A Dynamic Power Management Schema for Multi-Tier Data Centers

By

Aryan Azimzadeh

April, 2016

Director of Thesis: Dr. Nasseh Tabrizi

Major Department: Computer Science

An issue of great concern as it relates to global warming is power consumption and efficient use of computers especially in large data centers. Data centers have an important role in IT infrastructures because of their huge power consumption.

This thesis explores the sleep state of data centers' servers under specific conditions such as setup time and identifies optimal number of servers. Moreover, their potential to greatly increase energy efficiency in data centers. We use a dynamic power management policy based on a mathematical model. Our new methodology is based on the optimal number of servers required in each tier while increasing servers' setup time after sleep mode to reduce the power consumption. The Reactive approach is used to prove the validity of the results and energy efficiency by calculating the average power consumption of each server under specific sleep mode and setup time. We introduce a new methodology that uses average power consumption to calculate the Normalized-Performance-Per-Watt in order to evaluate the power efficiency. Our results indicate that the proposed schema is beneficial for data centers with high setup time.

A Dynamic Power Management Schema for Multi-Tier Data Centers

A Thesis

Presented To the Faculty of the Department of Computer Science

East Carolina University

In Partial Fulfillment of the Requirements for the Degree

Master of Science in Software Engineering

By

Aryan Azimzadeh

April, 2016

A Dynamic Power Management Schema for Multi-Tier Data Centers

By

Aryan Azimzadeh

APPROVED BY:

DIRECTOR OF THESIS: ____

Nasseh Tabrizi, PhD

COMMITTEE MEMBER: ____

Mark Hills, PhD

COMMITTEE MEMBER: ____

Venkat N. Gudivada, PhD

CHAIR OF THE DEPARTMENT OF COMPUTER SCIENCE: _________________________

Venkat N. Gudivada, PhD

DEAN OF THE GRADUATE SCHOOL:____

Paul J. Gemperline, PhD

DEDICATION

I dedicate my thesis to my beloved parents “Afsaneh and Abbas” for always supporting me and showing me the right path in my life, and to Dr. Tabrizi for giving me a chance to prove myself.

ACKNOWLEDGMENTS

I would like to express my gratitude to my advisor, Dr. Nasseh Tabrizi, for the continuous support of my thesis and related research, for his patience, motivation, and immense knowledge. His guidance helped me throughout my research and writing of this thesis. I could not have imagined having a better advisor and mentor for my thesis study.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

Wh – Watt-hour

ms – milliseconds

req/s – request per second

LIST OF ABBREVIATIONS

Service Level Agreement (SLA)

Performance-Per-Watt (PPW)

Normalized-Performance-Per-Watt (NPPW)

Average (avg)

Average power consumption (P_{avg})

Sleep mode power consumption (P_{sleep})

Setup time (T_{setup})

LIST OF EQUATIONS

CHAPTER 1 – INTRODUCTION

Data centers are an essential part of Internet services and have a growing role in businesses beyond the computer industry, in fact, all networking activity relies on data centers. Considering the massive collection of servers, energy consumption should also be understood. The energy consumption in data centers is one of the biggest factors contributing to excessive expenses [1, 2]. The results of the study show that data centers consume about 2.8% of the total electricity in the USA [3]. Moreover, these centers' energy consumption represents about 3% of global energy use [4]. The main consumers of power within data centers are cooling systems and computing resources. Researchers estimate that cooling systems contribute around 30% towards data centers' energy consumption [5].

In response to concerns about growing power consumption in data centers, many businesses are attempting a new strategy called green computing. The concept of green computing is to save energy, improve efficacy, and achieve environmental protection [6]. Recent advances in energy efficiency have yielded huge improvements in both desktop and server computer technologies. At the same time, industries are faced with contributing problems that relates to computer system, including the energy consumption, exhausted emissions, building resources, high maintenance costs, global warming, and high water enterprise [7, 8]. Green computing can reduce the energy consumption of computer systems, improve their operational efficiency of emissions, and increase recycling efficiency, which could promote environmental protection and conservation of energy [9].

Today's data centers are mostly working under AlwaysOn Policy, which wastes a lot of power during periods of lower loads [10]. Researchers have proposed various solutions to reduce energy consumption by optimizing servers with a sleep mode. A servers' setup time is one of the recent challenges in dynamic power management. Although several researchers including [10] believe that it is not efficient to have a high server setup time, but this research will explain that this claim is not always true.

Current approaches to managing the server sleep state include the predictive approach, the Reactive approach, hybrid approaches, and dynamic provisioning approaches in operations research amongst others. The primary objective of this thesis is to compare the hybrid and Reactive approaches and to show that under specific circumstances the combination of these two methodologies can be used as an alternative approach to power management in green data centers.

Chapter 2 is an overview of the taxonomy of green data centers. Chapter 3 will overview a related work and explain the challenges that researchers are facing in dynamic power management with server systems. Chapter 4 will explain the methodology, and Chapter 5 will validate the methodology through the results produced. Finally, a summary of this thesis will be presented in Chapter 6.

Thesis Contribution

We introduce a new methodology to present the benefit of sleep states in data centers with high server setup times. This methodology uses combination of hybrid and Reactive approaches which are used to find the minimal number of servers. Then we show that Performance – Per – Watt (PPW) can be improved by increasing servers' setup time under specific range of sleep states. Finally, we demonstrate our results are superior to existing methods including AlwaysOn and Reactive policy.

CHAPTER 2 – A TAXONOMY AND SURVEY OF GREEN DATA CENTERS

From technical aspects, green computing can be studied in software and hardware technologies. Software technology includes design methods that enhance program efficiency, computing models such as High Performance computing, Distributed computing, and Cloud computing. Hardware aspects include technologies that reduce energy consumption, emissions footprint, and can increase economic efficiency and recycling technology.

The green data center study is classified into the following categories: computing, cooling, geographical, and network. In this chapter we consider the recent approaches in data centers form energy prospective.

