Traffic Intensity Based Energy Efficiency Architecture for Data-Centers

Syed Shah Muhammad¹, Salman Qadri^{*2}, Hasnain Ahmed¹, Syed Furqan Qadri³, Najia Saher², Sarfraz Ahmad¹
Tanveer Aslam², Syed Ali Nawaz² and Muhammad Tariq Pervaiz¹

¹Department of Computer Science Virtual University of Pakistan, Lahore Punjab Pakistan.

²Department of IT, Islamia University of Bahawalpur, 63100 Bahawalpur Punjab, Pakistan.

³School of Computer Science and S.E, Shenzhen University, Shenzhen, China.

salman.qadri@iub.edu.pk,

ABSTRACT

The world is moving towards cost-effective and time-constrained solutions. The uses of applications and automated devices have been growing day by day. In computing, resources available in personal computers are limited due to less storage capacity and lower computation speeds. Using all applications on personal systems may not be cost-effective. Therefore, the trends of online storage and computing have become popular. On the other hand, there must be some serving end for these users. One of the major issues, due to the growth of data centers is the increase in power usage of a larger number of servers and network devices. These devices are power-hungry and consume energy even during idle hours even if there are no network traffic loads. The cost of energy used and dissipated is increased in this situation. In this paper, we have given a solution for efficient usage of energy efficiency in data center networks based on traffic loads. We have proposed a model to use traffic intensity to decide the number of machines inactive conditions so that we can save the energy consumption in data center networks. We have implemented this proposed model and simulated it to validate it.

KEYWORDS: Energy efficiency, power consumption, data centers, online storage, traffic intensity, energy aware routing.

1. INTRODUCTION

Data centers are technologies which are providing resources of information technology at the backend of a cloud. They are providing hardware, software, and infrastructure on a payment basis without any hurdle of deploying a new infrastructure according to requirements. Renowned companies like Google, Yahoo, Amazon had established their data center for services. These data centers are built on a large number of servers and network devices. In turn, all these servers and network devices use a large amount of energy. The power reduction and energy losses are major concerns of this age [1]. According to a survey energy usage for data centers is around 31 GW all over the earth [2]. Energy consumption is also continuously increasing. Only in U. S., this energy has been observed to increment up to 40% from 2007 to 2010 [3]. The energy consumed by data center networks in the US was 1.7 % to 2.2 % of total electricity usage in 2010 [4]. The major part of the data center is the data center network which plays an important role in traffic flow among resources. Datacenter networks (DCN) are supposed to work

even for the maximum traffic loads. This traffic peak may not be the same all the time [5]. According to [6] a data center's average usage is 20 to 30 %. Also, resources are using energy even if there is no traffic flow [7]. Many solutions for energy efficiency in DCNs are available [8–12]. Due to the large amount of energy consumption in data center networks has become abundant [13–15]. Keeping in view all these facts there is a need to provide energy-efficient solutions for data center networks. Many solutions for data center networks had been proposed by researchers. Commonly, data centers use a 1:1 subscription ratio for communication over links but practically there are all to all scenarios as per network traffic loads. In most hours, the network devices and servers are under-utilized due to variable usage at day and night time. In most situations, network traffic flows in data centers are high during office hours and lowered at night times. Due to this the devices and links in data center networks are underutilized but consume the same energy. In the US the energy consumption was 3 billion kWh in one year [15].

This is due to fact that many of the chips, fans, line cards inside the network devices consume a large amount of energy even when not participating in traffic flows. Also, the network devices use around 30% of energy in data centers [16]. The energy-aware solutions can make efficient use of devices [17]. As the data centers are the backbone of all services provided over the cloud [18], therefore the data center networks are the major energy-consuming infrastructure of the world [19]. Some of the researchers have observed the fact that the energy consumption of data centers will be raised to 8% of total energy usage of the world by 2020, which was around 1.5% in 2011 [20,21]. Some of the authors have proposed virtualization of network resources and the consolidation in physical machines can reduce energy consumption [22]. Many of the solutions provided and the governing protocols do not take care of proportionality measures between network load and several user devices in communication. There might be a lower subset of devices that is enough for off-hours when traffic load is smaller. A solution based on bandwidth calculation for making the energy proportional network in data centers has been proposed by the researcher [23]. PTNet topology has been introduced by a researcher which provides scalability along with minimum latency, high performance, and lowest energy consumption [24]. There are some solutions but they involve long routes and some of them have high computing complexities. Also, performance issues and delivery delays have been observed. Based on these observations there is still a need for energy-efficient routing in data center networks

In this paper, energy conservation is based on routing decisions. This will be energy-aware routing for energy efficiency in data center networks. The basic idea is to use a fewer number of devices involved in routing with minimum effect on network performance. These devices can be shut down for this purpose.

First, we have formalized the problem for routing and then to prove that it is NP-hard. After this, we have proposed a routing algorithm to achieve the energy-efficient solution. This routing algorithm will focus on achieving energy-efficient routing with reduced computation complexity along with the minimum effect of the performance.

