloudlets of the federation.

The load balancer is a part of the broker to balance the total work load between all the available systems in the cloudlet federation. When a task requires services from a system, it first requests the broker to provide an optimal cloudlet node to run the task considering load balancing and energy consumption. If an optimal cloudlet is available, the broker will allocate the requesting task to that cloudlet, otherwise the broker will forward the task to the remote cloud as a worst case. A system model of the system is shown in Figure 1.

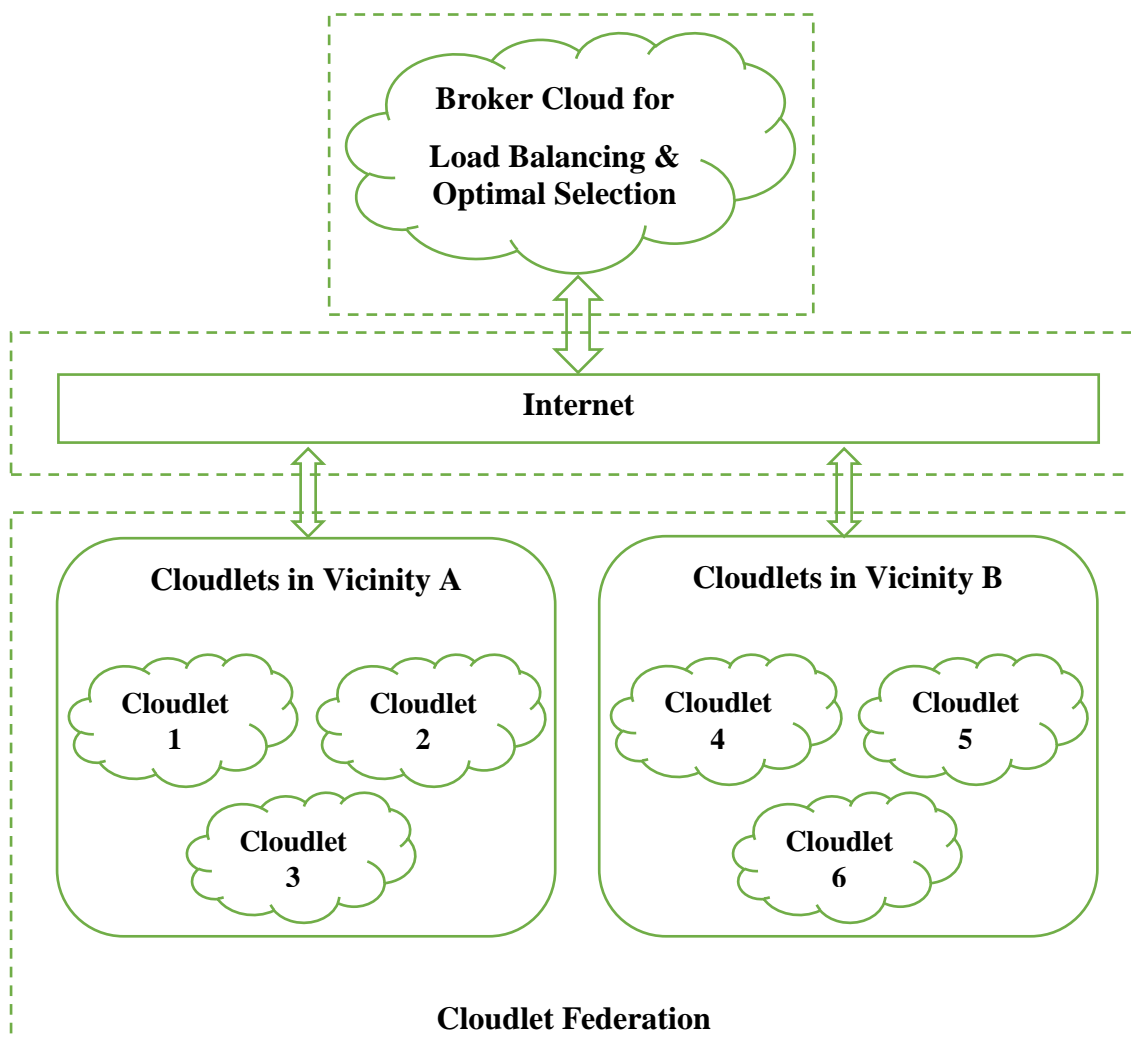


Figure 1. Cloudlet federation model

B. PROBLEM FORMULATION

Consider the systems S_1 , S_2 , and S_3 are executing different tasks with running time having variable energy requirements. Let's assume a task 'k' with arrival time 'a' having execution time 'e' needs to be run on a system 's' in a minimum energy consumption state. The set of already running tasks is represented by $K = \{k_1, k_2, k_3, k_4, \dots, k_n\}$, with arrival time $A = \{a_1, a_2, a_3, a_4, \dots, a_n\}$ having execution time $E = \{e_1, e_2, e_3, e_4, \dots, e_n\}$ on accessible systems $S = \{s_1, s_2, s_3, s_4, \dots, s_n\}$. A sample dataset is presented in Table 2 with systems, tasks and execution time of 3 each task in seconds.

Table 2. Sample dataset

System 1	System 2	System 3
k_1 (2s)	k_4 (5s)	k_6 (1s)
k_2 (3s)	k_5 (2s)	k_7 (3s)
k_3 (5s)		k_8 (5s)
		k_9 (2s)