2.1 GREEN DATA CENTER TAXONOMY

A. Cooling

There is extensive literature supporting the approaches to make data centers greener. One aspect of these approaches is effect of climate condition, which is reported in [11, 12]. The authors have reported that evaporative cooling and the use of waste heat from IT equipment were sufficient to support direct fresh air-cooling system. They used two methods for fresh air cooling: indirect and direct fresh air cooling [11]. In another study the authors [12] present a methodology, which consists of classification of cooling efficiency from sampled sensor and calculation of the priority metrics from statistics on cooling efficiency classes. One can see the features of temperature variation in its inlet temperature[1] (65°F to 80°F) and power consumption

[1] In 2008, American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE) expanded the recommended temperature range at the inlet of the server from 68°F to 77°F (the 2004 level) to 65°F to 80°F. However, many data centers traditionally have set their temperatures as low as 55°F. As a result, many data centers can save energy simply by raising the thermostat.

of IT equipment. To save cooling energy, historical sensor data was used to prioritize IT equipment for workload performance.

B. Computing

The further studies in green data centers include Distributed Resource Management with temperature constraint [13] and Green Resource Management under Fault Tolerance constraint [14]. These studies propose new resource management algorithms to optimally control load distribution and reduce the operational costs of data centers. There are three major challenges to this approach. These include data centers' stringent IT peak power budget, over heating problems, and distributed resource management in data centers in highly desired for system scalability [13]. Other publications have focused on minimum energy consumption constraints to prevent Service Level Agreement (SLA) violation. Their results show that the energy increase by migration has an exponential relationship with the failure rate [14]. A novel approach for managing power consumption in modern processors is dynamic voltage and frequency scaling. This allows processors to work at a suitable frequency, thus eventually reducing the energy consumption of servers [15-17]. Other approaches include the migration of virtual machines such that a minimum number of physical machines perform a specific task while the rest are kept idle [14]. The authors consider server failure when a server breaks down unusually and timing failure when processor can't finish a task during a specific time.

The two main approaches for reducing energy consumption in computer servers are dynamic voltage frequency scaling, and dynamic power management [18, 19]. While the dynamic voltage frequency and scaling focuses on optimizing the energy use of CPUs keeping the remaining server components function at their usual energy level, the dynamic power management focuses on saving energy by powering down all the server's components. The significant one is effectiveness of sleep states in data centers [10]. A different study proposes optimal power management for each server farm [20]. The method proposed reduces server power consumption by turning the servers to sleep mode. Performance metrics used in this method are delays and job blocking probability while minimizing the energy consumption. They discuss the advantages of sleep states by focusing on i. the variability in the workload trace, ii. how they use dynamic power management and iii. at the end the size of the data centers. Their results show that sleep state enhances dynamic power management; correct sleep state management can thus be very effective in large data centers. In the related article the authors find out how many servers to keep active and how much workload to delay to maximize energy saving while meeting their latency constraint. In this method they focused on how large of a workload they can execute at a given time and how much of the workload can be deferred to a later time. This research contributes an linear programing formulation for capacity provisioning by using dynamic management on deferent workloads this method determines when and how much workload each server should take. It also presents their designs, which are optimization-based online algorithms relying on the latency requirement [21]. Other authors propose finding the minimum number of servers that should be active at a specific time to meet the necessary requirements. They explore offline and online solution called "lazy capacity provisioning" that is proposed exploited from the offline solution. Their findings show that the lazy capacity materials are 3-competitive that it gives a substandard solution not larger than 3 times the optimal solution [22].

One of the effective approaches in computing aspect of green data centers is power mapping management. The authors [23] show a new technique that reduces by over an order of magnitude the amount of signaling power necessary to less than 2.5W. They show that USB device is able to generate a signal allowing non-intrusive plan(s) to identify the power connectivity of a system.

The related work to reduce CO2 emission in data centers is about the architecture of integrated gas district cooling [24]. One of the approaches in this area is to introduce the chilled water supply gap model and approaches show a combined gas district cooling and data center control model. Their results show the precision of their model strongly depends on the difference between room and outdoor temperature, and functioning steam absorption chillers. Their work suggests gas district cooling with room and outdoor temperature sensor; steam absorption chillers and heat storage tank can reduce CO2 emission.

Using optimized MapReduce energy efficiency the researchers successfully reduced energy consumption by focusing on reducing the energy impact of data movement [25]. They proposed an analysis framework for evaluating costly built in MapReduce data movement. They use a Hadoop MapReduce computer cluster to evaluate the energy efficiency of MapReduce data movement and manage the power and energy of the three major MapReduce data movements: Hadoop file system read, Hadoop file system write, and data shuffle. The reasons why they focused on data movement are: a) Data movement consumes a lot of energy in data centers because it keeps computer servers waiting for data, b) MapReduce right now is a major computing archetype in data centers for large scale processing, c) efficient storing and processing of large scale data is a practical challenge to most data centers. Many studies have been conducted in this area, which include reducing the volume of data in motion using data compression, increasing data movement speed using high speed interconnects, and applying dynamic voltage and frequency scaling to reduce CPU power consumption during data movement [25].

C. Geographical Factors

There have been some geographical studies related to green data centers. In [24] the authors propose a workload-scheduling algorithm to reduce brown energy consumption in geographically distributed data centers. They targeted different factors of green energy usage. Using their algorithm, users can dynamically schedule their workloads when the solar energy supply best satisfies their energy demand. This algorithm achieves the goal of 40% and 21 % less brown energy than other green approaches [26].

Further research on the geographical determinants of green computing proposes the idea to take wind farm location as an example to stabilize the variable and intermittent wind power. Their results are based on the real climate traces from 607 wind farms that can save 59.5% of energy. Their algorithm relies on the weight of the portfolio, which is out of renewable energy portfolio optimization. If the algorithm removes one location its weight is set to zero. and if it is selected by the algorithm its weight is assigned a percentage of the balance of the total installed capacity constructed there [27].

The review of our study is summarized in Table 1 [28].