The algorithm is devised to first calculate the throughput based on the network traffic patterns which serves as a basic parameter. Also, the core nodes are determined which have a maximum load at a given time t . these nodes are kept alive for routing. These core nodes are selected based on their usage which can never be put to shut down. After that, the algorithm excludes the devices from routing paths until there is no effect on performance. In the end, these excluded switches are turned off. The major contributions are:

1. Problem formulation for energy efficiency in data center networks and its solution.
2. Finding the core nodes involved in the routing and the nodes to be excluded from routing.
3. Focusing on the energy-efficient model with lowest compromise on performance and computing complexity.
4. Implementing and evaluating the proposed solution

Simulations have been conducted to evaluate the effectiveness of this proposed energy-aware routing algorithm. Finally, the results show that this is an efficient solution for energy conservation in data center networks under different network traffic loads.

Procedures to utilize the traffic intensity to control active nodes in a data center in such a way to allow the only required number of nodes running and switch off the additional nodes. The idea is to establish a relationship between the requests and the required number of nodes to handle the incoming requests efficiently and optimize the data center performance and energy utilization. At the end of this paper, we have shown the simulation results in virtual data-center using the NS3 to generate the traffic and Docker containers to act data center nodes. Before establishing the mathematical relationship, we discuss some important terminology that shall be used in this paper. The request is a formal request from a user or to an end client that needs to access resources from a server node in the virtual data-center. In real-life scenarios, this request could be transactions, video or voice session, a file transfer, etc. Requests could have certain characteristics like priority, time, and order. In our simulation, we shall consider that all requests have the same priority, completion time, and order of fulfillment as per the request arrival order. The response is the server's reply after using or deploying some resources of it. This could be simple processing. Once the request is fulfilled server generates a message the request is completed.

Request Completion Time (RCT) is the time elapsed from request generated by a user and the user received a response. RCT is the sum of the processing time plus network delay or queuing delay. The capacity of server nodes is currently inactive state corresponding to active traffic intensity. Active traffic intensity is the value reported by the network. The nodes are the servers in the data center that can fulfill the user request. Server nodes are connected in groups referred to here as a cluster. Several nodes are connected to a switch or bridge. Once all the nodes in a cluster to switched on, the relevant switch or bridge can be also turned off to save energy.

This paper refers to the request-response model having the request completion time. The general

definition of traffic intensity as defined by the ERLANG is applicable here; however, the calls and related specifications are translated to the request-response model. Originally the errant determines the optimal trunks required given PABX, but this paper will determine the number of server nodes required to fulfill the user requests with optimal performance and data center efficiency.

The rest of the paper is arranged as follows. In section 2 the latest work done in the relevant field has been elaborated. In section 3, we have formulated the problem and presented the algorithm for energy-aware routing in data center networks. In section 4, the results are discussed to validate the authenticity. In section 6, the work is concluded. Section 7 gives future directions for this research.

2. RELATED WORK

The major issue faced by data centers from a smaller scale like public and private sector organizations to very large scale data centers like Facebook, Amazon, Google, Microsoft, is the energy consumption. Due to this reason, finding the solution to optimize energy usage has always been a research-focused area. This is not only to reduce energy consumption but also to protect the environment. A large amount of work has been done to minimize energy consumption. The researchers have proposed different approaches to minimize energy consumption in different ways. In the following, we have presented some state of artworks regarding energy efficiency in data centers.

2.1. Energy-aware architectures

In category, the researchers focus to design or use such topologies or architectures for the data centers which help in saving the energy utilized. For this different energy efficient architectures have been proposed. Nano data centers [21], have been proposed in this regard. In this topology, home gateways are used to save the services and content rather than in data centers. A P2P infrastructure can be used to access this data. Some other topologies have also been proposed for the energy efficiency of wireless sensor networks [22-23]. A large amount of energy can also be saved by using transceivers with a minor energy consumption compared to switch interfaces [24].

NovaCube [19] and flattened butterfly topology [20]. A reasonable amount of energy can be conserved by making the use of such topologies, by optimizing the energy usage by nodes and network devices. Some of these above-mentioned approaches are feasible for wireless networks and others for a small network.

2.2. Power Efficient Devices

In this approach, the major motivation is to make the nodes and network devices use the energy as per network traffic. For this Dynamic Voltage (DVFS) [18] can be used. DVFS technique is used to make the

power usage of devices such as CPU as traffic loads. They used the idea of voltage to frequency proportionality. This had been used to energy used by a chip proportional to the network traffic load. Researchers have used a variation on/vary off approach to implementing this idea. The workload can be distributed among a smaller subset of nodes rather than spanning it to all nodes. They proved that this could be an efficient way to conserve energy. This approach is limited to minimize the energy usage by the CPU while all other devices are using the same huge amount of energy.