Total execution time of three tasks k_1 , k_2 , and k_3 on system 1 is 10s, whereas k_4 and k_5 have a total execution time of 7s on system 2 and four tasks k_6 , k_7 , k_8 , and k_9 have a total execution time of 11s on system 3. The notations used throughout the paper are presented in Table 3.

Table 3. Table of notations

Notation	Definition
A_t	Task Arrival Time
D_t	Dispatch Time
W_t	Wait Time
S_i	Initial System
S_o	Optimal System
T_o	Optimal Time
T_e	Task in execution
T_c	Completed Task
T_n	New task ready to be dispatch
T_g	Total energy

The energy consumed by each task that is allocated to the system can be calculated by using

the formula given below:

$$g = P(\text{Watt}) * t(\text{sec})$$

where P is power of current system and t is the execution time of running tasks. A sample dataset has been presented in Table 4 considering three systems with their energy levels and four tasks with their required energy.

Table 4. Sample Dataset

	System	Task W	Task X	Task Y	Task Z
Systems	Initial Energy Level	Required Energy	Required Energy	Required Energy	Required Energy
System 1	32J	2J	8J	11J	-
System 2	38J	7J	3J	-	-
System 3	37J	1J	4J	6J	2J

Let's assume some tasks are already waiting in queue on the respective systems. A new task k_{10} arrives for scheduling and is dispatched to system 2 because it has minimum total energy as compare to available systems. A simple presentation is shown in Figure 2.

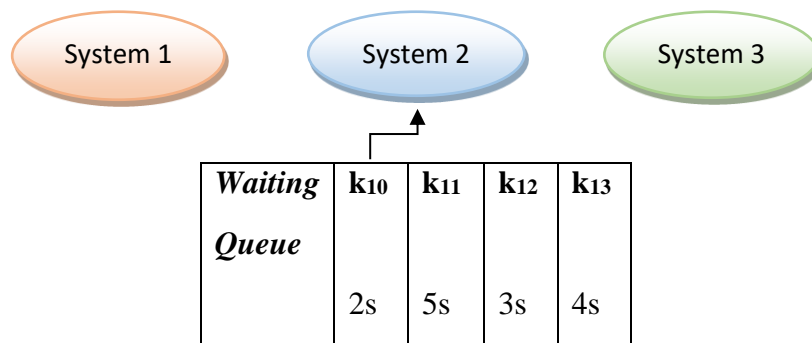


Figure 2. Overview of system

Cloudlet Federation have multiple systems and each system has multiple tasks k_i define by:

