USING ANT COLONY OPTIMIZATION ON THE QUADRATIC ASSIGNMENT PROBLEM TO ACHIEVE LOW ENERGY COST IN GEO-DISTRIBUTED DATA CENTERS

By

RICHARD OSEI

Bachelor of Arts/Science in Computer Science & Mathematics

Langston University

Langston, Oklahoma

2008

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
December, 2015

USING ANT COLONY OPTIMIZATION ON THE QUADRATIC ASSIGNMENT PROBLEM TO ACHIEVE LOW ENERGY COST IN GEO-DISTRIBUTED DATA CENTERS.

Thesis Approved:

Dr. Christopher Crick

Dr. Nohpill Park

Dr. Ronak Etemadpour

ACKNOWLEDGEMENTS

He that started His good work in me has brought it to completion. My first thanks go to God Almighty for giving me the strength, direction and wisdom to complete my thesis. My sincerest gratitude goes to my committee members, Dr. Nohpill Park, Dr. Ronak Etemadpour and especially to my adviser Dr. Christopher Crick who has been such a blessing. My thanks go to him for his support, invaluable suggestions and guidance as a whole. I wouldn’t have come this far in my thesis without his help and creative ideas.

I would like to take this opportunity to also thank my wife and kid, for giving the time I needed to complete this project.

Acknowledgements reflect the views of the author and are not endorsed by committee members or Oklahoma State University.

Name: Richard Osei

Date of Degree: DECEMBER, 2015

Title of Study: USING ANT COLONY OPTIMIZATION ON THE QUADRATIC ASSIGNMENT PROBLEM TO ACHIEVE LOW ENERGY COST IN GEO-DISTRIBUTED DATA CENTERS

Major Field: Computer Science

There are many problems associated with operating a data center. Some of these problems include data security, system performance, increasing infrastructure complexity, increasing storage utilization, keeping up with data growth, and increasing energy costs. Energy cost differs by location, and at most locations fluctuates over time. The rising cost of energy makes it harder for data centers to function properly and provide a good quality of service. With reduced energy cost, data centers will have longer lasting servers/equipment, higher availability of resources, better quality of service, a greener environment, and reduced service and software costs for consumers. Some of the ways that data centers have tried to using to reduce energy costs include dynamically switching on and off servers based on the number of users and some predefined conditions, the use of environmental monitoring sensors, and the use of dynamic voltage and frequency scaling (DVFS), which enables processors to run at different combinations of frequencies with voltages to reduce energy cost. This thesis presents another method by which energy cost at data centers could be reduced. This method involves the use of Ant Colony Optimization (ACO) on a Quadratic Assignment Problem (QAP) in assigning user request to servers in geo-distributed data centers.

In this paper, an effort to reduce data center energy cost involves the use of front portals, which handle users' requests, were used as ants to find cost effective ways to assign users requests to a server in heterogeneous geo-distributed data centers. The simulation results indicate that the ACO for Optimal Server Activation and Task Placement algorithm reduces energy cost on a small and large number of users' requests in a geo-distributed data center and its performance increases as the input data grows. In a simulation with 3 geo-distributed data centers, and user's resource request ranging from 25,000 to 25,000,000, the ACO algorithm was able to reduce energy cost on an average of $.70 per second. The ACO for Optimal Server Activation and Task Placement algorithm has proven to work as an alternative or improvement in reducing energy cost in geo-distributed data centers.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION

In this thesis, another way of reducing energy cost in geo-distributed data centers by the use of Ant Colony Optimization (ACO) on a Quadratic Assignment Problem (QAP) (such as matching a user request to a server in a data center) is explored. The rising cost of energy has made reducing energy cost at data centers a very important goal. For example, electricity alone cost Google over $38 million a year [1]. Even though one of the factors for selecting a location for a data center is energy cost, others include labor costs and availability, highway accessibility and quality, proximity to major markets and customers, availability and cost of real estate options, amount of local and state economic development incentives, availability of telecommunications infrastructure, the cost of utilities, and tax and regulatory climate [2]. Figure 1 shows Google data centers around the world.

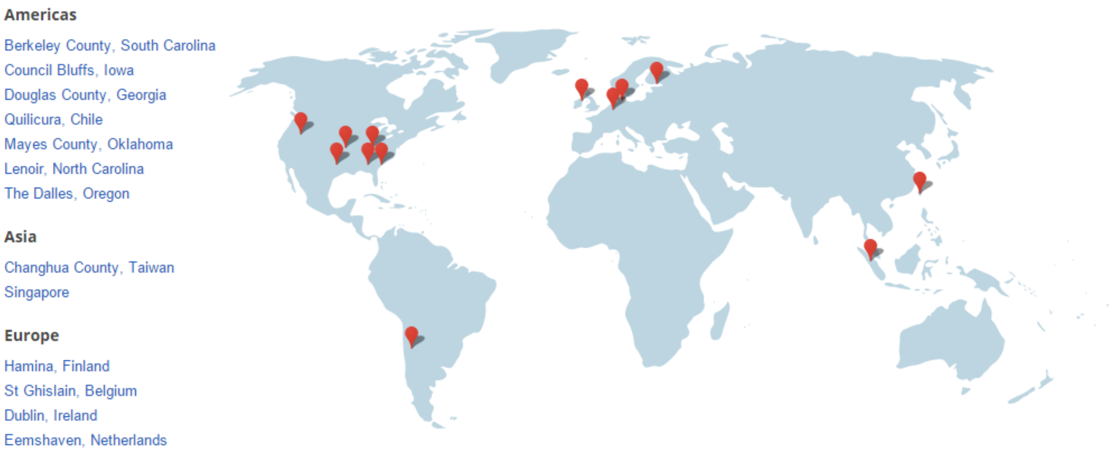


Figure 1. Google data center locations [3].

Data centers and cloud computing have become a major part of software and application delivery. Data centers hosting these cloud-based applications make software applications readily available to consumers anywhere and at any time. Data centers also make it possible to reduce the cost of distributing software, by eliminating to the need to write the applications to media, such as a disk or USB storage. Today, data centers have become a big part of accessing software, with most software providers switching to a cloud-based application, for example, Microsoft Office, Adobe Creative Suite, and many others. "All of our online activity is delivered through data centers, and the more we send email, watch online videos, use social media like Facebook, and conduct business online, the more demands on data centers will grow" [4]. As more and more data centers are being built, due to the switch from desktop applications to cloud based applications, the need to reduce energy has become an important factor.