2.3. Approaches for Renewable Energy

Some of the researchers showed that solar energy, wind, cool regions, water, and wind provide support to reduce a large amount of energy consumption in data centers. Net-zero networks have also been proposed Arlitt et al. in 2012 [17]. In this approach, an amount of energy can be regenerated as the amount of energy consumed by the data centers in one day. Nguyen et al. (2013) [16] focused to provide renewable energy for the data centers. They used solar and wind energy systems, along with the sensors for temperature and moisture as a testbed to implement the energy-efficient solution for data centers. They implemented their solution in different US projects and achieved fruitful results for energy efficiency. These renewable energy solutions have some limitations. First, the initial infrastructure costs and the delivery cost of renewable energy from these sources is high. Secondly, these sources depend on the weather conditions, as the solar works in sunlight. The locations of these data centers also play a role limit the efficiency of such solutions.

2.4. Virtualization

A single physical machine can be used as multiple machines by using a technology called virtualization. In this technique, more than one instance known as Virtual Machines (VM) is created using a single machine [11,25]. These VMs interact with the operating system as a real machine. Generally in this approach, we can use one machine which is acting as more than one machine thus reducing the energy used by other machines. One of the popular tools used for virtualization is VMware [26].

Some researchers [27] have user requirements and network devices usage to suggest algorithms for the placement of virtual machines. Also a comprehensive study had been conducted [28] to understand effect of two different energy consumption techniques (RSS and MNP) for different types of data center architectures.

Also GreenCloud [29] concept of virtualization in data centers. For this, the authors first investigated the network traffic and proposed the routing algorithm for the placement of virtual machines. They got results for energy efficiency along with

good network performance. They carried out this work in two parts: first by finding the network traffic loads and the placement of VMS as per requirements. The extra machines which were mapped to VMs can be turned off to optimize energy usage.

2.5. Routing level power aware approaches

Different researchers have worked on energy-aware routing to provide the solutions. In this category, energy efficiency is applied to communication devices in such a way that energy can be conserved in a considerable amount. The basic idea behind this rationale is that the devices not participating in communication consume the same amount of energy. This concept can be used to devise energy-aware solutions by merging traffic flows in such a way that only a specific group of communication devices is used. The rest of the devices are put to power off to save energy. Energy-aware routing algorithm [13], Energy optimization on-device interface approach [30], Elastic tree [9], Artificial intelligence abstraction model for Energy efficiency in DCN [2] and traffic merge [31] approaches are some examples in this domain.

2.6. Elastic Tree

One of the energy aware solutions is elastic tree [9]. In this approach, the authors have focused to find a minimum number of communication devices that can be used with a lesser impact on network performance. This approach is based on three modules: optimizer, routing module, and power controlling module. These three modules are implemented to get the power-aware routing in data center networks. The optimizer finds the lowest number of communication devices which is enough to provide the traffic flows required in the data center. In the routing part, the required routes are calculated. In the third module, the states of devices are adjusted. They proposed to use three optimizers for the calculation of the minimum number of devices which are enough to fulfill the routing requirements. This optimized solution is achieved as per traffic conditions in the network. For each flow, they calculated the routes which could be used and chose the left one. Assigning traffic flows is NP-hard and takes long time duration of hours. The scalability is also a major issue for the optimizer which can handle only up to 1000 devices. Greedy bin packing is used to enhance the scalability but still, the solution is not optimized. Also, this approach was unable to give the required quality of service as they were unable to calculate the exact routes

The communications devices specifically, switch consume large amount energy in data center networks. In traffic merging [31] technique, focused to reduce the energy consumption in data center networks. They merged the traffic from different links reaching to links by utilizing the lowest subset of ports. This merging technique was used in such a

way that many of the ports are using the lowest energy consumption manner. The traffic is merged such that the number of traffic flows is reduced to use the number of minimum ports for communication. This method saved maximum energy in data centers, but this approach has a limitation that it can only save the energy consumed by the network. Another risk involved in this approach, some of the merged links or ports may lose communication.

There is another way to address the energy efficiency is energy-aware routing proposed by researchers [13]. The basic idea is to use the lowest number of switches for routing the traffic in data center networks keeping the network performance in consideration. They calculated the route and performance of putting the machines off. Then in the second phase, they put the unwanted machines to off condition keeping the performance up to a certain level. However, this approach has some issues. The computing time for the elimination of machines is incurred. Also, this saves only energy used by switches and not the energy used by the links and servers. In some cases, the performance is degraded. This could be fatal for the data center where the real-time applications are served.

A fair bandwidth allocation technique using artificial intelligence abstraction is proposed by researchers [2]. They used the blocking island (BI) standard for bandwidth allocation to energy efficiency on data center networks. They implemented this idea in two phases. First, they applied BI to efficiently use the network devices, reduced the communication flows, and then found the path which consumes minimum complexity. In the second part, they presented an energy-efficient routing algorithm. This algorithm selects routes as per traffic intensity by shutting down the unwanted routes. This approach has lowered the calculation complexity for traffic loads but still, it has some limitations. Calculating all routes and selecting the optimized path is still so much complex.

