Energy Consumption Analysis of Scheduling Algorithms for Cloud Computing Systems

Patrick Hovsepian
California State University, Fullerton
College of Engineering & Computer Science
Fullerton, CA, USA
phovsepian@csu.fullerton.edu

Yun Tian
California State University, Fullerton
College of Engineering & Computer Science
Fullerton, CA, USA
ytian@fullerton.edu

***Abstract*—With more and more data and service moved to different cloud computing systems, progress in the area of energy consumption and environmentally friendly practices must be made so these advancements do not plateau. The goal of this study is to provide exceptional insight regarding which scheduling algorithms will yield environmentally sustainable, commonly referred to as 'green,' practices for energy efficiency. This will be achieved by tracking the similarities and differences of multiple scheduling algorithms, considering various datacenter topologies and their task sizes (MIPS). Trace files produced by the simulator will be used to create a visual representation of the observed data and analyze the energy consumption of the servers and switches (core, aggregation, and access).**

***Keywords*—algorithms, cloud computing, consumption, data center, energy, green, scheduling, simulation, task, topologies.**

I. Introduction

Cloud computing, in its simplest definition, can be characterized as a method of delivering hosted services and tools over a network. In recent years, the pieces of technology which compose cloud computing have rapidly developed into powerful self-sufficient tools that serve as the backbone of many IT environments. Although the state of cloud infrastructures and ecosystems continues to evolve at an astonishing rate with the help of breakthroughs in the realm of computer science, the necessary implementation of 'green practices' for all IT environments yet has to be standardized.

With evidence of global warming and the energy consumption crisis [3], the environmental impact of data centers needs to be addressed by all the various segments which belong to the information and communications technology (ICT) sector. According to the Natural Resources Defense Council, approximately 91 billion kilowatt-hours of electricity were consumed by data centers in 2013 which equates to roughly 150 million metric tons of carbon pollution [4].

There are numerous factors that contribute to the immense power consumption of data centers such as: 24/7 availability of servers, network switches, cooling equipment, systems support, and UPS (uninterruptable power supply) systems [6][7]. An extensive power management system for all of these aspects would surely reduce their consumption, but possibly at the price of their computing efficiency. An approach needs to be taken that does not jeopardize the integrity and the advantages of data centers, while being mindful of the impact they have on the environment around them.

The rest of the paper is organized as follows. In Section II we discuss the Arrangement and Procedure required our projects experimental portion. Then we will then proceed to display our observed Experimental Data in Section III. After the data has been displayed, a comprehensive explanation of Analysis and Results will be made follow by a Conclusion.

II. Arrangement and Procedure

A. Understanding the Environment

We chose GreenCloud Simulator due to its ability to produce numerous deviations of data center environments with flexible parameters. GreenCloud gives us the ability to hone in on specific attributes of a data center, while keeping the big picture in place. Not only does GreenCloud focus on a normal data center topology, but it specifically highlights cloud simulation environments. While some cloud simulation tools might be as detail oriented as GreenCloud Simulator, GreenCloud stands apart in that it is Open Source. Lastly, GreenCloud Simulator focuses on producing results of energy consumption given the simulated data center that has been chosen.

B. Understanding Parameters

Due to the detailed nature of GreenCloud Simulator, there are numerous parameters that may be configured and explained. For this experiment, we are going to talk about a few of the parameters that impact the results the most. We will focus on Servers, Tasks, and Scheduling Algorithms.

1) Server Count – The most performance altering parameter in a datacenter are the servers. By altering this parameter, we can prove that certain server configurations yield increased productivity and efficiency.

2) Task Size – Task size is an extremely important factor in cloud computing. How many clients will be making requests the data center at one time, what will be the computational demand of the clients' requests, and what is the baseline of performance expected by the client? By manipulating task size GreenCloud allows total control of what kind of requests the data center will be processing.

3) Scheduling Algorithms – Scheduling algorithms act as dispatchers of tasks to servers. Their goal is to keep process running at all times for maximum CPU utilization. This is achieved by selecting processes in memory that are ready for execution.

978-1-7281-3783-4/20/$31.00 ©2020 IEEE

C. Network Topology

Every network, regardless of its size, follows a particular design or arrangement for all of its designated nodes and physical lines. This is commonly referred to as a topology.

There are three key topologies that are used within GreenCloud Sim and they are: two-tier architecture, three-tier architecture, and three-tier high-speed architecture.

1) Two-Tier Architecture: This architecture (see Fig.2) will not be used in the experiment because modern day datacenters do not follow this topological model.

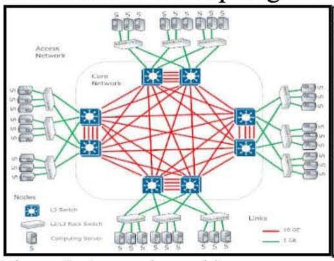

Figure 2: A two-tier architecture.

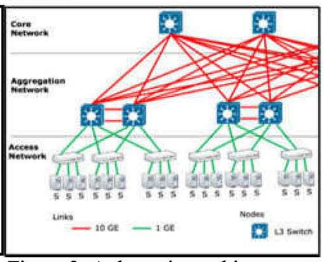

Figure 3: A three-tier architecture.