1.1 Data Center/Cloud Computing

In recent times, cloud computing (cloud, in the cloud, cloud based) has become a buzzword not just in information Technology (IT) but in other sectors like banking, finance, retail, health, utilities, education, airlines and many others. However, an encounter with the concept of "cloud computing" can be a bit confusing because there are so many different definitions. The cloud, which refers to a data centers operations, signifies that the entire IT industry is transforming from a physical world towards a virtual world. Having the right definition of cloud computing lays a solid foundation for understanding the purpose of data centers [2]. Data centers can then be defined as a networking of systems that provide services (email, storage, website, applications, etc.) for consumers in diverse places over the internet.

The term ‘cloud computing,’ which is still evolving, is less than two decades old. The first academic use of the term cloud computing was casted by Ramnath K. Chellapa in 1997 in his paper, “*Intermediaries In Cloud Computing: A New Computer Paradigm*” [2]. Ramnath explains that the computing standard will not be controlled within a technical boundary but by commercialization of software. Software companies finding ways to reduce cost and at the same time serve their customers better, has increased the growth of data centers. Data centers allow software providers to reduce cost in terms of distribution, and customers gaining access to software are no longer limited to a machine at a specific location, but can access their data and applications from anywhere on any machine. Today, in terms of service type, cloud computing can be categorized as Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). Data centers are growing because SaaS, IaaS and PaaS have become the preferred way of computing and networking [5].

1.2 Data Center Services

All of our online activity is delivered through data centers, and the more we send email, watch online videos from sites like Youtube and Netflix, use social media like Facebook, and conduct business online (online shops), data centers will continue to grow [4]. Services at a data center can be categorized in to three groups: (1), Infrastructure as a Service (IaaS), which provides computing resources in a virtualized environment (the cloud), using a public connection over the internet. What is provided specifically is the hardware (servers) for computing a task. (2), Platform as a Service (PaaS) is a service provided for developers to build application and services over the internet. An example of PaaS are a website builders like Wix.com, online mobile application development sites, etc. (3), Software as a Service (SaaS) refers to any online services where customers are able to access software over the internet. Office 360, Twitter and Facebook are all example of SaaS [5].

The increase in growth of data centers is due to flexibility of access on enabled devices, networking, cost efficiency, and safety that “the cloud” provides. There are also issues that are rising with the growth of data centers, not with the service they provide, but in maintaining the infrastructure.

1.3 Data Problems

Availability of resources and high power consumption is a well known issue with data centers. Google has more than 500K servers in their data centers; it costs them more than $38M worth of electricity each year. It has been estimated that the average CPU utilization for Google is around 40%, even though its services are provided around the world with millions of users every second accessing its data center resources. Google’s efficiency is based on the fact that Google has invested a significant amount of effort on making their data centers “greener”: more efficient [1]. Just like Google, many big data centers do a great job of efficiency, but this only represent 5% of data centers’ energy use. Small, medium, corporate and multi-tenant operations are much less efficient and represent the 95% of inefficiency of energy use, because these data centers do not have any methods in place for energy use reduction. It has also been estimated that data centers waste a large amount of energy powering equipment, with the average server operating at only 12-18% of capacity [4]. Figure 2 shows the amount of electricity that is used by data centers, ranging from small size to high-performance computing data centers.

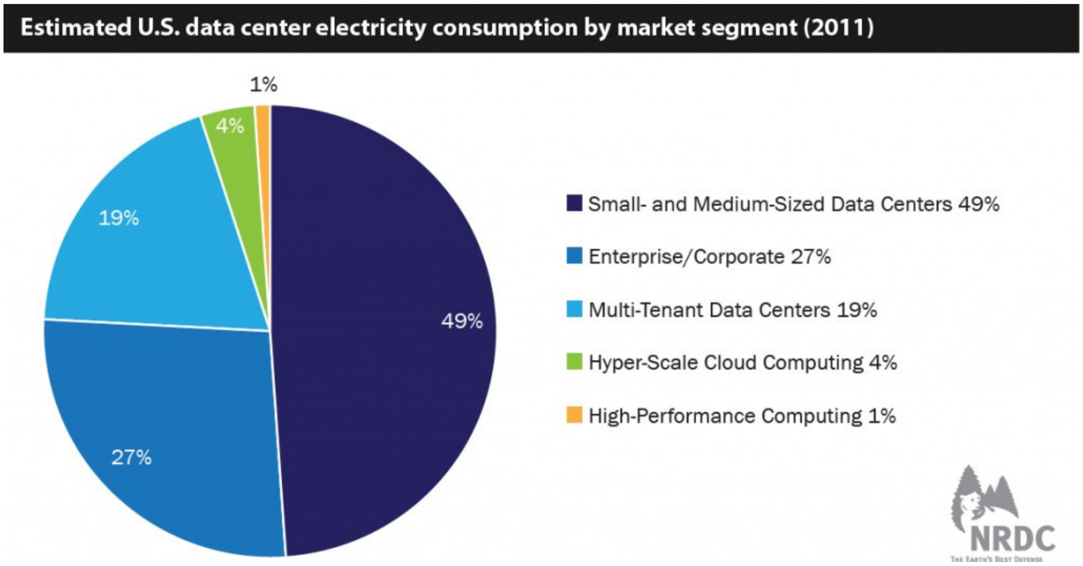


Figure 2. Estimated U.S. data center electricity use [4].

Since electricity cost varies based on location and even fluctuates with time, lowering energy cost has become a very important goal for data center companies. Figure 3 shows three of Google's data centers and their cost of electricity over time [6].