Energy saving by the optimization of interface is proposed by researchers [30]. It changes the state of the port of a switch as per the traffic queue length. They make use of traffic loads to minimize the energy consumption on switch ports. Initially, they assumed that network devices are not active. These switches are activated upon the traffic arrival. The traffic is considered as packets. The queue is observed after the arrival of each packet. The port is downgraded to a lower level when the buffer goes on a certain lower level *Town*. The port can be made up if the buffer level grows to a certain level *Tup*. In this practice, a port can have many different states. It may be in a sleep state, or some other states like 100 Mbps, 1 or 100 Gbps as per buffer situations. This approach saves a reasonable amount of energy but some of the limitations of this approach are also there. The computation time is high. In some cases, network reliability is compromised and performance

is degraded. To address such issues we have proposed a new approach, which saves maximum energy with minimum impact on network performance.

3. PROPOSED MODEL

For the implementation of a traffic intensity based energy-efficient solution, we require certain additional capabilities to be included in the data center environment. These include a power monitoring module, routing algorithm, server management API, Network Management API. In addition to this network, nodes must be capable of propagating the traffic intensity information as described in the paper. The working principle of this model is elaborated in the sections below.

3.1. Monitor

The monitor is a daemon that keeps monitoring the load on nodes and determines the criteria whether new nodes need to be switched on or the existing node needs to switch off taking account of some preferences that cover the time required for start time. Consider $T(s)$ is the node startup time and the average request serving duration is D .

Average Request Completion Time = T_{comp}

Total number of requests being server = $\sum_1^N R_n$

Estimated remaining time of completion = $\sum_1^N \sum_1^{R_N} T_r$

3.2. New Request

Figure 1 depicts the operation of the algorithm. CC represents the current capacity in traffic intensity units and CTI refers to the current Traffic intensity value in a given interval. Request refers to a new request from a client or user that needs to reserve or use some resources from the datacenter. We consider only the computation

resources only. VDC is the virtual node controller. N-Free represents the available nodes that are not being used and have been shut down in some previous interval or were never started previously. The monitor evaluates the current capacity of the datacenter in TI units.

Figure 1: Operation of the algorithm of Monitor Algorithm

The datacenter nodes are started and stopped as per the CTI and CC values. K is a parameter that addresses the issues of the node start and stops time and M represents the monitored value. The start is a function that triggers the virtual node to boot up the n th node and a similar stop is a function that triggers the shutdown of the n th node.

3.3. Start Node

Nodes are started if their demand of resources and current capacity reaches the startup and shutdown threshold. All nodes in the datacenter are assumed to have the same capacity and capabilities. So starting a node n is just a matter of selection of a node from the shutdown pool. During the node boot up the incoming requests are forwarded to already running nodes having the least load. Once the new node is started new requests are forwarded to the new boot-up node. Existing requests are never transferred between the nodes.

3.4. Stop Node

A Node is required to be shut down if there is no request being served by it and all other running nodes are running with a load below the threshold. If there is no request being served by a node and other nodes are operating near to threshold or above the threshold, the stop operation is rejected and the load balancer forwards the new requests to this node. The Stop operation makes sure that there are enough resources available in the data center if the node stopped. This condition is limited only to the current interval where TI values are considered. If the interval expires algorithm recalculates the over status and decides whether the node needs to be shut down or there will be future requests that shall need this node.

If multiple nodes are running and the current capacity is sufficiently lower than the current interval TI, the load balancer tries to move the incoming

request to specific nodes that fulfill the criteria for future operation. This means the nodes that are decided to be kept on shall receive all future requests in a given interval. This is the result of freeing up the non-interesting nodes. The decision of shutting down a node is critical as compared to starting up a node. In a later case, the random node is selected from the off-pool whereas the selection of shutdown node is done with coordination of load balancer and monitor. So if the algorithm has decided to shut down a node, there is not guaranteed that the node will be shut down due to currently running requests. The shutdown operation may be delayed depending on the load balancer. Also, it is possible that in a given interval where a shutdown has been decided by the algorithm may never actually happen in this interval.

3.5. Stop Cluster

In a datacenter, nodes are physically clustered and each cluster may have a size of C where C is an integer determined by the data center capacity planning and dimensioning teams. Each cluster has associated networking and other power equipment. Shutting down a cluster is quite interesting as it saves greater power consumption as compared to shutting down a node. When the cluster is shut down related networking components and power components are also shut down resulting in greater saving of power. The criteria to shut down a cluster are simple. If all the cluster nodes are down and $CC \ll CTI$ in their interval the cluster will be shut down.

```

Active_Servers[];
Server_Request_Capacity = Sc
Total_ServerFarm_Request_Capacity = Fc

Request_Intensity_in_Current_BP = R

Excess_Capacity = Ec
Excess_Servers = Es

Selected_Servers = SS
Running_Servers = RS

Foreach(bp in BP) {
    If (Fc > [R + K]) {
        Ec = Fc - R
        Es = Ec / Sc
        Select (n = Es) Servers = SS
        Transfer_Load (SS, RS)
        Schedule_All_Future_Request_in_BP_to_RS();
    }
    Else if (Fc < [R + K]) {
        Required_Capacity = R - Fc + K
        Required_Servers = Round (Required_Capacity / Sc)
        Select OFF-SERVER[] = OFFSERVERS

        Foreach (OS IN OFFSERVERS) {
            switchOn();
        }
        Schedule_All_Future_Request_in_BP_to_RS();
    }
}