$$S = \sum_{i=1}^n k_i \mid i = 1 \text{ to } n$$

$$\forall \text{ tasks } k \in K$$

ENERGY AWARE LOAD BALANCER

The proposed technique provides an energy aware load balancing which focuses on the energy of the system in which there are different choices of selecting present systems at Cloudlet Federation. The suitable system has minimum total energy T_g according to the current situation. This technique requires list of systems, list of power consumption for each system and list of running tasks. A simpler presentation of proposed framework is shown in Figure 3.

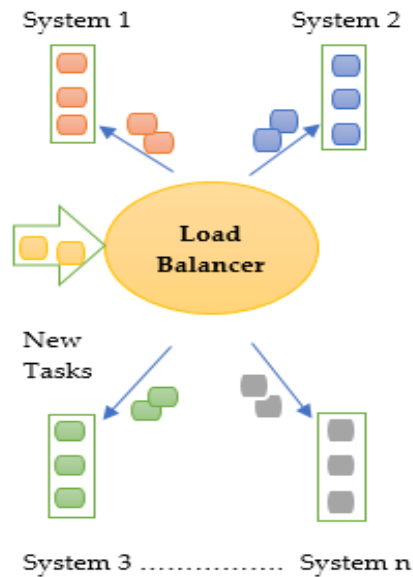


Figure 3. Proposed technique for load balancing

The tasks present in the queue are waiting for the balancer to be assigned to the available systems by balancing the load and energy consumption at each system. The proposed energy aware load balancer schedules the task to the system with minimum initial energy level and the energy required by the tasks in waiting queue.

A. SYSTEM ENERGY CALCULATION

The total energy of a system contains the sum of energies of all the tasks executing on it from i to n and can be computed by using the below equation:

$$T_g = g_1 + g_2 + g_3 + g_4 + \dots + g_n$$

Where g_1, g_2 are the total energy of task 1 and task 2 respectively and so on. The total energy consumed by a task to be executes on an allocated system can be computed as:

$$g = P * t$$

Where P is the processor power of current system and t is burst time of task.

B. OPTIMAL SYSTEM SELECTION

The presented algorithm 1 focuses on searching the optimal system for execution of tasks. The algorithm takes systems list, their power state, energy requirements for the tasks as input values, check the status of system and then after doing the comparison finds available optimal system.

Algorithm 1: Optimal System Selection (OSS)

Input: List of systems,

list of power consumption for each system,

list of running task on each system

Output: Optimal system S_o

```

1  Begin:
2    Let optimal system  $S_o$  be NULL
3    for each task at load balancer do
4      Check status  $T_c$ ,  $T_e$ , and  $T_n$ 
5      if  $T = \text{new}$ 
6        for each select system with min  $P/C$  do
7          Check  $T_g$ 
8          if  $S_i \leq S_i + 1$ 
9            end for
10         end if
11          $S_o = S_i$  with minimum  $T_g$ 
12       end for
13     return  $S_o$ 
14  end:

```

PERFORMANCE EVALUATION

In this section, performance analysis of the offered technique introduced in section III has been studied.

A. PERFORMANCE METRICS

To differentiate the effectiveness of the presented technique several performance metrics have been preferred. The first one is task arrival time A_t , the second is waiting time W_t , the third metric is execution time T_e of the task and the fourth metric is power consumption of the system.

B. EXPERIMENTAL SETUP

This setup has been installed in a Cloudlet Federation on the systems having 2.4 GHz Intel Core i3-2370 M processor, RAM 2GB, DDR3 drive and cache 3MB. The proposed model has been installed with the help of 3 systems. Below is the representation of setup in Figure 4.