Table 1. Taxonomy Summary of Green Data Centers

Branch	***Approach***	***Methodology***	***Comments***	***References***
Cooling	equalizing effect of 1.3 degree C in server room thermal environment	Classification , Optimization		[3]
Computing	Resource management with temperature constraints	Optimization	Good way to reduce temperature	[4]
Computing	Resource management under fault tolerance	Experimental, Optimization	Good to reduce operational cost	[5]
Computing	Effectiveness of sleep state	Experimental	Adaptive sleep modes	[16]
Computing	Server sleep scheduling	CMDP	Assign jobs to specific time slot	[17]
Computing	Server workload Delay scheduling	CMDP /Optimization		[18]
Network	Power Mapping management	Simulation, Optimization	Reduce effective cost from power consumption	[10]
Architecture	Energy efficient gap model for gas district cooling systems	Simulation	Good method to reduce heat	[20]
Computing /Network	Optimizing Map Reduce energy efficiency	Optimization	Good For heterogeneous environment	[21]
Geographical/Cloud	Energy efficient workload scheduling	Optimization		[22]
Geographical	Stabilizing the variable in wind farms	Analytical modeling	Suitable for big data centers	[23]

CHAPTER 3 – RELATED WORKS

In this section, the related work in data center dynamic power management is discussed. The prior work in different aspects of dynamic power management will be explained and highlighted. In order to demonstrate the resulting method as superior, one must explore related systems and analyze the tradeoffs in various approaches.

When using power management, in order to improve the energy efficiency of data centers, three techniques are commonly employed: selected servers shutdown, frequency and voltage provisioning, and dynamic power management [29]. There are three different kinds of dynamic power management: predictive and Reactive [30]. The predictive approaches will envision the future request rate using previous data in order to recognize when the servers must be turned on [30]. On the other hand, the Reactive approaches will react to the request immediately by turning the servers on or off. There is also another branch, which is the hybrid approach. The hybrid approach includes both predictive and Reactive methods

3.1 Predictive Approaches

One of the approaches in this area is to use different types of predictive policies, such as exponentially weighted average, moving window average, and linear regression in order to predict the future request rate and add or remove servers based on the results. The authors determined that using moving window policies and linear regression enabled the best results for the workload traces that they considered. Thus, this methodology provides a means to more efficient power consumption than static approaches have in the past [31].

In another approach, researchers used auto regression policy to predict the request rate for specific arrival patterns and used the result of this calculation to determine the threshold policies that were able to trigger the servers on and off. Their dynamic power management policy is energy efficient for periodic request rates repeating on a daily basis [32].

3.2 Reactive Approaches

In [33] the authors used a theoretical method as a control in order to manage resources to applications in a multi-tier data center. They used specific queuing theory to predict response time and allocated resources based on the estimated response time and power consumption.

Approach used by the author involved using a Reactive feedback mechanism to monitor a multi-tier web application. The author evaluated CPU utilization and response time and changed the number of servers based on these calculations, making the point that using multiple sleep states in servers could have significant improvement in energy savings [34].

There are some other approaches which mostly study modeling and dynamic provisioning on the performance side of multi-tier, and the approaches barely focus on power consumption [35, 36].

In [10] the authors proposed two different approaches called: Reactive and SoftReactive. Their results are achieved under different traces. Sleep states appreciate dynamic power management. They describe certain types of traces, evaluate them, and figure out which is the best match for sleep states. Reactive approach responds to changes in requests and loads by turning the servers to sleep mode and waking them back up when the load increases. There has been big concern for the Reactive policy; In Reactive policy the servers go off so quickly when not needed, but when the loads rise, it takes time for servers to come back on again. So, to cure this problem, they introduce another policy called SoftReactive. In SoftReactive approach, the server goes to idle mode for a short time before it turns off. This delay in transition gives the opportunity for the server to wait for possible arrival load. If the server gets requested during the delay time, then the server goes back to the regular mode. The researchers set timers for each server to turn off, and the idea prevents the mistake of turning the server on at the wrong time. The problem raises in this methodology when the researchers put too many servers in the idle mode. To solve this issue, they introduced a routing plan, which distributes jobs onto the low amount servers, so the unneeded servers will go into sleep mode.

3.3 Hybrid Approaches

Hybrid approach includes both predictive and Reactive approaches. Predictive methods are used in long-term workload trends, and Reactive methods are used in short-term unpredictable trends [30].

In a different study [37], the authors first used the Reactive method for unpredictable trends in request rate and later used the predictive method for long-term trends in request rate. Separately, the authors proposed a solution called PowerNap that has a way to switch its state from high performance to low power (sleep mode) and vice versa to respond to the rapid server loads. Using this methodology, the authors were able to put the servers in sleep mode long before the servers go into idle mode, so they are actually replacing the low server utilization periods with an energy efficient sleep mode [38].

Further study reported in [29], where the authors introduce new methodology consisting of multiple approaches. They use dynamic provisioning, frequency scaling, and dynamic power management methods to make multi-tier data centers more energy efficient. They propose two algorithms; one focuses on the optimal number of servers by dynamically provisioning them, and the other algorithm, mostly focuses on the CPU speed and the duration of sleep states for each server.

Unfortunately, thus far, based on our extensive literature review, hybrid approaches have had problems predicting workloads and Reactive approaches, but this thesis reports on our attempt to overcome this problem by combining some aspect of hybrid and Reactive approaches together.

CHAPTER 4 – METHODOLOGY

In this chapter, the methodology based on two previous approaches using dynamic power management will be described. The goal of this research is to point out the fact that, under specific conditions, two different methodologies can be combined as an improved green approach, in the field of dynamic power management in data centers.

The methodology involves one front-end load generator and one front-end load balancer, which distributes request from the load balancer to expected application servers. The load balancer is also being responsible for turning the application server to sleep mode and waking them up. There are also several Memcached, servers to fetch data required to service the requests [10]. Memcached "is an in-memory key-value store for small chunks of arbitrary data (strings, objects) from results of database calls, API calls, or page rendering." Furthermore, power management techniques are applied on the front-end application server side.