```

Figure 1: Server Addition/Removal Algorithm

```

Foreach (as in AS) {
    If (CurrentLoad(as) < 90%) { CS->addserver(as)
    }

    While (size(Request) > 0) {
        Foreach (r in requests) {
            Foreach (cs in CS) {
                r->redirect(cs);
            }
        }
    }
}

```

Figure 2: Request Allocation

This operation also takes into consideration the startup and shutdown times of the nodes as well as the networking and power equipment.

3.6. Start Cluster

Cluster start is triggered when the off-pool is exhausted and there are off-clusters. This follows the values of the monitor and CTI. If $CTI \gg CC$ and off pool = 0 then a new cluster is triggered to start. The cluster start time is also considered in this operation.

All the new requests are managed by the load balancer.

3.7. Request Distribution Algorithm

In previous sections, we have discussed the basic concepts and terminology. Now this section covers the working of an algorithm to distribute the incoming requests to server nodes and select an optimal number of servers to serve the traffic load. The algorithm compares the total server farm

capacity against the capacity required to serve the TI in each BP.

It calculates the excess servers and based on the excess capacity and initiate additional server switch on or switch off. All these decisions include the K variable that accommodates the servers boot up period and request redirection time. Only new requests are scheduled for the new allocation. The scheduling algorithm works in such a way that it selects the servers with the least running load with a threshold of 90% load. If the load approaches 90% new requests for a switch on are made.

4. RESULTS

In this section, we discuss the simulation details of the comparison environment.

The simulation consists of three parts as below.

- Virtual Datacenter (VDC)
- Controller & Power Analyzer (C&PA)
- Simulated Network Environment (SNE)

VDC provides sixteen server nodes arranged in 4 clusters each containing 4 server nodes. Each node runs a specialized web application that responds to the user requests on certain TCP ports after processing some information as requested by the user. VDC is connected to a simulated environment that uses an NS3 simulator. The network topology is planned as per the capacity planning using the Traffic Intensity Principles discussed in earlier sections of this paper.

Traffic is generated from an NS3 simulated network topology following the selected traffic intensity profile. In the simulated network, users are distributed in 16 Point of Presence Points (POPs) where each POP has different traffic intensity profiles. Different types of access POPS (Max of 4) are aggregated at aggregation POP. Mixing of the different types of POPs in the aggregation is very important as it allows better utilization of the network resources

Figure 3: Simulation Environment

Figure 4: Power Consumption of Virtual

It also allows the performance enhancements if the traffic intensity based Packet Scheduling or Queue Management Algorithms like TIPS [R1]. Anyhow here we assume that all the network topology follows a commonly known architecture and traffic sources or users are placed randomly.

4.1 Simulation

The above algorithm was simulated in a virtual datacenter build with containers. For the request generation, we used NS3 to generate requests with the required characteristics. Each server node is run in a Linux container having a web server, MySQL database, and some other minor utilities for housekeeping and analysis purposes. The controller node is also implemented in a Linux controller. Controller and NS3 modules are fed with the traffic intensity information via a shared resource.

Total Access Switches: 16 (corresponding to 16 access POPs)

Nodes

Free Running Power Consumption = 250mW

10%load Per Percentage Wattage = $50/10 = 5W$

10 to 70% = $400 - 250 = 150/70$

Server load is monitored in each sampling period. Server load is analyzed for power consumption per unit of server load. The first approach uses a predefined traffic load with round-robin server allocation Second approach uses the same predefined load with a traffic intensity based server allocation algorithm. For both approaches, the server load is measured in each sampling period. Corresponding power consumption was calculated for each sampling period. Figure 6 and 7 shows the power consumption of datacenter nodes. These results are captured with the random load without considering the CTI values and TISC being not operational. Different colors of the graph represent different datacenter nodes and the Y-axis represents the power consumption and the X-axis shows the time line.

Figure 5: Power consumption of datacenter nodes

Figure 6: Power consumption of datacenter nodes with random load

Figure 7: Comparison of the energy consumption with traffic intensity based controller and without controller

Figure 8: Comparison of the energy consumption with traffic intensity based controller and without Controller

The comparison of the energy consumption with traffic intensity based controller and without it is shown in figure 8 and figure 9. Cumulative energy utilization in the datacenter is shown for both cases in figure 10. The energy curve for the datacenter with TISC always remains below the energy curve of the datacenter without TISC. TISC results in a significant reduction in power consumption and the idle servers running. This also enables a reduction in the idle network and other related infrastructure. However, the impact of these is not included in this work.