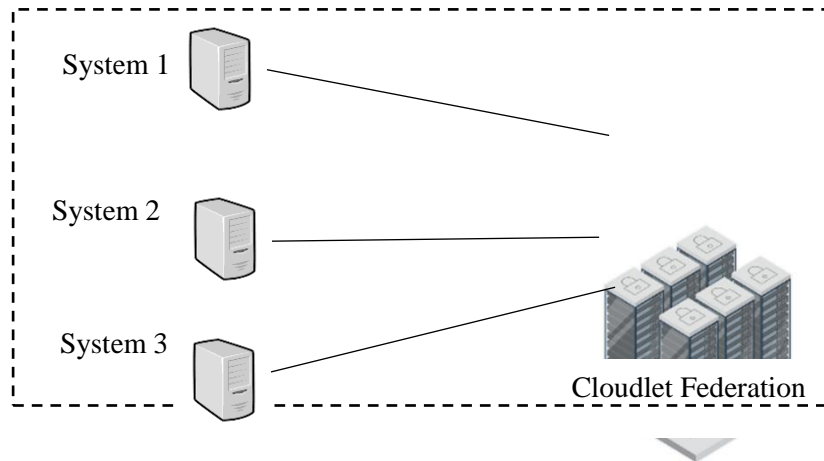


Figure 4. Experimental setup

C. RESULTS

The proposed algorithm has been executed three times to attain an optimal system having average minimum amount of energy. EALB considers total energy that is the sum of system's initial energy and the energy required by the tasks in queue. Sample results regarding systems' energy level are shown in the Table 5.

Table 5. Energy Level of Systems

Systems	Considered Energy	Considered Energy	Considered Energy
	Level for System 1	Level for System 2	Level for System 3
EALB	44J	40J	48J
Others	32J	35J	38J

If a new task T has arrived all other techniques will select System 1 ignoring the energy required by the tasks in queue hence providing an inaccurate selection. However, EALB will select System 2 considering both energy levels i.e., system's energy level and energy required by the tasks in queue hence providing a more accurate selection. A sample containing averages

of initial energy level of the systems, energy required by the tasks in queue, and total energy level of the system is shown in figure 5 to conclude the accuracy of correct selection.