In this methodology, we calculate the optimal energy consumption using (Eq. 1) [29], then we differentiate E with respect to CPU speed, and then we converted to the power consumption using (Eq. 2). The goal of this conversion is we used T_{setup} based on the 95 percentile of customers response time. Our methodology uses TPC-W [39] based workload in multi-tier data center.

$$E = P(s, 1.0)[(T - t)(\rho(1 - k) + k) + tk'] \qquad (1)$$

Where P is the power consumption, ρ represents the utilization of a system, s represents the CPU speed, t represents sleep state duration, k represents the ratio of the idle power consumption to the peak power consumption, k$'$represents the ratio sleep power consumption to peak power consumption and T represents time interval length.

$$P = \frac{E}{T_{setup}} \qquad (2)$$

Where P is a power consumption, E represents energy consumption and T_{setup} is the setup time .

$E = 250\ (Wh)$ is our optimal energy consumption as we are using the same conditions for our evaluation that is reported in [29], where $1.2 < s$ <3.0, 0 < k < 1, and 0 < k$'$< 1.

The parameters of interest include the average power consumption, setup time, response time and the number of active servers. Setup time is defined as the time that servers take to turn back on from sleep mode. Although, long setup times are not recommended commonly, we will show that if specific time slots are considered in our calculation in combination with the specific number of servers, it can be efficient to use long setup times. Improving energy efficiency by increasing the servers' setup time is our main focus in this research.

Our consideration for hybrid aspects include the CPU speed and also how to get to the minimal number of servers. So we get the expected minimal number of servers using (Eq. 3) [29]. The minimal number of servers from (Eq. 3) should meet the SLA requirement, and can help to achieve good ratings in the power saving approach. The CPU utilization can be obtained by monitoring the supported tools by operation systems. Then we analyze the number of requests by a server in different time frames.

$$v_i = \frac{L_i + r_i}{T_{SLA} + \tau_i} \qquad (3)$$

Where, v_i represent the minimal number of servers in each tier, L_i is number of queued for each tier, r_i is number in incoming request for each tier, T_{SLA} is the target response time and τ_i is estimated throughout tier i.

$$Front - end\ servers = \left\lceil \frac{r}{60} \right\rceil \qquad (4)$$

Where r represents request rates.

In our approach, the number of servers from (Eq. 3) will be validated with the peak number of requests in (Eq. 4) [10]. Each front end server can handle 60 req/s as the we are considering the same condition reported in[10]. This result is based on a T_{95} threshold of 500 form the results mentioned in Figure 2. In this research we compare the results with the AlwaysOn policy and Reactive policy. Note that in the real world, the request rate cannot be calculated in advance, but we assume the request rate in advance from the AlwaysOn policy.

A peak request of 800 req/s is assumed in [10], so we use the same peak request for specific benchmark dynamically over 30 minutes, and our dynamic power management scheme calculates the number of servers for each tier during the next time interval. Based on (Eq. 4) , $\left\lceil \frac{800}{60} \right\rceil$ =14 servers for the AlwaysOn policy are needed at all times, but this number can vary in our methodology.

With Reactive policy, the servers react to the ongoing request rate and can adjust their capacity in real time. However, in our approach it has been said that Reactive policy suffers from long setup times. We show that it can prove power efficient to use it in our way. The methodology demonstrates, when we increase the servers' setup time and also use the minimal number of servers and which is calculated in (Eq. 3) and (Eq. 4), remarkable results will be obtained in field of power efficiency.

Our approach sets the servers to sleep mode if

- The actual number of servers are more than $\left[\frac{r}{60}\right]$ assuming the servers are called back from sleep
- There is delay in the incoming requests.

In order to determine for how long, the servers are put in sleep mode and the response time for each request is estimated. (Eq. 5) [29] is used to get to approximate response time in 30 minutes setup time.

$$Response\ time = \frac{(\ L_i+1)n}{s} \tag{5}$$

Where, L_i is the number of requests, n is the number diciplines that CPU need to process the request and s is the CPU speed.

The sorted response time and T_{95} of response time for the TCP-W benchmark for each time slots are shown in Figures 1 and 2. As shown in Figure 2, the T_{95} starts from T= 12 minutes, so in our calculation we exclude time frames before T= 12 minutes to meet The SLA limit of 2000ms. The hybrid approach keeps the response time below 2000ms, thus making it easier to allocate the expected number of servers [29].