The overall difference in energy saving is almost half of the traditional data center approaches. However, if the networking and other infrastructures are added to the analysis, it will result in large energy savings.

4.2. Virtual Container Environment (VCE)

VCE provides a data center function. The datacenter nodes are Docker-based containers that run various services including a WebApp, Database server, and related Linux packages.

Figure 9: Energy consumption with traffic intensity based controller and without Controller

All the datacenter nodes are homogeneous i.e. all nodes provide the same functionality or same service. They have a similar CPU, Memory, and Storage Specification, so that when a user requests a service, it may be allocated to any of the servers usually having the least load.

4.3. Network Simulation Environment (NSE)

NSE is NS3 based simulation environment that provides network topology to simulate the model network and generate requests that are routed to VCE.

4.4. Traffic Intensity Based Container Controller (TIC)

TIC is a traffic intensity based container controller. Its job is to monitor the data center nodes and their load and it decides whether a node has to be switched off or switched on and hence providing a basis for saving energy.

TIC has two important functions one is to ensure that it has correct traffic profile information and the second it has correct node information and its load and estimated remaining processing time. TIC also schedules the requests and assignments to serve nodes. There are different approaches for the scheduling and allocation of requests to servers e.g. priority, RR, WRR, etc. in this research simple round-robin methodology was adopted among the least loaded and on data center nodes.

4.5. Container Power Analyzer (CPA)

CPA's role is to calculate the power consumed based on the predetermined calibration of power

consumption behavior on the host system. CPA records the power consumption per minute that is further used for comparison purposes.

4.6. Traffic Model

Users connected to access switches generate requests for server resources and generate load as per user requirements. The traffic intensity profile for these nodes is randomly chosen with no predetermined rule. However, while designing the network as well as data center capacity planning and dimensioning Erlang principles are used so that the user load can be efficiently met by the network & data centers. The specific details of capacity planning and dimensioning shall not be covered in this paper rather a generic Erlang model is referred to.

Figure 11 shows the traffic model of uplinks of the access nodes of datacenters. It represents several requested generated by users corresponding to the given access node. Figure 12 shows the same values but in a different view

#/Sample	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Node 1	10	10	10	10	0	0	0	0	0	0	18	18	18	18	18	2	2	2	2	2
Node 2	5	5	5	5	10	10	10	10	10	10	15	15	15	15	15	4	4	4	4	4
Node 3	4	4	4	4	15	15	15	15	15	15	20	20	20	20	20	5	5	5	5	5
Node 4	12	12	12	12	18	18	18	18	18	18	10	10	10	10	10	6	6	6	6	6
Node 5	15	15	15	15	10	10	10	10	10	10	5	5	5	5	5	7	7	7	7	7
Node 6	5	5	5	5	8	8	8	8	8	8	15	15	15	15	15	8	8	8	8	8
Node 7	4	4	4	4	2	2	2	2	2	2	10	10	10	10	10	10	10	10	10	10
Node 8	3	3	3	3	5	5	5	5	5	5	20	20	20	20	20	3	3	3	3	3
Node 9	2	2	2	2	6	6	6	6	6	6	18	18	18	18	18	18	4	4	4	4
Node 10	1	1	1	1	7	7	7	7	7	7	8	8	8	8	8	5	5	5	5	5
Node 11	2	2	2	2	2	2	2	2	2	2	9	9	9	9	9	4	4	4	4	4
Node 12	4	4	4	4	1	1	1	1	1	1	10	10	10	10	10	4	4	4	4	4
Node 13	5	5	5	5	2	2	2	2	2	2	15	15	15	15	15	8	8	8	8	8
Node 14	10	10	10	10	3	3	3	3	3	3	4	4	4	4	4	8	8	8	8	8
Node 15	2	2	2	2	5	5	5	5	5	5	3	3	3	3	3	2	2	2	2	2
Node 16	1	1	1	1	6	6	6	6	6	6	1	1	1	1	1	1	1	1	1	1

Figure 10: Traffic model of uplinks of the access nodes of datacenters

Figure 11: Graphical view of traffic model of uplinks of the access nodes of datacenters

Table 1: Traffic Intensity

Total Power with TISC	1306981	33.40%
Total Power without TISC	2606053	66.60%
Total	3913034	49.85%

4.7. Characteristics of the Traffic Model

The particular characterization of the traffic model selected was that nodes generate traffic in a randomly distributed fashion and the node distribution is also sparse. Second important thing is that traffic generated from different access nodes never exceeds the max limit assisted to the access pop. This ensures that the network capacity planning and dimensioning have been carried out in a way to cater the maximum needs of the users.

4.8. Traffic Intensity Information:

Consider is the traffic intensity of the node n and be the sampling time, be the average session time, be the actual requests sent from a node, be the request serving capacity of a single node. Then, We can define an access node, $TI_n = \frac{RA * TSES}{CR * TS}$,

We have selected a hypothetical request model where user requests are randomly distributed across the sampling intervals. The selected request model and the corresponding traffic intensity are given in table 1 and table 2 respectively.