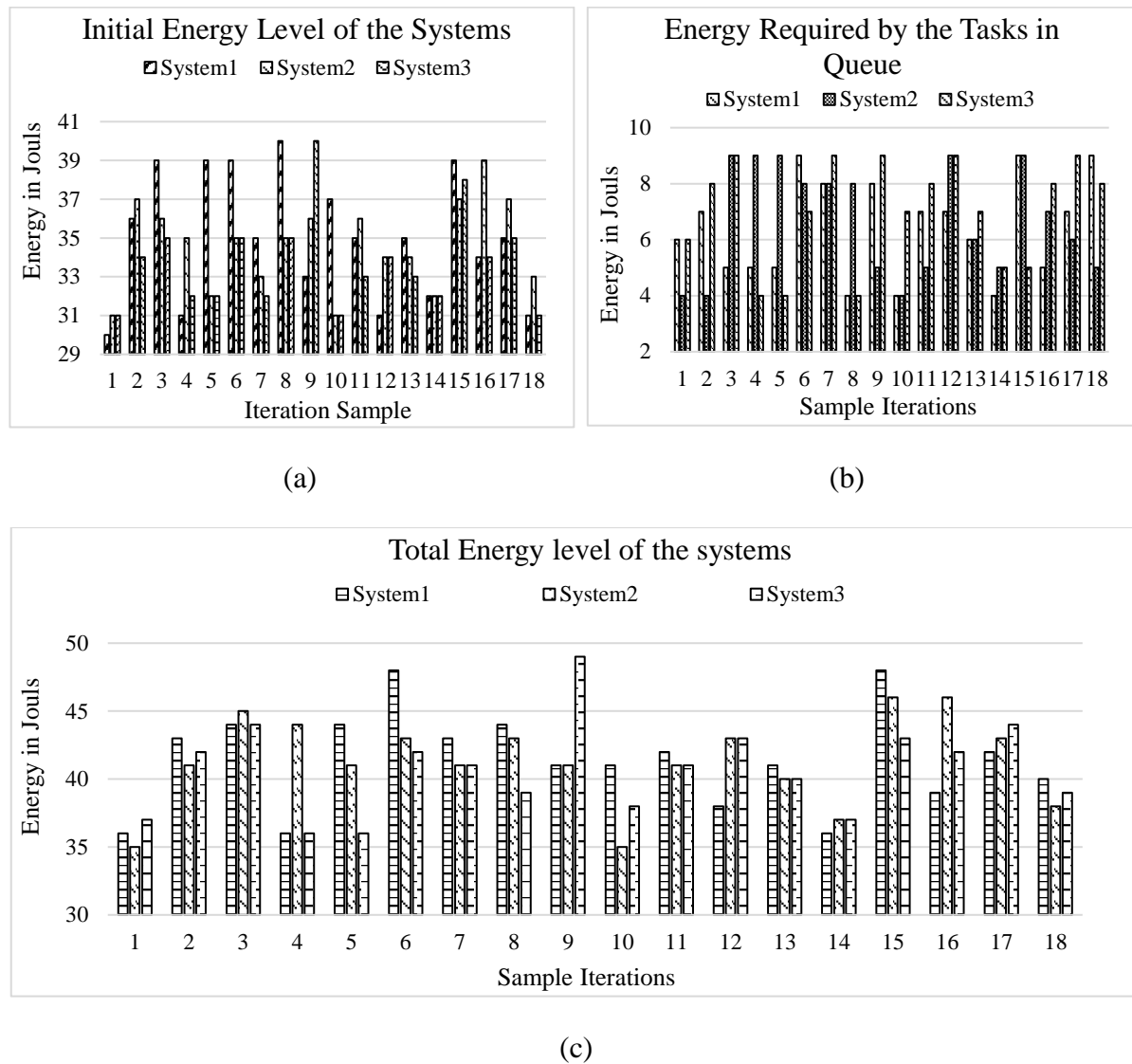


Figure 5. Energy Graph

There exist 17% cases where other techniques have failed to select the correct optimal system due to considering only the initial energy level of the system and ignoring the energy requirements of the tasks in queue. The contradictory selections based on initial energy level of the systems by other techniques and based on the total energy level of the system by proposed scheme has been shown in figure 6.