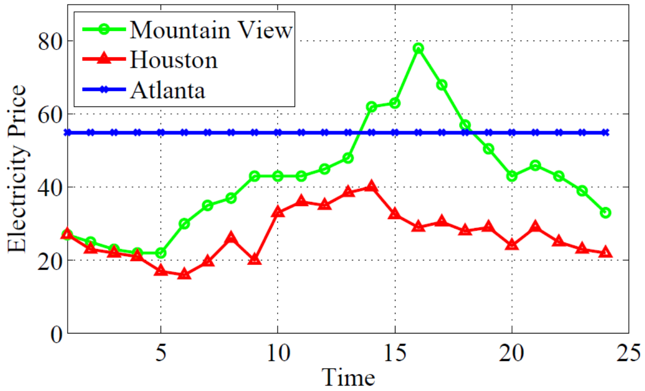


Figure 3. Google Data Center Electricity Cost [6].

In a survey conducted by Seagate in 2013 comparing Unites States of America (USA) and China data centers, both countries top challenge is data security. Figure 4 show the list and percentages of the top ten challenges of a data center in USA and China [17].

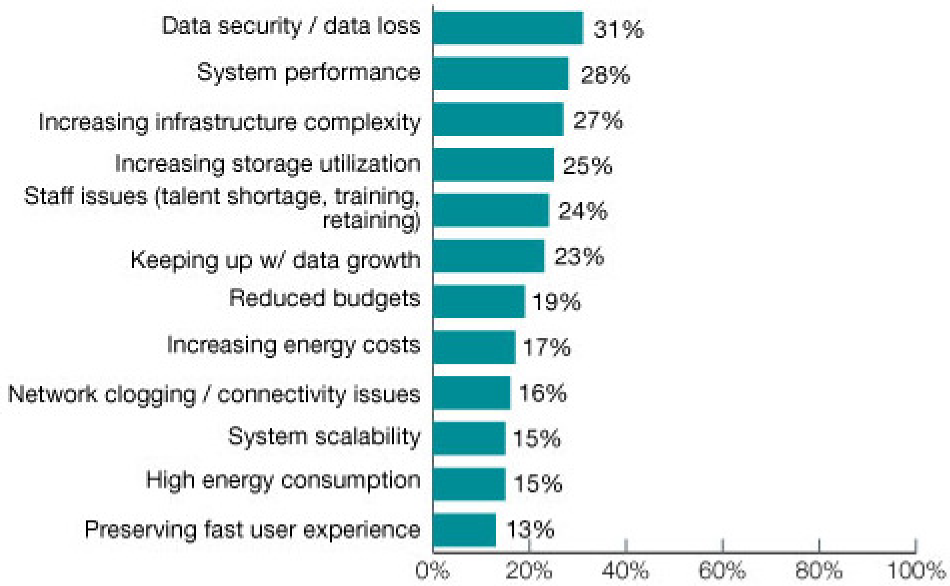


Figure 4. Top 10 data center challenges [17].

According to the survey, energy cost reduction is the number one challenge for data centers. Increase in energy cost at 17% and high energy consumption at 15%, making a total of 32% of the top 10 challenges of data centers. Security used to be the number one concern, but the rising cost of energy, has made energy reduction one of its priorities.

CHAPTER II

REVIEW OF LITERATURE

The rising cost of energy has led many researchers to find new ways to reduce energy usage at data centers. Some of the ways researchers are reducing energy usage include: the use of "on/off algorithms" [7], the use of environmental monitoring sensors, dynamic voltage and frequency scaling (DVFS), CPU consolidation [8], mechanisms to eliminate idle power waste [9], energy aware virtual machine replication and migration in data centers [10], power-aware geographical load balancing [11], and use of electrical energy storage systems [12].

One of the well-known ways to reduce energy consumption is the on/off algorithm, which consists of turning off unused or idle servers. It has been estimated that the idle machines use between 25-60% of peak power [7]. The on/off algorithm involves dynamically switching on and off servers based on the number of users, period of the day and a preset condition. Some on/off algorithms use data mining techniques in deciding when and how many servers to turn on/off. So depending on pervious usage, the algorithms dynamically turn on the minimum servers that are required to give a specific quality of service [13]. Other on/off algorithms involve the decision to turn as certain number of servers on/off regardless of users or time. This type of algorithm exploits the global system information, in terms of number of required working, idle, off, turning on, and turning off servers [7]. The on/off algorithm has been proven to reduce energy cost, but it degrades the quality of service the data center can provide.

Gartner Press Release states that IT produces 2% of the world's carbon emissions and it will reach 3% by 2020. As a trend of Green IT awareness, environmental sensors have been used to reduce energy usage and carbon emissions at data centers. In this method, readings from the sensor are used to deploy work for servers using a scheduling algorithm [14]. It was concluded that Green IT could be achieved with the combination of IT equipment and site infrastructure. This approach involves a cost for restructuring a data center and performance is also sacrificed.

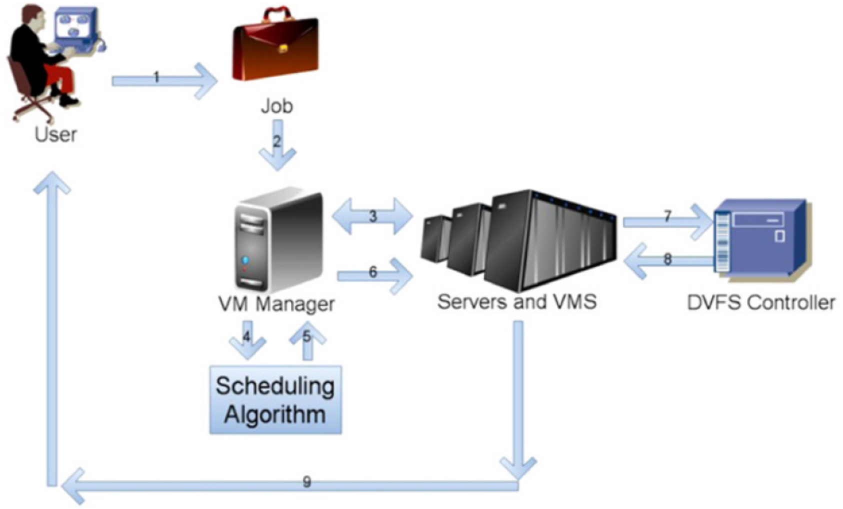


Figure 5. DVFS system architecture [Wu et al. 14].