Figures 11 and 12 show the traffic intensities corresponding to the traffic intensity of each access pop and it shows that some characteristics of the traffic are preserved.

5. EVALUATION METHODOLOGY

This section discusses the evaluation and comparison methodology of the simulation results. The simulation is carried out in three steps. In the first step, a predefined traffic model is used for single server single access pop. In this phase following relationships are established.

5.1. Phase -1

- requested generated by users
- Request completion time
- Respective CPU load

- Respective power consumption by for node
- Respective power consumption by the network interfaces

5.2. Phase -2

In the second phase, all the access pops generate random traffic, and requests are served in a routine procedure. During this step, the following are observed.

- The total requests generated by users
- Total requests served by the datacenter
- Average request completion time

5.3. Phase -3

In phase 3 access nodes generate traffic in a predefined pattern governed by the traffic intensity profile and the controller applies the traffic intensity-based algorithm to route the requests to nodes based on the load.

This research work focuses on an abstract level comparison of the overall energy utilization in the datacenter.

5.4. Phase -4

It calculates the energy consumption traditionally without implementing any energy optimization algorithm. In this case, all servers are switched on and incoming requests are allocated in the round-robin fashion. Some servers in the case may be overloaded and some servers may be under loaded due to the different amount of computation or request time. Since there is no way of considering the traffic intensity so requests are allocated in a simple round robin fashion regardless of the load characteristics.

5.5. Phase -5

In this phase traffic intensity based energy efficiency algorithm is implemented that tracks the current server utilization, evaluates the current TI characteristics, dynamically adds and removes the active servers. The requests are allocated in a round robin fashion among the candidate servers rather than all servers.

Both phases are run for a fixed duration and server load, power consumption, memory utilization, request completion rate, and request completion latencies are monitored and compared with each other.

Figure 12: average request completion time with respect to server load

Figure 13: Servers Utilization with Random Load without Traffic Intensity Based Server Controller (TISC)

Figure 13 shows the average request completion time concerning server load. It shows that as the server load increases request completion time only changes after a certain threshold.

6. DISCUSSION

Finally, the percentage of energy consumption as observed without implementing Traffic Intensity Based Server Controller 66.60%. While, we implemented it Traffic Intensity Based Server Controller 33.40%. The energy efficiency achieved in this model is 49.85%.

7. CONCLUSION

This work analyzed the impact of Traffic intensity-based energy efficiency for a modern datacenter. All the analysis was based on a virtual data center simulated using Linux containers running various services like file transfers, voice sessions, and email and webs services. The results presented in this work show that TISC based approach in modern datacenter results in significant energy savings, esp. due to the elimination of the idle servers and idle infrastructure. Further TISC based load balancing results freeing up more servers that are underutilized. Both of these approaches result in a significant reduction in power consumption.

In this work, the analysis of the networking and related infrastructure was not considered that apparently can improve the results due to known idle components available to turn off or hibernate. Furthermore, in this work, we did not consider the

hibernation on the servers that may eliminate the issues with server startup and shutdown. The restart operation on servers has its side effects.

REFERENCES

- [1] Khan, S.U.; Zomaya, A.Y. *Handbook on data centers*; 2015.
- [2] Heller, B.; Seetharaman, S.; Mahadevan, P.; Yiakoumis, Y.; Sharma, P.; Banerjee, S.; "McKeown, N. ElasticTree: Saving Energy in Data Center Networks." In Proceedings of the USENIX Symposium on Networked Systems Design and Implementation, NSDI 2010; USENIX Association Berkeley, CA, USA: 28-30 April 2010; San Jose, California, pp. 249–264,2010.
- [3] Heller, B.; Seetharaman, S.; Mahadevan, P.; Yiakoumis, Y.; Sharma, P.; Banerjee, S.; "McKeown, N. ElasticTree: Saving Energy in Data Center Networks. *Proc. 7th USENIX Conf. Networked Syst. Des. Implement.*, 17–17, 2010.
- [4] Shang, Y.; Li, D.; Xu, M. "Energy-aware routing in data center network." In Proceedings of the Proceedings of the first