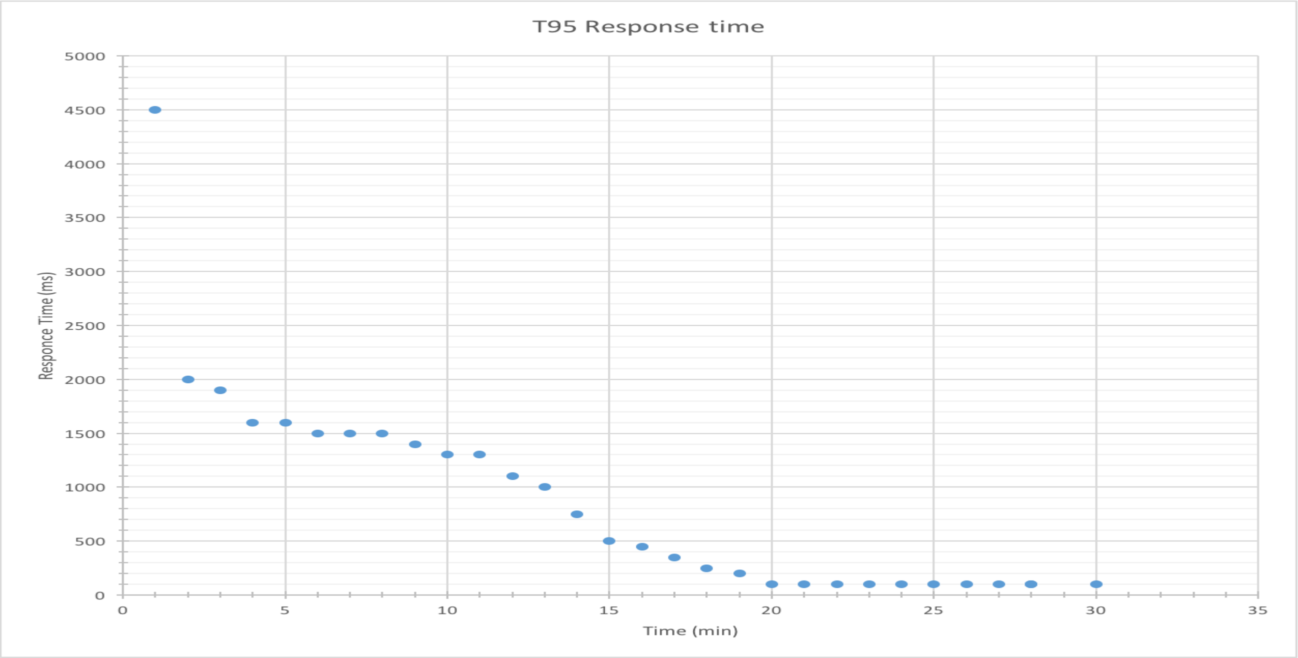

Figure 1: Sorted response time

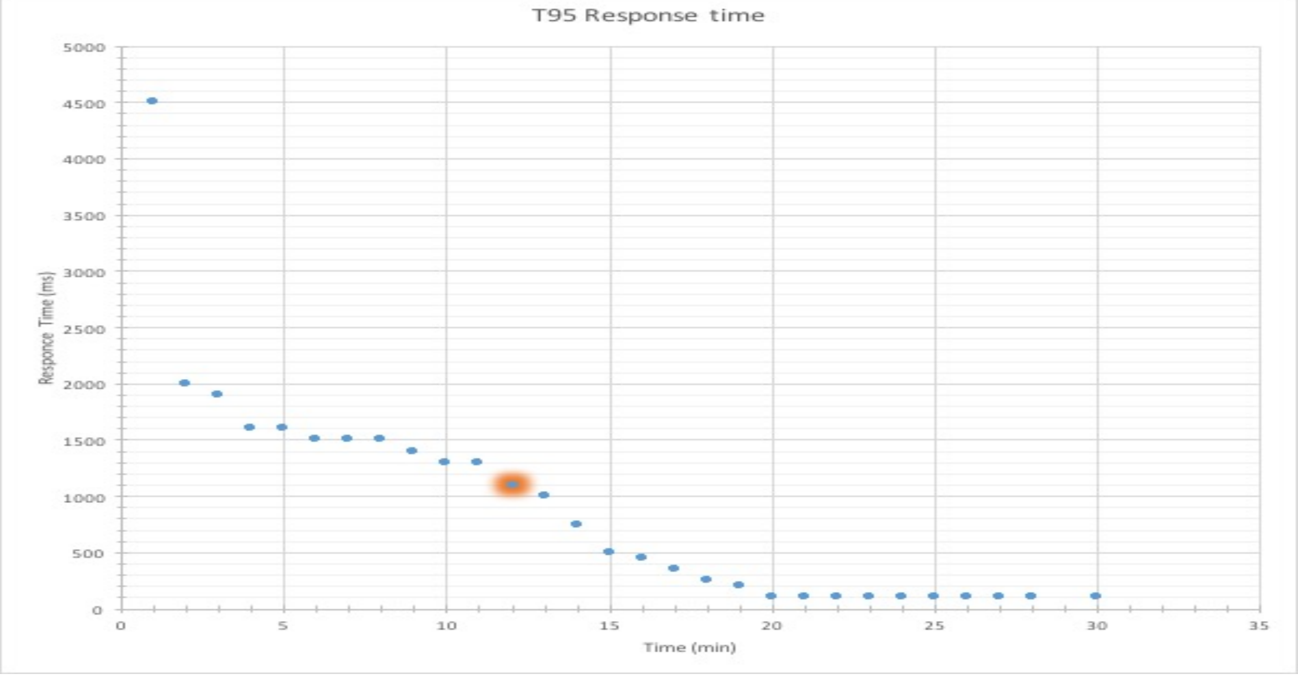

Figure 2: T_{95} response time

We also need to get the the average power consumption, P_{avg}, for our calculation. The approach was to replicate the influence of using sleep state, by not sending the request to the server when it is marked for sleep and changing its power consumption by P_{sleep}. To prove our approach is energy efficient we use NPPW, but before NPPW is calculated, PPW, is required. We need PPW for both Reactive and AlwaysOn policy [10].

4.1 AlwaysOn Policy

AlwaysOn is a static power management policy, which most of the industries nowadays are using. The policy has a constant number of active front-end servers at all times. To figure out how many servers this policy uses, the amount of request rates that each front-end server can handle must be observed [10]. This is the critical point, when the 95th percentile of certain threshold will be implemented.

This policy is designed to meet peak request rate, but it does not have the ability to envision when peak request rate occurs. The average power consumption for the AlwaysOn policy is always high. Moreover, the 95th percentile of response time and average power consumption under AlwaysOn are unchanged in a favor of sleep states. That is why the AlwaysON policy was chosen to compare the approach established by this research and the Reactive approach.

4.2 Average Power Consumption Calculation

In this section we explain how to calculate the average power consumption, P_{avg}, in our Reactive approach. To get the $P_{avg}$for specific setup time, the power consumption in that time must be calculated first. P_{avg} is different for various setup times when the P_{sleep} is zero. The

P_{avg} is calculated based on setup times which starts from time slot 12 minute. Paragraph three also mentioned that $T_{setup} = 12\ minutes$ is the the start point of our T_{95} of response time. First we calculate the $P_{avg}$when the server $P_{sleep}$is zero and then increase the setup time in this state to get power consumption. Although $T_{setup} = 12\ minutes$ is a start point of our T_{95} of response time, we start the calculation from $T_{setup} = 15\ minutes$ because based on our calculation the power consumption before $T_{setup} = 15\ minutes$ is not efficient. After the first calculations of P_{avg} for $P_{sleep} = 0$ then P_{sleep} increases [10]. We predict P_{avg} for a given T_{setup} and P_{sleep} by analyzing the results as in [10]. Note that, all the results for P_{avg} is in Reactive mode. Figure 3 shows the results for P_{avg}.