Another method for reducing energy cost is the use of dynamic voltage and frequency scaling (DVFS). It is the process where processors are configured to achieve particular tradeoffs between computing speed and energy consumption [15]. DVFS enables the processors of servers to run at different frequencies with a given voltage to reduce the power consumption of the processor. So decreasing the processor voltage and frequency lower the energy usage and reduces cost. When execution performance is not is important, DVFS can reduce energy cost significantly. Other methods that use DVFS involve a scheduling algorithm. Some scheduling algorithms involve assigning more requests to data centers with low energy cost [6], while others use a pre-defined server maximum and minimum in assigning workloads so that servers with high energy consumption are controlled [16]. Figure 5 shows an example setup where DVFS and a scheduling algorithm are used.

CPU consolidation has been another way of reducing energy cost by pre-defining utility functions based on response time requirements for each application running in the data center. This method also involves the use of DVFS in each core of the server. Since higher execution frequency will result in a significant increase in power consumption, this approach uses DVFS for each core in a server and also a central resource manager and distributed local agent in assigning resources. Energy consumption is reduced by the use of local agents to parallelize the solution and decrease the decision making time, and the central manager utilizes optimization methods for the solving the request dispatching problem based on the results of the distributed agents [8]. This method uses multiple steps and constraints which delay the actual job to be processed. It does not also take in account of the amount of time that is wasted on decisions and assignments. In this thesis approach, the use of ACO allows quicker decision making and dispatching users’ requests, which allows the processing of more jobs faster and minimizing the energy usage.

A technique for eliminating idle power waste known as “PowerNap” is another way by which researchers reduce energy consumption. PowerNap refers to the minimization of idle power and transition time. PowerNap operates in low efficiency regions of current blade center power supplies. It conserves energy while system are idle since idle servers still draw about 60% of peak power [9]. This method focuses on conserving energy when idle and with some modification of hardware, but this thesis approach conserves energy during both processing and idle times.

Energy aware Virtual Machine (VM) replication and migration in data centers is a method by which cloud computing providers can reduce energy consumption with a little performance degradation. This is done by assigning dissimilar workloads to the same server, to reduce the number of active servers. Also, by placing multiple copies of a VM on different servers and distributing the workload among these VM copies, the need to activate more servers is reduced. Experimental results show a 20% reduction in energy consumption using this method [10]. This method does not utilize the whole capacity of a datacenter, has inactive or idle servers, and also degrade performance since fewer servers are used to process users’ requests. This thesis approach reduces energy consumption while utilizing most of the servers in the data center.

Power-aware geographical load balancing focuses on online service applications. Since data centers associated with cloud computing are often geographically distributed, energy is reduced by extending online application placement and migration based on predictions about the application lifetime, workload intensities, and dynamic energy prices. “Geographical load balancing (GLB) can be defined as a series of decisions about online assignment and/or migration of virtual machines (VMs) or computational tasks to geographically distributed datacenters in order to meet the service level agreements (SLAs) or service deadlines for VMs/tasks and to decrease the operational cost of the cloud system [11].” This process reduces energy by performing periodic VM placement and migration management for online service application based on the prediction of application active periods, workload types and intensities, and electrical energy prices, and assigns requests to minimize the cost of energy [11]. Even though decisions and assignment are made based on previous usage, there is no guarantee that the trend will continue. The simulation of this algorithm shows a 27% to 40% of energy reduction, but a change in trend might degrade its performance. This thesis algorithm find ways to assign workload to data center to reduce energy cost without depending on trend that could easily be changed.

The use of electrical energy storage systems is another method used to reduce energy cost at data centers. Energy storage devices (ESD) are used to supplement the data center energy usage during peak times and store energy during normal operation times. Since power saving require frequent discharge/charge batteries, availability and lifetime of batteries are limited [12]. The method has led to a 28% of savings on energy cost, but requires constant ESD replacements.

All the methods and algorithms presented here in this chapter has been proven to reduce energy consumption either in an experimental environment or in the real world. This paper presents a another algorithm that does not require purchasing of new equipment, or redesigning or restructuring data centers, but assigns users’ requests to data centers such that the whole geo-distributed data centers’ energy is minimized using Ant Colony Optimization which mimics natural ants.

CHAPTER 3

METHODOLOGY

This thesis explores another method by which energy usage at data centers could be reduced by using Ant Colony Optimization (ACO) on a Quadratic Assignment problem (QAP). Since the activities at a data center involve matching users' requests to a server resource, which is a form of QAP, then applying ACO, which is known to produce good results with Nondeterministic Polynomial Complete (NP-Complete) problems, should produce better results.

Even though NP Complete problems are said to be some of the most complex problems in computer science, researchers are looking for ways to solve them due to their vast applications in the real world [18]. Also NP-Complete problems are known to be equivalent to each other, therefore, a good algorithm for one problem could be applied to another problem [19]. There are many problems that have been proven to be NP-Complete. Besides QAP, some NP-Complete problems include: integer programming, capacitated minimum spanning tree, longest path problem, shortest weight-constrained path, metric k-center, subgraph isomorphism problem, Euclidean minimum spanning tree, subset sum problem, shortest common supersequence, and string-to-string correction problem. There are two ways to solve an NP-Complete problem: (1) exact solution approach and (2) approximation approach. Exact solution exhausts all possible combinations and select the minimum cost path as solution; the computation time grows exponentially with the size of the problem. Approximation approach is fast, but does not guarantee an optimal solution but a near optimal solution in a reasonable computational effort. Some examples of approximate algorithms include: Closest neighbor, Greedy, Insertion, Christofide, Genetic algorithm, and Ant colony optimization. Since the approximation approach does not give the optimal solution, Tour Improvement algorithms such as 2-opt, 3-opt, Lin-Kernighan, tabu search, and simulated annealing are used to enhance the solution [20].

The Quadratic Assignment Problem (QAP) is said to be one of the most difficult optimization problems to solve optimally. QAP was introduced by Koopmans and Beckman in 1957 as a model for location problems. The development of algorithms, either exact or approximate, is challenging and has been studied by Operations Research/Management Science, Industrial Engineering, and Computer Science [21]. QAP is the problem of assigning *n* facilities to *n* locations so that the cost of the assignment, which is a function of the way facilities have been assigned to locations, is minimized. QAP deals with two data sets, and the algorithm matches the sets on a one-to-one ratio such that the cost is minimized. The exact solution approach is appropriate for small data sets, but large data sets cannot be solved in a reasonable time due its computational limits. Approximation approaches such as Simulated Annealing, Neural Networks, Genetic Algorithms, Tabu Search, and Ant Colony Optimization have a reputation for producing good solution within a reasonable amount of time [31].