2) Three-Tier Architecture (TT): Most modern-day datacenters follow this hierarchical model for their network architecture. These architectures consist of an access layer, aggregation layer (or distribution layer), and core layer. "The availability of the aggregation layer facilitates the increase in the number of server nodes while keeping inexpensive Layer-2 (L2) switches in the access network, which provides a loop-free topology [8]." This architecture can be seen in Fig. 3.

3) *Three-Tier High-Speed Architecture (TTHS):* These architectures are designed to leverage short reach multi-mode optical fiber or high bandwidth GbE to create networks with extremely fast communication speeds. They aim to optimize node counts, core capacity, and aggregation layer switches for increased throughput [8]. Leveraging 100 GbE links between nodes to reduce hardware dependencies and bottlenecks.

The difference in network link speeds between the Three-Tier Architecture (TT) and the Three-Tier High-Speed (TTHS) can be seen in Table 1.

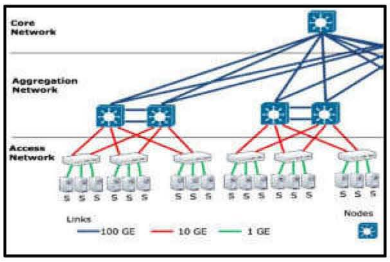

Figure 4: A visual representation of a three-tier high-speed architecture.

Table 1: Topology Link Speeds

	Core to Aggregation	Aggregation to Access	Access to Host
TT	10 Gb/s	1 Gb/s	1 Gb/s
TTHS	100 Gb/s	10 Gb/s	1 Gb/s

D. Task Scheduling Algorithms

Cloud computing from a data center requires a very particular architecture which is dependent on the virtual architecture implemented throughout the infrastructure. The basic unit of computational power within a data center is a virtual machine, which offers resources such as: CPU, RAM, and storage. Unfortunately, the resources offered by virtual machines are not unlimited and there is only so much load each one can handle. Therefore, scheduling algorithms must be used to enhance service quality when carrying out tasks, supplying the expected output on time for a given request, and control the efficiency for all tasks [9].

Typically, a cloud user will submit their request to the data center where all other tasks are in the main queue. A data center controller will map the submitted, and incoming tasks, to the host (virtual machine) that best suits requirement. All tasks must be approved before proceeding to the second layer - the network layer.

Task scheduling aims to effectively utilize all available resources and their consumption, while trying to minimize the amount time it takes to complete the tasks. "All tasks should be balanced by a task scheduler to maintain quality of service, efficiency and fairness [9]." Listed below are the five scheduling algorithms that were used to analyze and investigate energy consumption for varying task sizes.

1) Green Scheduling Algorithm – The Green scheduling algorithm performs a best effort workload consolidation on a minimum set of servers [10]. "…makes a greedy consolidation of the load: it looks for the first resource provider in the input list that can successfully finish a task. Because of that, it needs information about the current load of ResourceProviders [11]." This scheduler tracks the buffer occupancy of network switches on the path in a continuous fashion and whenever it detects congestion, it stays away from those routes even if they direct loads to servers that satisfy the computational needs of a request [9].

2) Round-Robin Scheduling Algorithm – The Round-Robin scheduling algorithm makes uniform decisions and assigns tasks in a cyclic fashion based on time quantas. This algorithm equally administers tasks among all resources, which removes any result of overhead from necessary pre-processing.

3) Random Scheduling Algorithm – The random scheduling algorithm used in the Green Cloud Simulator assigns preferred tasks to machines (servers) in a random distribution, which is uniform by default [11]. Regardless of the load, heavy or light, on the machine, the scheduling algorithm will assign the task to random server. This algorithm is not very complex and will not require any overhead of pre-processing.

4) HEROS Scheduling Algorithm – This scheduling algorithm, which stands for Heterogeneous Energy-efficient Resource allocation Optimizing Scheduler, combines the best features of DENS and e-STAB to make a vey aware decision for heterogeneous environments. HEROS's decision making capabilities derive from it's ability to calculate a server score and allocate the tasks to servers with the highest scores [11]. Two main components drive and calculate a higher score: the server selection function and the communication potential function. The server selection function is calculated from numerous factors such as: power consumption (PpW – performance per Watt), performance given a certain MIPS load, maximum server load, and maximum acceptable load [11]. The communication potential function is based off the DENS communication load which takes into account actual and current link loads [11].

5) BestDENS Scheduling Algorithm – The DENS scheduling algorithm, which stands for Data Center Energy-efficient Network-aware Scheduling, presents a method that focuses on combining the efforts of energy efficiency and network awareness [10]. By balancing energy consumption, individual job performance, and traffic demands, the DENS approach optimizes the tasks trade off and variation of traffic patterns [10]. "The DENS methodology minimizes the total energy consumption of a data center by selecting the best-fit computing resources for job execution based on the load level and communication potential of data center components [10]." Fig. 5 displays the textual description of the DENS Algorithm.