P_{sleep}(watts)

T_{setup}(min)	**0**	**28**	**56**	**84**
15	1000	1279	1558	1837
16	937	1216	1495	1774
17	833	1161	1440	1719
18	883	1112	1391	1678
19	789	1068	1347	1620

Figure 3: Results for our approach with respect to P_{avg}

As seen in Figure 3, P_{avg} decreases when we increase the T_{setup} at the same P_{sleep}. On the other hand, when P_{sleep} increases, P_{avg} increases at the constant level of the T_{setup}. For instance, for $P_{sleep} = 0$, when the T_{setup} increases from 15 minutes to 19 minutes, P_{avg} decrease from 1000 Watts to 789. On the other hand, for $T_{setup} = 19$ minutes, P_{avg} boost from 789 Watts to 1620 watts. We will explain later in this chapter that low P_{avg} is so beneficial for our system.

4.3 Performance Per Watt (PPW) *Calculation*

PPW is extremely important to our calculations. Note that higher PPW is better to get to the improved energy efficiency. (Eq. 6) shows that for each specific T_{setup}, we get the same value of T_{95} by increasing P_{sleep} . So the calculation has only five different T_{95} values as we are considering five different T_{setup}. T_{95} increases as T_{setup} increases. Figure 4 shows the PPW calculation. The results for PPW show that by increasing P_{sleep} at specific T_{setup}, PPW decrease and by contrast, when P_{sleep} is constant, PPW increases by increasing T_{setup}. That is why when we have the maximum value of PPW when T_{setup}= 19 minutes and P_{sleep}= 0. Note that, PPW for AlwaysOn is unaffected by changes in P_{sleep} and T_{setup} and it has a constant value of $1.7 \cdot 10^{-6}\ (ms.watts)^{-1}$ which reported in [10]. We will use these values of PPW in the next chapter to compute NPPW.

$$PPW = \frac{1}{P_{avg} \cdot T_{95}} \qquad (6)$$

P_{sleep}(watts)

T_{setup}(min)	**0**	**28**	**56**	**84**
15	$2*10^{-6}$	$1.5*10^{-6}$	$1*10^{-6}$	$1*10^{-6}$
16	$2*10^{-6}$	$1.8*10^{-6}$	$1.5*10^{-6}$	$1.3*10^{-6}$
17	$3*10^{-6}$	$2*10^{-6}$	$2*10^{-6}$	$1.6*10^{-6}$
18	$5*10^{-6}$	$4*10^{-6}$	$3*10^{-6}$	$2*10^{-6}$
19	$6*10^{-6}$	$4*10^{-6}$	$4*10^{-6}$	$3*10^{-6}$

Figure 4: Results for our approach with respect to PPW

CHAPTER 5 – RESULTS

We computed PPW for various sleep states duration from chapter 4. Now these values are used to prove that not only are our results superior to the AlwaysOn policy, but they are also superior to the Reactive approach [10]. To prove this we will need to get NPPW for all T_{setup} and P_{sleep}. (Eq. 7) shows how to calculate NPPW by normalizing PPW for Reactive by PPW for AlwaysOn.

$$NPPW = \frac{PPW}{PPW^{AlwaysOn}} \tag{7}$$

When NPPW exceeds 1, it demonstrates that our approach is superior to AlwaysOn. This means that our result is more energy efficient. By using the optimal number of servers (approximately 60 servers) that is based on our calculation (Eq. 3) and (Eq. 4) and also comparable results reported in [10], the results that are shown in Figure 4 are observed. The results from Figure 4 are then used as PPW in (Eq. 7) to calculate NPPW. As mentioned in chapter 4, for AlwaysOn Policy $PPW = 1.7 \cdot 10^{-6}\ (ms.watts)^{-1}$.

Figure 5 shows our result for NPPW for slowly varying traces. White regions demonstrate higher NPPW, where NPPW > 1 argue that our approach is superior to AlwaysOn Policy.

P_{sleep}(watts)

T_{setup}(min)	**0**	**28**	**56**	**84**
15	1.17	0.89	0.59	0.59
16	1.17	1.06	0.89	0.76
17	1.76	1.17	1.17	0.94
18	2.9	2.35	1.76	1.17
19	3.53	2.35	2.35	1.76

Figure 5: Normalized-Performance-Per-Watt (NPPW) under our approach

Figure 5 shows that NPPW increases as T_{setup} increases and P_{sleep} decreases. As an illustration we have a maximum NPPW of 3.53 when the $T_{setup} = 19\ minutes$ and $P_{sleep} = 0$.

Our findings show that by using our new methodology, adopting sleep states under Reactive and hybrid policy can provide demonstrable benefit in terms of NPPW when we increase the servers setup time. Using the specifications form calculation and the result of NPPW, we were able to achieve significant improvements in energy efficiency compared to AlwaysOn policy and previous Reactive approach

Figure 6 shows that the results of our method compared to those previously acquired by other groups while scaling the number of servers up from 14 to 60, magnification increases

NPPW. While not usually recommended, the results make our approach more desirable as compared to AlwaysOn and Reactive policies.