Ant Colony Optimization (ACO) mimics the movements of ants in the search for food, and the return to their nest. The minimum path between an ant's nest and food source becomes the best solution to ACO. The ants leave a trail of pheromones on their path, which also guide other ants to that possible food source. In ACO ants are placed in random locations, and are allowed to move from one location to the other without visiting the same location more than once. The ants leave a pheromone trail as they move along a path. In ACO the pheromone trail is taking into account as the ants move from one location to the other, and the pheromone trail evaporates over time. The ant with the shortest path has the strongest pheromone trial, making it the easiest for other ants to follow [20]. Figure 6 shows the process of the ACO.

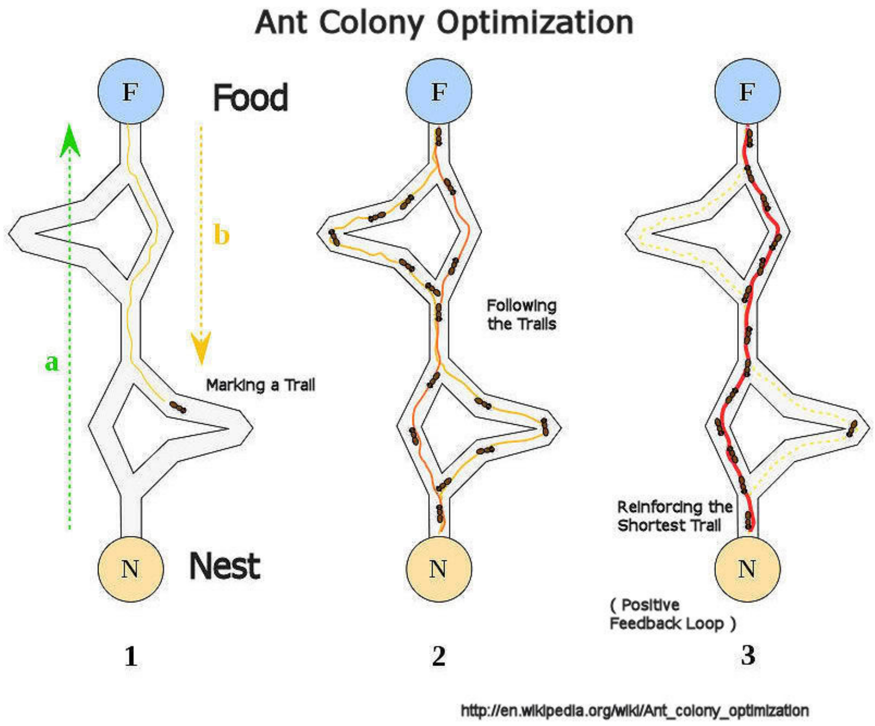

Ant Colony Optimization
F
Food
b
a
Marking a Trail
N
Nest
1
F
Following the Trails
N
2
F
Reinforcing the Shortest Trail
N
(Positive Feedback Loop)
3
http://en.wikipedia.org/wiki/Ant_colony_optimization

Figure 6. ACO Process [Image from wikipedia.org]

Ants display behaviors that have long fascinated human beings. A surprising behavioral pattern displayed by ants is the ability to find the shortest paths between their nest and food source. This behavioral pattern of ants has inspired computer scientists to develop algorithms for optimization problems. Introduced in the early 90's by Marco

Dorigo, ACO algorithm success is demonstrated by the wide range of problems to which it has been applied [22].

Some of the early problems ACO was applied to since its introduction includes: Sequential Ordering Problem, which deals with “finding a minimum weight Hamiltonian path on a directed graph with weights on the arcs and on the nodes, subject to precedence constraints among nodes” [23], [24], Vehicle Routing Problem [25], Traveling Salesman Problems which ACO has a great success when applied [20], [26], and Quadratic Assignment Problem which is a way of minimizing the cost in mapping two sets of data on a one-to-one bases. Many experimental methods have been developed to solve QAP’s using a variety of procedures. Some of these methods includes: simulated annealing of Connolly, the tabu search of Taillard, hybrid genetic-tabu search of Fleurent and Ferland and scatter search by Michelon and Tavares [27]. Some of the recent application of ACO on QAP includes Biobjective QAP [28], intelligent fault diagnosis of rotating machinery [29], website structure improvement [30], and distributed database design [32].

In this paper, ACO is applied to QAP on Optimal Server Activation and Task Placement Algorithm to minimize energy cost in a geo-distributed data centers. Two programs were implemented based on Optimal Server Activation and Task Placement Algorithm. (1) Optimal Server Activation and Task Placement Algorithm (algorithm presented on page 8 of Optimal Task Placement with QoS Constraints in Geo-distributed Data Centers using DVFS article [6], and (2) Ant Colony Optimization for Optimal Task Placement with QoS Constraints in Geo-distributed Data Centers using DVFS. For simplicity, throughout this paper, algorithm 1 will be referred to as QoS and 2 as AntforQoS.

In this paper, a user request or resource request is defined as any request from users to the data centers for any service provided by the data center (Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS)). The experimental results presented in this paper assume that the cost for processing one request takes one second, and also the energy use is charged per second.

3.1 Optimal Server Activation and Task Placement algorithm (QoS)

The Optimal Server Activation and Task Placement algorithm (QoS) is a new algorithm that has been proven to reduce energy cost at data centers. In this thesis, the QoS is used as baseline to compare our approach of energy reduction. The QoS algorithm involves the assignment of users' resource requests to a server in geo-distributed data centers. Figure 7 shows an example of a geo-distributed data center.