Figure 5: DENS algorithm [10].

DENS Algorithm

Initialization
  set weighted coefficient $\alpha = 0.7$, $\beta = 0.2$, $\gamma = 0.1$
  set proportional coefficient $\varphi = 2$
  get server load l
  get queue size at access and aggregate switches q
Server selection
  FOR all servers **DO**
    compute server load $L_s(l)$, rack load $L_r(l)$, and module load $L_m(l)$
    compute communications potentials of rack $Q_r(q)$ and module $Q_m(q)$
    compute metric factors related to servers $f_s(l,q)$, racks $f_r(l,q)$, and modules $f_m(l)$
    compute DENS metric as a weighted sum of $f_s(l,q)$, $f_r(l,q)$, and $f_m(l)$
  ENDFOR
  Select server with highest DENS metric

E. *Environment Setup & Configuration*

To set up GreenCloud Simulator, we installed VirtualBox on our local machines to run a virtual machine. We then installed the most current version of Ubuntu to VirtualBox to be the host operating system for our simulation environment. We configured GreenCloud to our specifications by altering all parameters that we need to change. Due to the fact that this environment will be computing large simulations for an ensemble of metrics, we allocated half the compute and half the memory resources of the local machine to virtual machine for optimum performance. The simulation code is written in Java and can modified in the Eclipse IDE for Linux based operating systems. After every simulation, trace files are created in the directories of the installed simulator which can parsed through for valuable metrics.

1) *Host Machine Specifications:* The host machine being used in this experiment needed to have a significant amount of resources to accommodate the native operating system and the guest operating system. An enterprise grade HP workstation with hyper threading was used to handle this task. The resource allocation for the type II hypervisor in this experiment can be deemed negligible. In order to further increase computational capabilities, the host operating system, Window 7 Professional, was switched to performance mode. The host machine's specifications can be viewed in Table 2.

Table 2: Host machine specifications.

OS	CPU Type	CPU	Memory
Windows 7 64-bit	Intel Xeon E5-2680 v3	24 cores @ 2.5GHz	32 GB DDR4 @ 1064.2 MHz

2) *Virtual Machine Specifications:* Simulations in the GreenCloud Virtual Machines are very demanding and require large periods of uninterrupted time for the processor. Running simulations for server counts that range anywhere from 10 – 399 are relatively easy for the virtual machine. Increasing the range count to 400 – 999, produces a noticeable drop in the speed of the simulation and the responsive of the virtual machines. Further increasing the range to 1000+, essentially renders the virtual machine as unusable because all available resources are dedicated to simulations. The virtual machine's specifications can be viewed in Table 3.

Table 3: Virtual machine specifications.

OS	CPU Type	CPU	Memory
Ubuntu 12.04	Intel Xeon E5-2680 v3	12 cores @ 2.5GHz	16 GB DDR4 @ 1064.2 MHz

III. Experimental Data

For our data centers' simulations, we used many different combinations of task sizes, server counts, and network topologies. These combinations let us look into the granular details of each data center size and hypothesize the best scheduling algorithm or topology based on the data.

A. *Experiment Variables*

The following section will give an in-depth explanation of the variables which were used for the various simulations.

1) Server Count: We based the size of 'small,' 'medium,' and 'large' data centers from industry case studies and data sheets. A small data center would be considered as a startup company, or a small company that does cloud business with only a few customers. We simulated four flavors of small data centers with a difference of 15 servers per flavor. A medium data center would be considered as a normal cloud based business that deals with hundreds, if not thousands of customers on a daily basis. While the use of many servers is a necessity for these companies, they do not need thousands of servers to cater to their customers. We simulated four flavors of medium data centers with a difference of 150 servers per flavor. A large data center would be considered as corporate companies that need vast amounts of computations and storage done on an hourly, if not minute, basis. Not only are thousands of servers required, but also well maintained and up to current specifications. We simulated three flavors of large data centers with a difference of approximately 500 servers per flavor. The server counts for the various datacenter sizes can be viewed in

Table 4: Server counts for various datacenter sizes.

Small DC	Medium DC	Large DC
30	300	1008
45	450	1512
60	600	2016
75	750	

Server count is a major component for energy usage in a data center because each server requires a minimum amount of power to run, even if not being used. That means a data center will operate, at a minimum, of the equation displayed below.

$(Number\ of\ servers)\ x\ (kWh\ per\ server)\ x\ 24\ hours \qquad Eq. 1$

2) Task Size: We based the size of our scheduling tasks on how much work we thought each server would need to go through every second. Green Cloud Simulator uses task loads in sizes of MIPS (Millions of Instructions Per Second). We chose three different task sizes for each of the server sizes for each data center. Besides the number of servers operating in a given data center, the MIPS tasks are the second most important metric to consider when computing energy usage, as every time an instruction goes through the data center, energy is being used to send that instruction. A pulse going through an individual server may not seem like a high amount of energy being used, but as you multiply the cost of each pulse by the thousands of millions, it adds up. That being said, the amount of work required by the data center directly correlates to the efficiency of the data center. The task sizes for the various server counts can be viewed in Table 5.