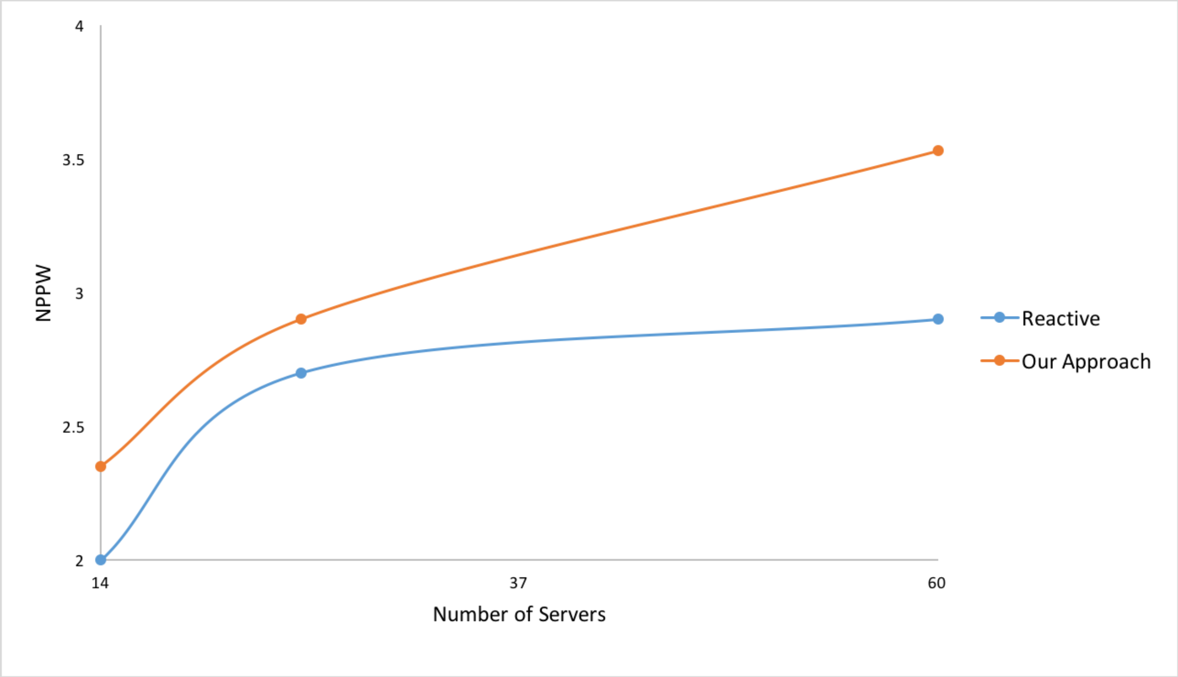

Figure 6: Effect of scaling on NPPW and comparison of our results to Reactive

CHAPTER 6 – CONCLUSION

In this research we classified the different aspects of green data centers and summed up them in Table 1. The new methodology was then introduced to examine the benefit of sleep states with high server setup times. The methodology uses the combination of Reactive and hybrid approaches, which is used to find the minimal and optimal number of servers. The methodology needed 95^{th} percentile of response time, that was calculated for the specific setup time. We used specific ranges of sleep states with different high setup times and proved that it can boost PPW. The PPW results validate our results to be superior to AlwaysOn. Then We calculated NPPW and proved that our approach is also superior to previous Reactive approach under specific circumstances. Finally, we compared our result by increasing the number of servers with Reactive approach; our examination shows the effectiveness of sleep states when the number of servers increases. In particular, the results express that the proposed schema introduced in this thesis can reduce the power consumption by 48% relative to static provisioning and AlwaysOn policy.

REFERENCES

[1] Barroso, Luiz André, and Urs Hölzle. "The case for energy-proportional computing." Computer 12 (2007): 33-37.

[2] Raghavendra, Ramya, Parthasarathy Ranganathan, Vanish Talwar, Zhikui Wang, and Xiaoyun Zhu. "No power struggles: Coordinated multi-level power management for the data center." In ACM SIGARCH Computer Architecture News, vol. 36, no. 1, pp. 48-59. ACM, 2008.

[3] www.cisco.com/en/US/prod/collateral/switches/ps5718/ps10195/ CiscoEMSWhitePaper.pdf

[4] A. Rallo, "Industry Outlook: Data Center Energy Efficiency." [Online]. Available: http://www.datacenterjournal.com/it/industry-outlook-datacenter-energy-efficiency/. [Accessed: 02-Sep-2014].

[5] Cavdar, Derya, and Fatih Alagoz. "A survey of research on greening data centers." In Global Communications Conference (GLOBECOM), 2012 IEEE, pp. 3237-3242. IEEE, 2012.

[6] Zhang, Xiaodan, Lin Gong, and Jun Li. "Research on green computing evaluation system and method." In Industrial Electronics and Applications (ICIEA), 2012 7th IEEE Conference on, pp. 1177-1182. IEEE, 2012.

[7] Li, Qilin, and Mingtian Zhou. "The survey and future evolution of green computing." In Proceedings of the 2011 IEEE/ACM International Conference on Green Computing and Communications, pp. 230-233. IEEE Computer Society, 2011.

[8] http://www.zdnet.com.cn/server/2007/1210/676572.shtml.

[9] Guo Bing, Shen Yan, and Shao ZL, “The Redefinition and Some Discussion of Green Computing”, Chinese Journal of Computers, vol. 32, Dec. 2009, pp. 2311-2319.

[10] Gandhi, Anshul, Mor Harchol-Balter, and Michael A. Kozuch. "Are sleep states effective in data centers?." In Green Computing Conference (IGCC), 2012 International, pp. 1-10. IEEE, 2012.

[11] Endo, Hiroshi, Hiroyoshi Kodama, Hiroyuki Fukuda, Toshio Sugimoto, Takashi Horie, and Masao Kondo. "Effect of climatic conditions on energy consumption in direct fresh-air container data centers." Sustainable Computing: Informatics and Systems (2014).

[12] Mase, Masayoshi, Jun Okitsu, Eiichi Suzuki, Tohru Nojiri, Kentaro Sano, and Hayato Shimizu. "Cooling efficiency aware workload placement using historical sensor data on IT-facility collaborative control." In Green Computing Conference (IGCC), 2012 International, pp. 1-6. IEEE, 2012.

[13] slam, Mohammad A., Shaolei Ren, Niki Pissinou, Hasan Mahmud, and Athanasios V. Vasilakos. "Distributed resource management in data center with temperature constraint." In Green Computing Conference (IGCC), 2013 International, pp. 1-10. IEEE, 2013.

[14] Ghoreyshi, Seyed Mohammad. "Energy-efficient resource management of cloud datacenters under fault tolerance constraints." In Green Computing Conference (IGCC), 2013 International, pp. 1-6. IEEE, 2013.

[15] Garg, Saurabh Kumar, Chee Shin Yeo, Arun Anandasivam, and Rajkumar Buyya. "Environment-conscious scheduling of HPC applications on distributed cloud-oriented data centers." Journal of Parallel and Distributed Computing 71, no. 6 (2011): 732-749.