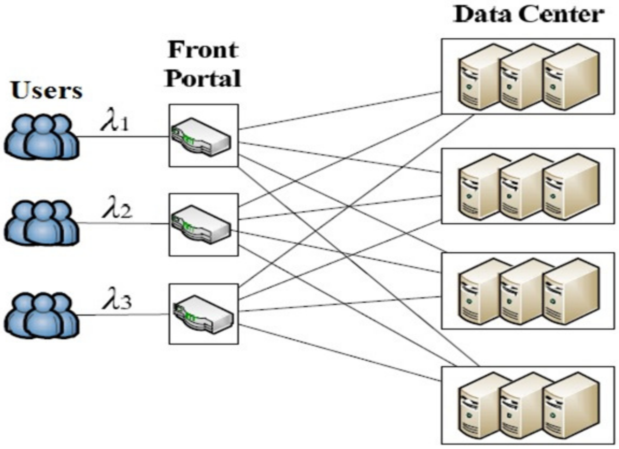
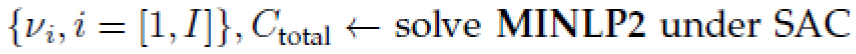


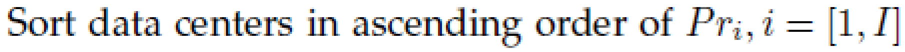
Figure 7. Example of geo-distributed data center.

In the algorithm, the total number of request is first uniformly distributed across all active servers in the geo-distributed data center. Then the cost for processing the request is compared, the data centers with high costs then migrate some of their requests to a data center with cheaper energy cost. To always comply with the SAC format, a binary-search concept is used to find the number of servers in the most expensive activated data center to deactivate some of its servers, and activate more servers in data centers with the least cost. In the Optimal Server Activation and Task Placement Algorithm (QoS algorithm presented on page 8 [6]), users resource request is allocated to data centers satisfying the optimal cost.



$\{\nu_i, i = [1, I]\}, C_{\text{total}} \leftarrow$ solve **MINLP2** under SAC .

First, the data centers are sorted based on their energy cost.



Sort data centers in ascending order of $Pr_i, i = [1, I]$

Pr is the energy cost at a data center. In order to allocate users request correctly, the order of the data center has to be changed based in their energy cost. Example, suppose there are three data centers:

No	**Servers**	**Energy Cost**
1	15000	0.4
2	65000	0.2
3	10000	0.3

First, the order of data center is " 1,2,3" Data centers are sorted by Pr value.

No	**Servers**	**Energy Cost**
1	65000	0.2
2	10000	0.3
3	15000	0.4

Now the order is " 2, 3,1 ". In this example, the data center that has the minimal *Pr* is the second data center, next is the third, followed by the first data center, given us 2, 3, 1 as the order. So, more user requests will be allocated to the second data center, next third data center, and then the first data center.

The allocation of users' requests is done with the following rule; if *Pr* value (electric power cost) is less, more user requests is allocated to that date center. The algorithm is designed in two phases. In the first phase, Server Activation Configuration (SAC) with the maximum servers that shall be activated and its optimal cost is obtained (line 1 to 22) [6], and in the second phase, which is also divided in to two, request are assigned to data centers in such a way that more request are sent to the data centers with the cheap energy cost since most of its servers with be activated during the first phase. During the first part of the second phase, in each data center, activated servers are minimized, and based on their value, electricity power cost is obtained in the second part.

$$\min_{\nu_i, N_i} : \sum_{i=1}^{I} N_i \cdot (C \cdot (\frac{\nu_i}{\gamma \cdot N_i} + \frac{1}{\gamma \cdot D})^{\alpha} \cdot \frac{\nu_i}{\gamma \cdot N_i} + P_{\text{static}}) \cdot Pr_i,$$

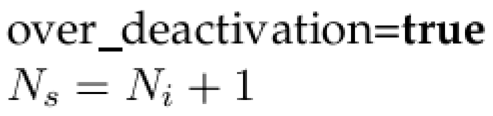
The above formula gets the total optimal cost, and also, N_i (activated servers in each data center) is minimized. Hence, an optimal cost is obtained based on the minimization of activated servers.

In processing of assigning users request to servers, an initial SAC is obtained with the maximum possible number of activated servers it is updated with the following formula.

$$\text{SAC} \leftarrow \{N_i = S_i, i = [1, I]\},$$
$$I_{\text{max}} = I$$

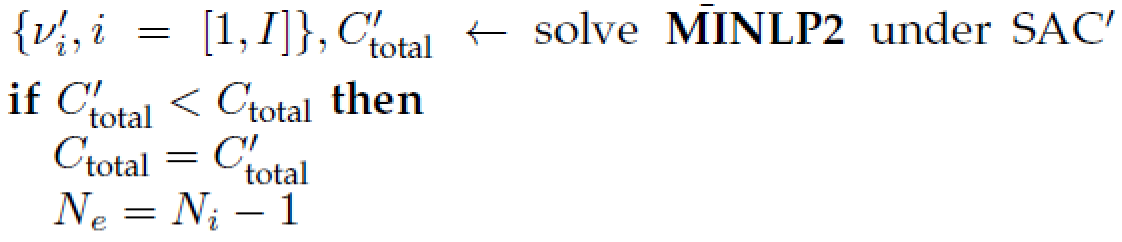
To get the optimal activated servers in each data center,

```
$N_s = 0,\ N_e = N_i$
while $N_s \le N_e$ do
    update SAC$'$ by setting $N_i = \lfloor \frac{N_s + N_e}{2} \rfloor$
```



```
over_deactivation=**true**
$N_s = N_i + 1$
```

in the second phase, as mentioned above, the total optima cost is obtained based on the number of activated servers. The optimal cost is obtained by the steps below.



```
$\{\nu_i', i = [1, I]\}, C'_{\text{total}} \leftarrow$ solve $\bar{\mathbf{M}}\mathbf{INLP2}$ under SAC$'$
**if** $C'_{\text{total}} < C_{\text{total}}$ **then**
  $C_{\text{total}} = C'_{\text{total}}$
  $N_e = N_i - 1$
```

This part of the algorithm gets the total cost. Requests are assigned to data centers in such a way that more requests are sent to the data centers with the cheapest energy cost since most of its servers were activated during the first phase.

The Optimal Server Activation and Task Placement algorithm reduce energy cost by deactivating servers in a data center with a high energy price and activating more servers in a locations where the energy cost is cheaper. Therefore more user requests are assigned to the data centers with cheap energy cost, hence, reducing the energy cost in the distributed data centers.