Table 5: Task sizes for various server counts

Task Sizes
10,000 MIPS
100,000 MIPS
1,000,000 MIPS

Table 6: Base values

Base values (b)
2
3
4
5

3) Network Topologies: It comes as no surprise that not all networks are built the same. They follow certain topologies, architectures, and best practices, but certain use cases might require significantly faster communication. That is why the three-tier (TT) and three-tier and three-tier high-speed (TTHS) topologies utilize varying switch counts and network speeds. For this experiment, we created TT and TTHS based upon two models – the Squared Switch Model (SSM) and the Cubed Switch Model (CSM). Each model also was designed with four varying base values.

These models were designed to preserve the total server count between each topology, while only varying the core, aggregation, and access switch count. The base value plays an important role in determining the core switch count because all values are dependent on this layer. The aggregation layer was strictly dependent upon the number of core switches and a fixed constant. The access layer will be explained in detail below. Tables 6, 7, and 8 display the base values, SSM formulas, and CSM formulas respectively.

Table 7: Squared Switch Model formula

	$TT_{squared}$	$TTHS_{squared}$
Core	$x_1 = b$	$x_2 = b \div b$
Aggregation	$y_1 = x_1 \times 2$	$y_2 = x_2 \times 2$
Access	$z_1 = y_1 \times b^0$	$z_2 = y_2 \times b^2$
Servers	$z_1 \times x_1 \times 3$	$z_2 \times x_2 \times 3$

Table 8: Cubed Switch Model formula

	TT_{cubed}	$TTHS_{cubed}$
Core	$x_1 = b$	$x_2 = b \div b$
Aggregation	$y_1 = x_1 \times 2$	$y_2 = x_2 \times 2$
Access	$z_1 = y_1 \times b^1$	$z_2 = y_2 \times b^3$
Servers	$z_1 \times x_1 \times 3$	$z_2 \times x_2 \times 3$

The base value determines the initial ratio of core switches between TT and TTHS. For example, if the base value for the formula is 3, there will be a there will 3 core switches for the TT architecture and 1 core switch for the TTHS architecture. Due to the budgetary constraints data centers might have when trying to achieve these switch ratios, the upper limit of the base value was determined to be 5.

The aggregation layer is generally dependent on the number of core switches, hence the fixed multiplicative factor of 2 for both models. In order to achieve the same number of servers for both models, the aggregation formula needed to remain static.

Both the models were named after the exponential factor of the base value within each access layer switch formula. The initial exponent factor for each architectures access switch formula was dependent on the model being utilized. For the SSM, the initial exponent for the TT architecture was 0 whereas the initial exponent for the CSM TT architecture was 1.

The constant for the servers' formula represents the number of servers per rack – 3. Each rack requires one access switch. This factor primarily affects the TTHS topology because of its requirements to maintain high communication bandwidth between all three layers. Hence, the higher switch counts at the access level.

Although there were numerous variables involved throughout testing, there were two parameters that served as the control factors among the models and architectures – the scheduling algorithm and the task size. Round Robin was chosen as a control factor because of its simplicity as a scheduling algorithm. A task size of 300,000 MIPS was chosen as a control factor because it represents a load that is slightly above average for server requests. These factors can be viewed in Table 9.

Table 9: Control factors among both models & architectures

Scheduling Algorithm	Task Size (MIPS)
Round Robin	300,000

B. Small Data Center Simulations

For the small data center, we simulated each of the three task sizes with each of the three data center sizes. In Fig. 6, we can see how much TOTAL ENERGY is used for each of the different server counts when using each of the sizes of MIPS instructions.

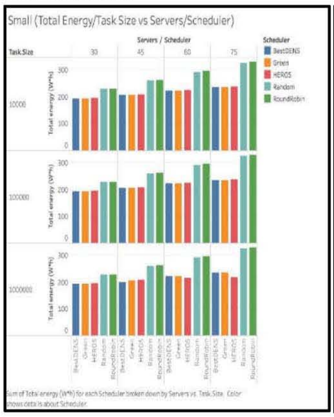

Figure 6: Total energy usage for all small data center simulations.

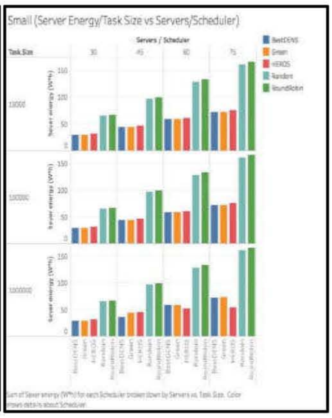

Figure 7: Server energy usage for all small data center simulations.

In Fig. 7 we are viewing the same server count and MIPS instruction count, but with SERVER ENERGY instead of total energy usage.

C. Medium Data Center Simulations

After performing 60 simulations of the small data center, we moved to 60 simulations of the medium data center. Once again, we simulated each of the three task sizes with each of the three data center sizes. In Fig. 8, we can see how much TOTAL ENERGY is used for each of the different server counts when using the variations of the MIPS instructions.