[16] Von Laszewski, Gregor, Lizhe Wang, Andrew J. Younge, and Xi He. "Power-aware scheduling of virtual machines in dvfs-enabled clusters." In Cluster Computing and Workshops, 2009. CLUSTER'09. IEEE International Conference on, pp. 1-10. IEEE, 2009.

[17] Kim, Kyong Hoon, Rajkumar Buyya, and Jong Kim. "Power Aware Scheduling of Bag-of-Tasks Applications with Deadline Constraints on DVS-enabled Clusters." In CCGRID, vol. 7, pp. 541-548. 2007.

[18] Pouwelse, Johan, Koen Langendoen, and Henk Sips. "Energy priority scheduling for variable voltage processors." In Low Power Electronics and Design, International Symposium on, 2001., pp. 28-33. IEEE, 2001.

[19] Benini, Luca, Alessandro Bogliolo, and Giovanni De Micheli. "A survey of design techniques for system-level dynamic power management." Very Large Scale Integration (VLSI) Systems, IEEE Transactions on 8, no. 3 (2000): 299-316.

[20] Niyato, Dusit, Sivadon Chaisiri, and Lee Bu Sung. "Optimal power management for server farm to support green computing." In Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, pp. 84-91. IEEE Computer Society, 2009.

[21] Adnan, Muhammad Abdullah, Ryo Sugihara, Yan Ma, and Rajesh K. Gupta. "Energy-optimized dynamic deferral of workload for capacity provisioning in data centers." In Green Computing Conference (IGCC), 2013 International, pp. 1-10. IEEE, 2013.

[22] Minghong, Adam Wierman, Lachlan LH Andrew, and Eno Thereska. "Dynamic right-sizing for power-proportional data centers." IEEE/ACM Transactions on Networking (TON) 21, no. 5 (2013): 1378-1391.

[23] Ferreira, Alexandre, Wael El-Essawy, Juan C. Rubio, Karthick Rajamani, Malcolm Allen-Ware, and T. Keller. "BCID: An effective data center power mapping technology." In Green Computing Conference (IGCC), 2012 International, pp. 1-10. IEEE, 2012.

[24] Okitsu, Jun, Mohd Fatimie Irzaq Khamis, Nordin Zakaria, Ken Naono, and Ahmad Abba Haruna. "Toward an architecture for integrated gas district cooling with data center control to reduce CO 2 emission." Sustainable Computing: Informatics and Systems (2014).

[25] Wirtz, Thomas, Rong Ge, Ziliang Zong, and Zizhong Chen. "Power and energy characteristics of MapReduce data movements." In Green Computing Conference (IGCC), 2013 International, pp. 1-7. IEEE, 2013.

[26] Chen, Changbing, Bingsheng He, and Xueyan Tang. "Green-aware workload scheduling in geographically distributed data centers." In Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on, pp. 82-89. IEEE, 2012.

[27] Dong, Chuansheng, Fanxin Kong, Xue Liu, and Haibo Zeng. "Green power analysis for geographical load balancing based datacenters." In Green Computing Conference (IGCC), 2013 International, pp. 1-8. IEEE, 2013.

[28] Azimzadeh, Aryan, Nasseh Tabrizi. " A Taxonomy and Survey of Green Data Centers." In International Conference on Computational Science and Computational Intelligence (CSCI), 2015 International, pp. 128-131. IEEE, 2015.

[29] Lim, Seung-Hwan, Bikash Sharma, Byung Chul Tak, and Chita R. Das. "A dynamic energy management scheme for multi-tier data centers." In Performance Analysis of Systems and Software (ISPASS), 2011 IEEE International Symposium on, pp. 257-266. IEEE, 2011.

[30] Gandhi, Anshul. "Dynamic Server Provisioning for Data Center Power Management." PhD diss., Intel, 2013.

[31] Krioukov, Andrew, Prashanth Mohan, Sara Alspaugh, Laura Keys, David Culler, and Randy Katz. "Napsac: Design and implementation of a power-proportional web cluster." ACM SIGCOMM computer communication review 41, no. 1 (2011): 102-108.

[32] Chen, Gong, Wenbo He, Jie Liu, Suman Nath, Leonidas Rigas, Lin Xiao, and Feng Zhao. "Energy-Aware Server Provisioning and Load Dispatching for Connection-Intensive Internet Services." In NSDI, vol. 8, pp. 337-350. 2008.

[33] Abdelzaher, Tarek, Yixin Diao, Joseph L. Hellerstein, Chenyang Lu, and Xiaoyun Zhu. "Introduction to control theory and its application to computing systems." Performance Modeling and Engineering (2008): 185-215.

[34] Horvath, Tibor, and Kevin Skadron. "Multi-mode energy management for multi-tier server clusters." In Proceedings of the 17th international conference on Parallel architectures and compilation techniques, pp. 270-279. ACM, 2008.

[35] Urgaonkar, Bhuvan, Prashant Shenoy, Abhishek Chandra, and Pawan Goyal. "Dynamic provisioning of multi-tier internet applications." In Autonomic Computing, 2005. ICAC 2005. Proceedings. Second International Conference on, pp. 217-228. IEEE, 2005.

[36] Urgaonkar, Bhuvan, Giovanni Pacifici, Prashant Shenoy, Mike Spreitzer, and Asser Tantawi. "An analytical model for multi-tier internet services and its applications." In ACM SIGMETRICS Performance Evaluation Review, vol. 33, no. 1, pp. 291-302. ACM, 2005.

[37] Gandhi, Anshul, Yuan Chen, Daniel Gmach, Martin Arlitt, and Manish Marwah. "Minimizing data center sla violations and power consumption via hybrid resource provisioning." In Green Computing Conference and Workshops (IGCC), 2011 International, pp. 1-8. IEEE, 2011.

[38] Meisner, David, Brian T. Gold, and Thomas F. Wenisch. "PowerNap: eliminating server idle power." In ACM Sigplan Notices, vol. 44, no. 3, pp. 205-216. ACM, 2009.

[39] TPC-W: Benchmarking An E-Commerce Solution. http://wwws.tpc.org/

information/other/techarticles.asp.