3.2 ACO for Optimal Server Activation and Task Placement algorithm (AntforQoS)

The difference between AntforQoS and QoS algrothim is their decision making process. In the QoS, resource requests are allocated based on reordering and assigning most of the requests to data centers with the minimum cost, whereas in the AntforQoS, resource requests are allocated based on the Ant Colony Optimization algorithm. Algorithms 1 and 2 below is the pseudo code for the AntforQoS.

Algorithm 1 - AntforQoS

```
Input : I, J , $\lambda_j$ ,$P_{r_i}$ , $S_i$,  Q , $Pheromone_i$ = 1, antnum, iterationMax, iteration = 0, i = [1,I],  j = [ 1, J],  evaporation, P_Static, alpha, gamma, C ;
Output : $C_{total}$ , $V_i$ , i = [ 1, I ]
Sort data centers in ascending order of $P_{r_i}$, i = [1, I ]
for i = 1 To I
    $V_i^{min}$ = $argmin_v$(Pi( Si , V ) ≤ Pi( Si - 1 , V ))
end for
if $\sum_{j=1}^{J} \lambda_j < \sum_{i=1}^{I} \lambda_i^{min}$ then
    SAC ⟵{ Ni = 0, ∀i = [1, I ] }
    $V^{\gamma} = \sum_{j=1}^{J} \lambda_j$
    for i = 1 to I do
        if $V^{\gamma} < V_i^{min}$ then
            Ni = argminN (P (N, $V^{\gamma}$) ≤ P (N – 1, $V^{\gamma}$))
            break;
        else if $V^{\gamma} \geq V_i^{min}$ then
            Ni = Si
            $V^{\gamma} = V^{\gamma} - V_i^{min}$
        end if
    end for
    Imax = i;
else
    SAC ⟵{ Ni = Si, i = [1, I ] } (SAC - Server Activation Configuration)
    Imax = I;
end if
{ $V_i$, i = [1, I ] }, $C_{total}$ ← solve Ant Colony System under SAC
        $C_{total}$ = Ant_Cost(N)
SAC′ = SAC, $V_i' = V_i$, ∀i = [1, I ]
over_deactivation = false
for i = Imax to 1 do:
    Ns = 0, Ne = Ni
    while Ns ≤ Ne do:
        update SAC′ by setting Ni = $\lfloor \frac{Ns+Ne2}{2} \rfloor$
        { $V'_i$, i = [1, I ] }, $C'_{total}$ ← solve Ant Colony System under SAC'
         $C'_{total}$ = Ant_Cost(N)
        if $C'_{total} < C_{total}$ then
            $C'_{total} = C_{total}$
            Ne = Ni − 1
        else
            over_deactivation = true
            Ns = Ni + 1
        end if
    end while
    if over_deactivation == true then
        return;
    end if
end for
```

Algorithm 2 – Ant Colony Optimization

```
// Initialize all ant parameters
Input : $Optimal\_Ant\_Cost_j$ ; cost = 0, $\mathrm{delta\_cost}_i$ , $ant\_road_{ant,i}$ , $Ant\_Cost_{ant}$
        Total_Request, ant = [1, antnum], i = [1, I], j = [1, iteratinMax];
Output : Optimal_Cost;

While( iteration < IterationMax):
    for ant = 1 To antnum:
        cost = 0; delta_cost[] = 0;
        //each ant complete its road by pheromone value;
        // ( each ant select road that has more pheromone.)
        Generate_Road( I, $ant\_road_{ant}$, Pheromone, Total_requests);
        // get cost of each ant.
        for i = 1 To I :
            $\mathrm{delta\_cost}_i$ = $N_i$*(C*Math.pow(( $ant\_road_{ant,i}$ / (gamma* $N_i$) +
                1/(gamma*D)),alpha) * $ant\_road_{ant,i}$ /(gamma* $N_i$)+ P_Static)* $\mathrm{P}_{r_i}$;
            cost = cost + $N_i$*(C*Math.pow(( $ant\_road_{ant,i}$ / (gamma* $N_i$) +
                1/(gamma*D)),alpha) * $ant\_road_{ant,i}$ /(gamma* $N_i$)+ P_Static)* $\mathrm{P}_{r_i}$;
        end for
        //evaporate pheromones;
        for i = 1; To I
            $\mathrm{Pheromone}_i$ = (1 - evaporation)* $\mathrm{Pheromone}_i$ ;
        end for
        // apply pheromone update;
        for i = 1; To I
            $\mathrm{Pheromone}_i$ = $\mathrm{Pheromone}_i$ + Q / $\mathrm{delta\_cost}_i$ ;
        end for
        $Ant\_Cost_{ant}$ = cost;
    end for
    Min_Cost = MaxValue;
    for ant = 1 To antnum:
        if Min_Cost > $Ant\_Cost_{ant}$ then
            Min_Cost = Ant_Cost[i];
        end if
    end for
    // the best ant cost is saved into global cost;
    $Optimal\_Ant\_Cost_j$ = Min_Cost;
    iteration = iteration + 1;
end while:
// Get Optimal Solution
Optimal_Cost = MaxValue;
for j = 1 To iterationMax :
    if Optimal_Cost > $Optimal\_Ant\_Cost_j$ then
        Optimal_Cost = $Optimal\_Ant\_Cost_j$ ;
    end if
end for
return Optimal_Cost ;
```

The general idea is to minimize energy cost in a geo-distributed data center using ACO to assign users' resource request to servers in a data center. ACO is used to decide the cost effective way for servicing the user's request. In this algorithm, energy cost is reduced by the use of the front portals (see figure 7) as ants, and use of the pheromone rule in the ACO algorithm (evaporate pheromone and update pheromone). User requests are then assigned to each data center by the pheromone value, and if the pheromone element value corresponding to each ant is greater than other pheromone element, more users are assigned to that data center.