- ACM SIGCOMM workshop on Green networking - Green Networking '10; ACM: 30 August - 03 September 2010; New Delhi, India, pp. 1–8, 2010.
- [5] Nguyen, K.K.; Cheriet, M.; Lemay, M.; Savoie, M.; Ho, B. "Powering a data center network via renewable energy:" A green testbed. *IEEE Internet Comput.*, 17, 40–49, 2013.
- [6] Arlitt, M.; Bash, C.; Blagodurov, S.; Chen, Y.; Christian, T.; Gmach, D.; Hyser, C.; Kumari, N.; Liu, Z.; Marwah, M.; et al. *Towards the design and operation of net-zero energy data centers*; 2012.
- [7] David, H.; Fallin, C.; Gorbatov, E.; Hanebutte, U.; Mutlu, O. *Memory power management via dynamic voltage/frequency scaling*; 2011;
- [8] Wang, T.; Su, Z.; Xia, Y.; Qin, B.; Hamdi, M. "Towards cost-effective and low latency data center network architecture". *Comput. Commun.*, 82, 1–12 2016.
- [9] Valancius, V.; Laoutaris, N.; Massoulié, L.; Diot, C.; Rodriguez, P. "Greening the internet with nano data centers." *Proc. 5th Int. Conf. Emerg. Netw. Exp. Technol. Conex.* 09, 37–48, 2009.
- [10] Baccour, E.; Fofou, S.; Ridha, H.; Hamdi, M. *wFlatnet: Introducing Wireless in Flatnet Data Center Network*; 2015;
- [11] Kim, J.; Dally, W.; Abts, D. "Flattened butterfly:" a cost-efficient topology for high-radix networks.; Vol. 35, 2007.
- [12] Baccour, E.; Fofou, S.; Hamila, R.; Hamdi, M. A survey of wireless data center networks. *2015 49th Annu. Conf. Inf. Sci. Syst.* 2015, 1–6.
- [13] Guan, X.; Choi, B.Y.; Song, S. "Energy efficient virtual network embedding for green data centers using data center topology and future migration. *Comput. Commun.*, 69, 50–59, 2015.
- [14] Shin, J.Y.; Sirer, E.G.; Weatherspoon, H.; Kirovski, D. On the feasibility of completely wireless datacenters. *IEEE/ACM Trans. Netw.*, 21, 1666–1679, 2013.
- [15] VMware infrastructure architecture overview.
- [16] Liu, L.; Wang, H.; Liu, X.; Jin, X.; He, W.; Wang, Q.; Chen, Y. GreenCloud: "a new architecture for green data center. In Proceedings of the Proceedings of the 6th International Conference Industry Session on Autonomic Computing and Communications" Industry Session, ACM, pp. 29–38, 2009.
- [17] Georgiou, S.; Tsakalozos, K.; Delis, A. "Exploiting network-topology awareness for vm placement in iaas clouds. In Proceedings of the Cloud and Green Computing (CGC)", 2013 Third International Conference on; IEEE, pp. 151–158, 2013.
- [18] Rui, P.; Bianco, A.; Fiandrino, C.; Giaccone, P.; Kliazovich, D. *Power Comparison of Cloud Data Center Architectures*; 2016;
- [19] Carrega, A.; Singh, S.; Bolla, R.; Bruschi, R. "Applying traffic merging to datacenter networks. In Proceedings of the Future Energy Systems: "Where Energy, Computing and Communication Meet (e-Energy), 2012 Third International Conference; pp. 1–9, 2012.
- [20] Wang, T.; Xia, Y.; Muppala, J.; Hamdi, M. "Achieving Energy Efficiency in Data Centers Using an Artificial Intelligence Abstraction Model." *IEEE Trans. Cloud Comput.*, 1–1, 2015.
- [21] Fang, S.; Li, H.; Foh, C.H.; Wen, Y.; Aung, K.M.M. "Energy optimizations for data center network: Formulation and its solution. In

- Proceedings of the Global Communications" Conference (GLOBECOM), IEEE; 2012; pp. 3256–3261, 2012.
- [22] Al-Fares, M.; Loukissas, A.; Vahdat, A. "A scalable, commodity data center network architecture." *SIGCOMM Comput. Commun. Rev.* 38, 63–74, 2008.
- [23] Berl, A.; Gelenbe, E.; Di Girolamo, M.; Giuliani, G.; De Meer, H.; Dang, M.Q.; Pentikousis, K. "Energy-efficient cloud computing. *Comput. J.*, 53, 1045–1051, 2010.
- [24] Cook, G. How clean is your cloud. *Catal. an energy Revolut.* 2012.
- [25] International, G. Greenpeace "likes" Facebook's new data center, but wants a greener friendship. 2011.
- [26] Koomey Growth in Data Center Electricity use 2005 to 2010. *Oakland, CA Anal. Press.* ,34008, 1–24, 2011.
- [27] Meisner, D.; Gold, B.T.; Wenisch, T.F. PowerNap: "eliminating server idle power." *Asplos*, 44, 205–216, 2009.
- [28] Koomey, J.G. Growth in Data Center Electricity use 2005 to 2010." *Anal. Press.* 1–24, 2011,.
- [29] Akella, A.; Benson, T.; Chandrasekaran, B.; Huang, C.; Maggs, B.; Maltz, D. "A universal approach to data center network design." In Proceedings of the Proceedings of the 2015 International Conference on Distributed Computing and Networking; p. 41 ,2015.
- [30] Tso, F.P.; Pezaros, D.P. "Improving data center network utilization using near-optimal traffic engineering". *IEEE Trans. Parallel Distrib. Syst.*, 24, 1139–1148, 2013.
- [31] Giroire, F.; Moulhierac, J.; Phan, T.K.; Roudaut, F." Minimization of network power consumption with redundancy elimination". *Comput. Commun*, 59, 98–105, 2015.