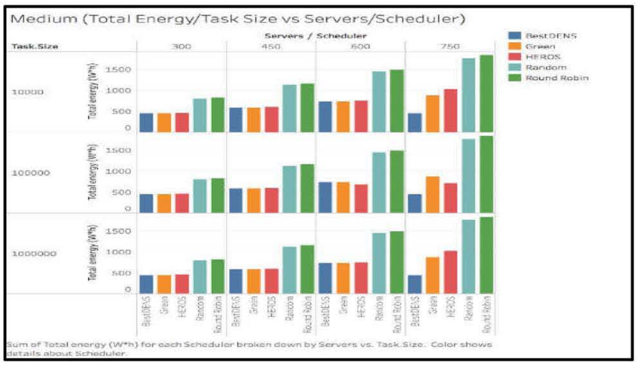

Figure 8: Total energy usage for all medium data center simulations.

In Fig. 9 we look at the same server count and MIPS instruction count, but with SERVER ENERGY usage.

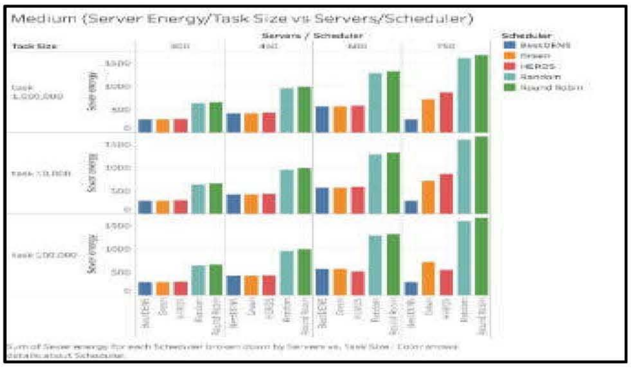

Figure 9: Server energy usage for all medium data center simulations.

D. Large Data Center Simulations

After performing 120 total simulations with small and medium data center sizes, we discovered the runtime of the simulations took extraordinarily long as the server count went up. For the large data center size, we were only able to perform 15 simulations, due to the large number of servers in each simulation. In Fig. 10 we can see how much TOTAL ENERGY is used for each of the different server counts when using 100,000 MIPS instructions.

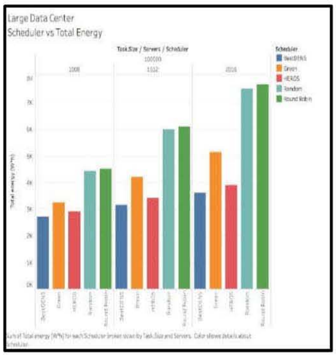

Figure 10: Total energy usage for all large data center.

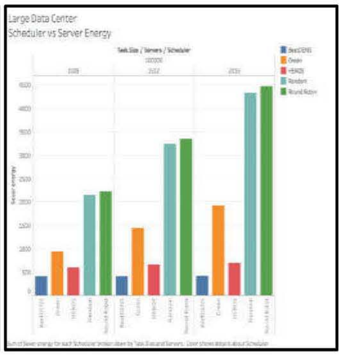

Figure 11: Total energy usage for all large data center (Server Energy)

In Fig.11 we see the same server count and MIPS instruction count, but with SERVER ENERGY usage.

E. Network Topology Simulations:

For the comparison of TT and TTHS, we simulated the various core, aggregation, and access switch counts using the SSM and the CSM. Eco-Friendliness describes the topology that consumed less TOTAL ENERGY. Fig. 12 displays the total count of Eco-Friendly Topologies for each model.

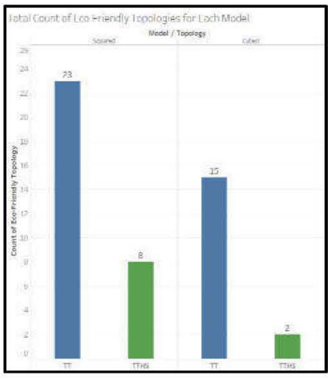

Fig. 12: Total count of Eco-Friendly topologies for each model.

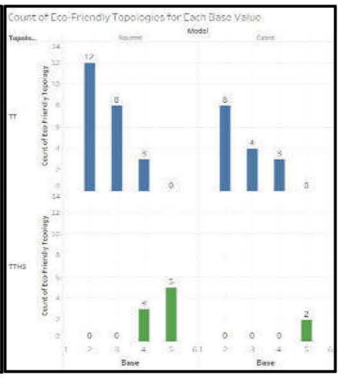

Figure 13: Count of Eco-Friendly topologies for each base in each model.

Fig. 13 displays the total count of Eco-Friendly Topologies for each base value for each model.

Fig. 14 displays the average energy difference for each base value for each model.