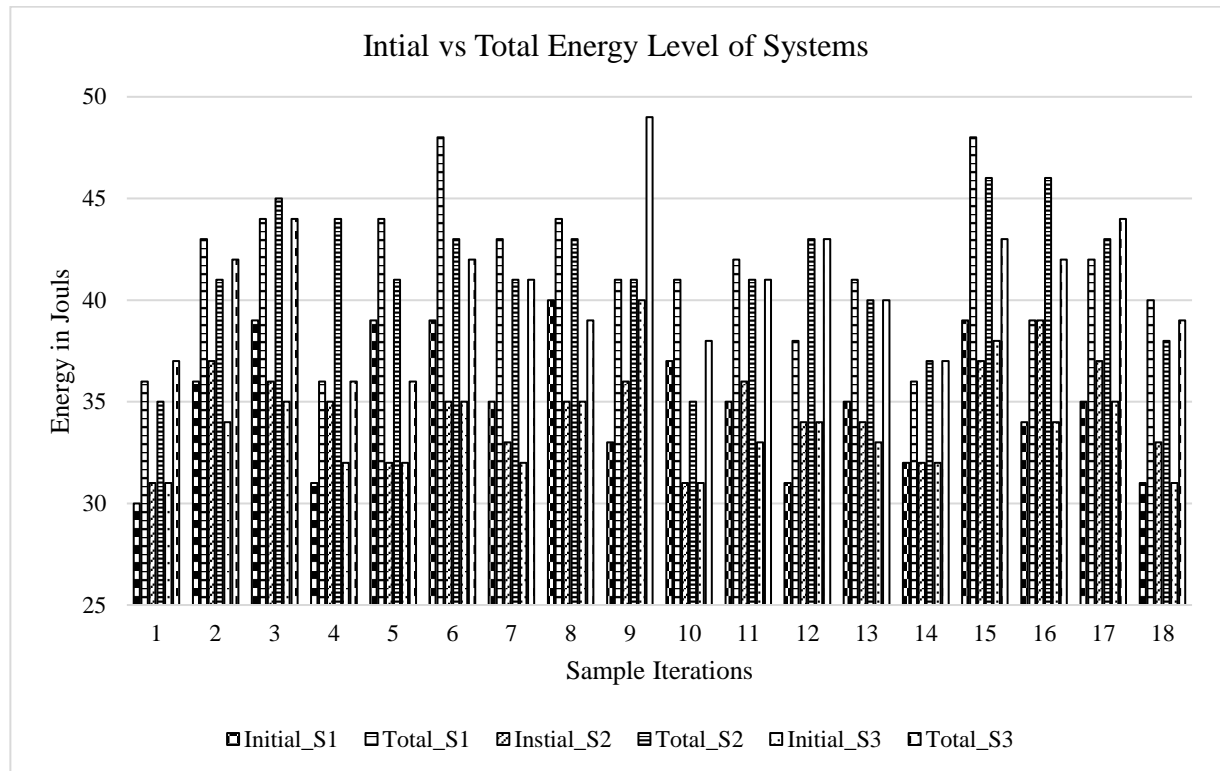


Figure 6. Initial vs Total Energy Levels

CONCLUSION AND FUTURE DIRECTIONS

Load balancing is a hot and challenging research area in current age which involves effectively distribution of the load to the available systems. In this paper, an Energy Aware Load Balancing technique is introduced which will assist to find the optimal system in task scheduling for the best balancing of resources and energy. The proposed scheme not only considers the current energy consumed by the system but also considers the energy required by the tasks in waiting queue and hence provide a more accurate allocation. In future, other performance parameters such as execution time and carbon dissipation can also be considered to develop a more efficient algorithm.

REFERENCES

- Nayyer, M. Z., Raza, I., & Hussain, S. A. (2020).** CFRO: Cloudlet Federation for Resource Optimization. *IEEE Access*, 8, 106234-106246.
- Mhedheb, Y., & Streit, A. (2016, April).** Energy-efficient task scheduling in data centers. In *International Conference on Cloud Computing and Services Science* (Vol. 2, pp. 273-282). SCITEPRESS.
- Shree, T., & Badal, N. (2016).** Energy Efficient Load Balancing Algorithm for Green Cloud. *International Journal of Engineering Research*, 5(03).
- Wang, B., & Liu, F. (2021).** Task arrival-based energy efficient optimization in smart-IoT data center. *Mathematical Biosciences and Engineering: MBE*, 18(3), 2713-2732.
- Bose, S., & Kumar, J. (2015).** A survey on energy aware load balancing techniques in cloud computing. *Int. J. Adv. Res. Comput. Commun. Eng.*, 4(5), 397-402.
- Adhikary, T., Das, A. K., Razzaque, M. A., & Sarkar, A. J. (2013, November).** Energy-efficient scheduling algorithms for data center resources in cloud computing. In *2013 IEEE 10th International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing* (pp. 1715-1720). IEEE.
- Berral, J. L., Goiri, Í., Nou, R., Julià, F., Guitart, J., Gavaldà, R., & Torres, J. (2010, April).** Towards energy-aware scheduling in data centers using machine learning. In *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking* (pp. 215-224).
- Galloway, J. M., Smith, K. L., & Vrbsky, S. S. (2011, October).** Power aware load balancing for cloud computing. In *Proceedings of the World Congress on Engineering and Computer Science* (Vol. 1, pp. 19-21).