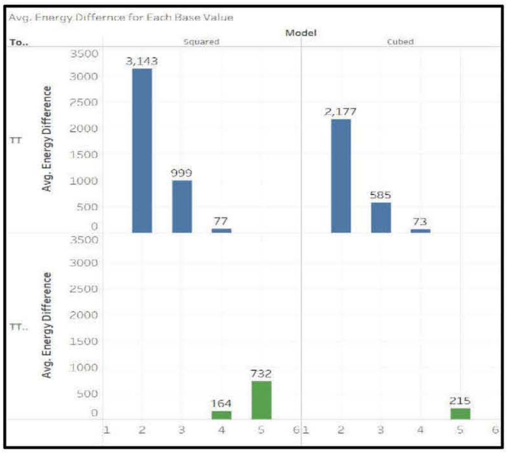

Figure 14: Total count of Eco-Friendly topologies for each model.

Fig. 15 displays the total server count for each base value for each model.

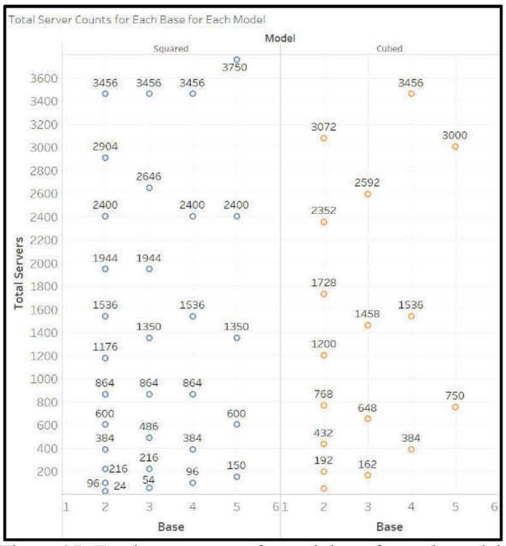

Figure 15: Total server count for each base for each model.

This concludes the Experimental Data Section.

IV. Analysis & Results

Before conducting this experiment, we had made some hypotheses regarding the outcome of energy output with respect to the scheduling algorithms and task sizes. We prepared for our experiments by setting up parameters for the different data center sizes. Our small data center was composed of 30-75 servers and had task sizes of 10,000 - 1,000,000 MIPS. Our medium data center was composed of 300-750 servers and had task sizes of 10,000 - 1,000,000 MIPS. Our large data center was composed of 1008 - 2016 servers and had a fixed task size of 100,000 MIPS. After running numerous simulations and compiling all of our data, we now had sufficient information to determine whether or not our expected results would be confirmed.

A. Algorithms vs. Tasks Sizes

In this subsection, we will examine the influence each scheduling algorithm had on the amount of server energy expended when working with certain task sizes.

1) Small Data Center – The Round-Robin scheduler was easily the worst performing scheduling algorithm when used with task sizes of 10,000, 100,000 and 1,000,000 MIPS as it required 464.1 Wh (Watt x hours) of server energy to perform the different tasks. The Random scheduler followed close behind as it used up 451.6 Wh exactly to finish the job for all three task sizes. The HEROS algorithm used 213.6 Wh for a task size of 10,000 MIPS, 213.4 Wh for a 100,000 MIPS task size, and 182.6 Wh for a 1,000,000 MIPS task size. The Green algorithm utilized 205.3 Wh for a task size of 10,000 MIPS, 205.3 Wh for a 100,000 MIPS task size, and 206.3 Wh for 1,000,000 MIPS task size. BestDENS needed 205.3 Wh for a task size of 10,000 MIPS, 205.3 for a 100,000 MIPS task size, and 197.7 Wh for a 1,000,000 MIPS task size. The best performing scheduler for a small data center managing a task size of 10,000 - 100,000 MIPS is a tie between the BestDENS and the Green scheduling algorithms. The best performing scheduler for a small data center managing a task size of 1,000,000 MIPS is the HEROS scheduling algorithm.

2) Medium Data Center – Once again, the Round-Robin scheduler was the worst performing algorithm as it used 4645 Wh of server energy to manage task sizes of 10,000, 100,000 and 1,000,000 MIPS. The Random algorithm required 4480 Wh for a task size of 10,000 MIPS, 4480 Wh for a 100,000-task size, and 4481 Wh for a 1,000,000 MIPS task size. The HEROS scheduler needed 2169 Wh for a task size of 10,000 MIPS, 1776 Wh for a 100,000 MIPS task size, and 2169 Wh for a 1,000,000-task size. As for the Green scheduling algorithm, it used 1994 Wh for all three task sizes. The BestDENS algorithm only used 1576 Wh of server energy to perform all three task sizes. The final verdict is that BestDENS is the absolute best scheduling algorithm for a medium data center managing task sizes between 10,000 and 1,000,000 MIPS.

3) Large Data Center – The Round-Robin scheduler was once more the worst performing algorithm of the 5 and used 10034 Wh of server energy for a task size of 100,000 MIPS. The Random scheduling algorithm was a close second worst as it required 9703 Wh in server energy for the 100,000 MIPS task size. The HEROS algorithm however only needed 1969 Wh of server energy with the 100,000 MIPS task size. As for the Green scheduler, it used 4334 Wh to manage the tasks. Finally, just as like in various other circumstances, BestDENS emerged victorious as it only used 1238 Wh of sever energy.

B. Algorithms vs. Data Center Sizes

In this subsection, we will the analyze the effect each scheduling algorithm had on the amount of server energy used when working with different data center sizes and different server sizes.

1) Small Data Center – The Round-Robin scheduler used 199.8 when working with 30 servers, 297.9 when used with 45 servers, 398.1 with 60 servers, and 496.5 with 75 servers. The Random algorithm spawned similar results as it used 194.7 Wh of server energy with 30 servers, 290.4 Wh with 45 servers, 386.7 Wh with 60 servers, and 483 Wh with 75 servers. The HEROS scheduler required 94 Wh with 30 servers, 137.3 Wh with 45 servers, 173.4 Wh with 60 servers, and 204.9 Wh with 75 servers. The Green scheduler was almost neck and neck with HEROS because it used 89.1 Wh with 30 servers, 132.8 Wh with 45 servers, 176.6 Wh with 60 servers, and 218.4 Wh with 75 servers. Then there was BestDENS using 89.1 Wh with 30 servers, 125 Wh with 45 servers, 176.4 Wh with 60 servers, and 217.8 with 75 servers. BestDENS and Green tied for best scheduler for small data centers with 30 servers and BestDENS performed better than all algorithms for small data centers with 45 - 60 servers. Somehow the HEROS algorithm was able to beat out the competition in a small data center with 75 servers. The worst scheduler was none other than the Round-Robin scheduling algorithm.

2) Medium Data Center – In this data center size, the Round-Robin scheduler used 1985 Wh with 300 servers, 2977 Wh with 450 servers, 3972 Wh with 600 servers, and 5000 Wh of server energy with 750 servers. The Random algorithm managed to follow very close behind as it used 1924 Wh for 300 servers, 2874 Wh with 450 servers, 3838 Wh with 600 servers, and 4806 Wh of server energy alone with 750 servers. HEROS outperformed the previous two and used 877 Wh for 200 servers, 1294 Wh with 450 servers, 1673 Wh with 600 servers, and 2269 Wh for 750. Green's regulations show that it requires 859 Wh for 300 servers, 1274 with 450 servers, 1707 with 600 servers, and 2140 with 750 servers. Lastly, BestDENS followed closely with 860 Wh with 300 servers, 1274 with 450 servers, 1707 with 600 servers, and 860 with 750 servers. With this data, we were able to discover that BestDENS was performing just the same as the HEROS and Green were with server sizes between 860 - 1707. It came as no surprise that Round-Robin performed the worst and BestDENS debatably the best.

3) Large Data Center – The largest differences between the schedulers and energy efficiencies are best visualized in this set of simulations. The Round-Robin scheduler used 2222 Wh for a large data center with 1008 servers, 3347 Wh with 1512 servers, and 4465 Wh with 2016 servers. The Random algorithm spent 2145 Wh of server energy with 1008 servers, 3241 Wh with 1512 servers, and 4318 Wh with 2016 servers. In HEROS we begin to see just how significantly less server energy is consumed because it utilized 602 Wh with 1008 servers, 664 Wh with 1512 servers, and 703 Wh with 2016 servers. The Green scheduler, though not the best, continued to show its colors by making use of 952 Wh with 1008 servers, 1450 Wh with 1512 servers, and 1932 Wh with 2016 servers. Ultimately, BestDENS showed how important it is to use less energy consuming scheduling algorithms as it used 413 Wh with 1008 servers, 408 Wh with 1512 servers, and 418 Wh with 2016 servers. With the large data center variables, we worked with, it was obvious that BestDENS and the HEROS algorithms ran victory laps around the other schedulers but BestDENS still reigned victorious.

C. Three-tier vs. Three-tier high-speed

In this subsection, we will examine the influence each model had on the amount of energy used for each topology.

1) Squared Switch Model – With respect to 31 distinct server counts, 74% of TT topologies in the squared model yielded Eco-Friendly results in comparison to the TTHS topology. Examining the counts of the Eco-Friendly topologies with respect to base values reveals a very interesting correlation – the larger the base value, the more likely that TTHS will yield an Eco-Friendly result. In Fig. 13, the counts for the Eco-Friendly Topologies even out at the base value of 4. At the base value of 5, only TTHS began yield Eco-Friendly Topologies. The cause of this correlation is strictly attributed to the fact that the higher base values are requiring for a lower count of core switches and a lower count of aggregation switches while preserving the total server count. Fig. 14, supports this observation as the average energy difference between a TT and a TTHS topology increases per base value. The energy consumption between the topologies at base value 4 is almost negligible because the average between them is significantly low. A total of 62 simulations were completed using the squared switch model, with a variety of total server counts which can be seen in Fig. 15.

2) Cubed Switch Model – With respect to 17 distinct server counts, 88% of TT topologies in the cubed model yielded Eco-Friendly results in comparison to the TTHS topology. Examining the counts of the Eco-Friendly topologies with respect to base values reveals the same correlation that was evident in the squared model – the larger the base value, the more likely that TTHS will yield an Eco-Friendly result. In Fig. 13, the transition to Eco-Friendly TTHS topologies does not occur until the base value of 5. Observing Fig 14, reveals the reasoning behind the delayed the transition. The average energy difference between both topologies was a mere 73 kWh, which means both topologies were very similar in terms of energy consumption for the given server counts of 384, 1536, and 3456. There was a total of 34 simulations using the cubed switch model. Unfortunately, the gaps between equivalent server counts were much more drastic between due to formula being utilized for this model. There were not as many equal server counts across the base values as there were for the squared model (e.g. 3456, 2400, 864).

D. Overall Comparison

Based on the data displayed above, our expected results were confirmed, but there were quite a few interesting behaviors observed. After all the data was aggregated and visualized, certain filters applied to the data revealed areas of the greatest energy consumption.

The Round-Robin and Random scheduling algorithms yielded the highest energy consumption across all ranges of server count and task sizes. The biggest factor in deciding the effectiveness of the scheduling algorithm in relation to energy consumption was the server size. As observed, the difference in the total and server energy consumption for small datacenters became more evident as the server size increased. For small data centers, BestDENS and Green algorithms performed the best; the difference between these algorithms in energy usage was almost negligible. However, we did observe one interesting change in best algorithm for small data centers with a larger number of servers and larger task sizes, where HEROS outperformed the other two. In medium data centers, we observed the same trends in power consumption for each scheduling algorithm. BestDENS and Green competed for best algorithm with a small server size, but as the server size grew to medium and large, Green drifted away from BestDENS; therefore, BestDENS was the best algorithm for medium data centers. For large data centers, we once again saw BestDENS outperform the other algorithms. However, we noticed a distinctly different trend with the HEROS scheduling algorithm for large data centers vs. other size data centers. HEROS outperformed Green in every task size category, which had not been done before. These five algorithms have the chance to outperform one another, given specific data center topologies and infrastructure.

Given the data and analysis regarding the switch models, there is no doubt that a TT topology will yield greener results. In certain cases, an argument can be made for TTHS topologies because of its lower energy consumption and its significantly improved performance for network communication. There will be certain data centers that cannot sacrifice performance for better energy consumption results, but at least they may decide the most suitable switch topology to satisfy their compute requirements and moderately reduce their environmental impact. If paired accordingly, a TTHS topology with the proper scheduling algorithm may yield greener results than a TT topology with a sub-par scheduling algorithm.

V. CONCLUSION

In this paper, we conducted an experiment to assess the energy consumption of scheduling algorithms for various data center task sizes and topologies. The analysis considered five different scheduling algorithms: Green, Round-Robin, Random, HEROS, and BestDENS, while taking into account three task sizes that differed by an entire order of magnitude: 1×10^4, 1×10^5, 1×10^6. In addition, two various switch models were leveraged to compare the TT and TTHS topologies. By exploring these variables in a controlled simulation environment – GreenCloud Simulator, we were able to offer insight and recommendations to which scheduling algorithm, data center size, task size and topology combinations yielded the most environmentally friendly results in terms of power consumption measured in kWh (kilowatt hours).

REFERENCES

[1] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," NIST Special Publication, SP 800-145, pp 2. Sept. 2011.

[2] Brenton, Chris. (2012, Jul. 05). *SANS Penetration Testing: Pen Testing in the Cloud* [Online]. Available: https://pen-testing.sans.org/blog/2012/07/05/pen-testing-in-the-cloud.

[3] Riebeek, Holli. (2010, Jun. 03). Global Warming [Online]. Available: http://earthobservatory.nasa.gov/Features/GlobalWarming/printall.php

[4] J. Whiney and P. Delforge, "Data Center Efficiency Assessment," National Resources Defense Council, New York, NY, Tech. Rep. IP:14-08-A. Aug. 2014.

[5] M. Guzek, D. Kliazovich, and P. Bouvry, "A Holistic Model for Resource Representation in Virtualized Cloud Computing Data Centers," University of Luxembourg, Luxembourg, Luxembourg, 2013.

[6] S. Murugesan, "Harnessing Green IT: Principles and Practices," *IT Pro*, IEEE Computer Society, pp. 3, Feb 2008.

[7] J. Liu, F. Zhao, X. Liu, and W. He, "Challenges Towards Elastic Power Management in Internet Data Centers," IEEE Conference of Distributed Computer System Workshops, Los Alamitos, CA, 2009.

[8] B.S. Gill, S.K. Gill, P. Jan. "Analysis of Energy Aware Data Center using Green Cloud Simulator in Cloud Computing," International Journal of Computer Trends and Technology (IJCTT), Vol 5. Num. 3, 2013.

[9] S. Atiewi, S. Yussof, and M. Ezanee, "A Comparative Analysis of Task Scheduling Algorithms of Virtual Machines in Cloud Environment," College of Information Technology, Tenaga National University, Kajang, 2015.

[10] D. Kliazovich, P. Bouvry, S. U. Khan, "DENS: data center energy-efficient network-aware scheduling," University of Luxembourg, Luxembourg, Luxembourg, 2011.

[11] M. Guzek, D. Kliazovich, and P. Bouvry, "HEROS: Energy-Efficient Load Balancing for Heterogeneous Data Centers," University of Luxembourg, Luxembourg, Luxembourg, 2015.