CHAPTER 4

FINDINGS

In order to test how ACO for Optimal Server Activation and Task Placement (AntforQoS) will perform compared to some of the algorithms that are already in use, Optimal Server Activation and Task Placement Algorithm (QoS) [6] and AntforQoS were implemented. This is a simplified simulation of the real world applications in use. Many data centers house thousands of servers that serve millions of people at a time, and these servers consume megawatts of electrical energy costing millions of dollars each year [6]. This thesis presents a streamlined simulation where the input data were randomly created with user resource request ranging from 0 to 50, and each number in the input file represent one user's request. In the experiments, we assume that it takes 1 second to process 1 request, and the cost is measured in seconds. The programs then calculates the number of users, combines the total resources requested, and distributes the users' request efficiently amount the geo-distributed data centers to reduce energy consumption. Below are some of the experimental results from comparing QoS and AntforQoS on small and large data sets.

Experiment 1 is a simulation to see how AntforQoS will perform when compared to QoS. In experiment 1, a comparison of the two algorithms were done base on the examples in [6], listing the numbers of servers Google has in 3 of its data center locations, Mountain View, Houston and Atlanta, having 15000, 30000 and 10000 respectively. Table 1 show the result of the simulation with users ranging from 30,000 to 110,000 and their combined resource request from 761,548 to 2,805,897. Figure 8 shows the growth in cost as users' request increase, and figure 9 also shows the how the two algorithms distribute users' request among the data centers to reduce energy cost.

Experiment 1 – 3 geo-distributed data centers. (Number of users – 30,000 to 110,000)

Number of users	Total resources request	QoS Processing Cost	AntforQoS Processing Cost
30,000	761548	5.276980	5.752068
40,000	1019682	8.141977	7.997873
50,000	1277766	10.957430	10.490628
60,000	1532466	13.735973	13.165937
70,000	1784032	16.594413	16.334056
80,000	2036067	20.018277	19.925638
90,000	2301017	23.617597	23.367127
100,000	2551898	27.025807	26.857439
110,000	2805897	30.476353	30.166278

Table 1 – Cost for processing users request (Small Scale Input).

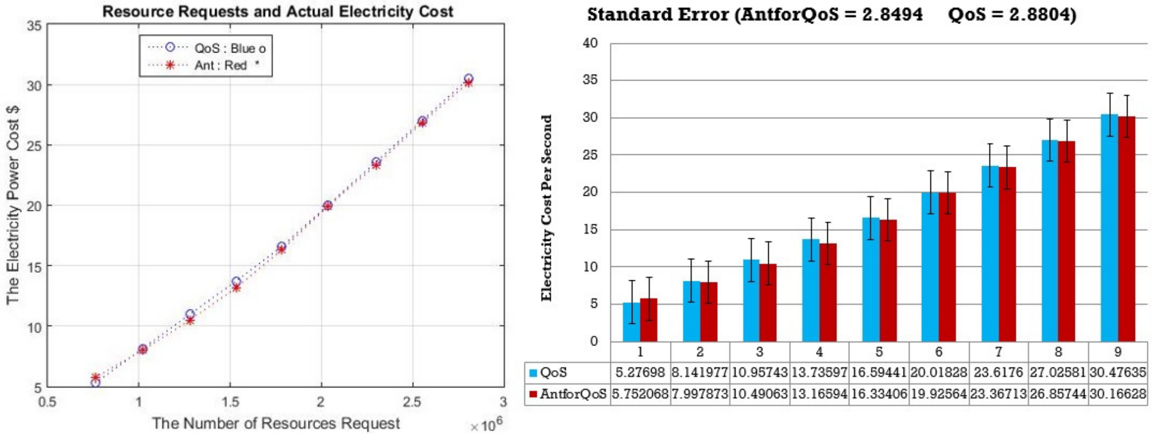


Figure 8 – A comparison of QoS and AntforQoS energy cost.

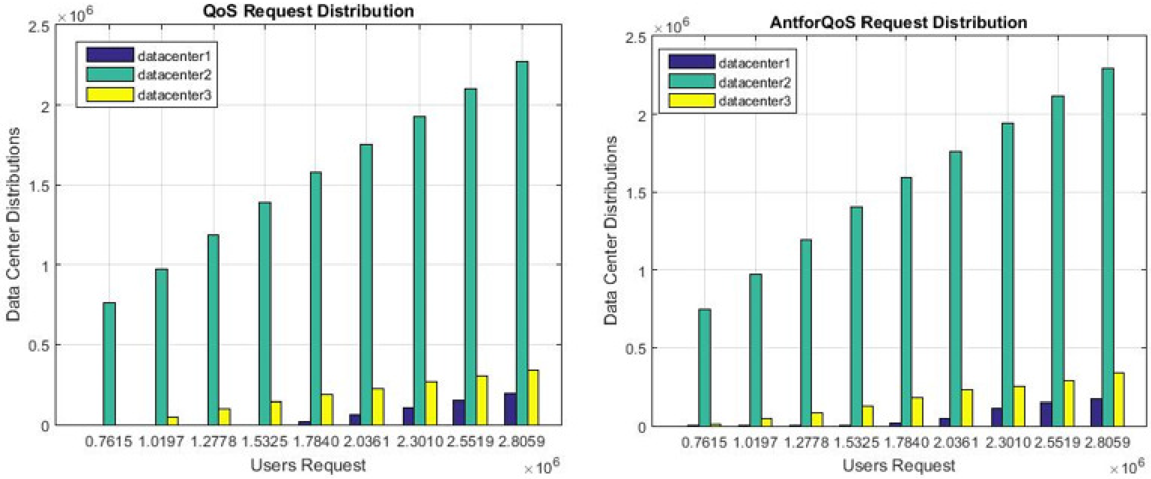


Figure 9 – QoS and AntforQoS users request distribution.

In experiment 2, following the same setup as experiment 1, servers in the cheapest energy location were increased, ranging from 30,000 to 65,000. The result in this experiment show that energy cost can be reduced in geo-distributed data centers by increasing the number of servers at a data center location with lower energy cost. Figure 10 depicts the result of the experiment, showing the decrease in cost as the number of servers in data center 2 increases.

Experiment 2 – 3 geo-distributed data centers. (Number of users – 30,000 to 110,000)

In experiment 2, the result shows that energy cost is saved by increasing the number of servers at one location (location with the least cost of energy).

No	Servers	Electricity power cost
1	15000	0.4
2	**30000**	**0.2**
3	10000	0.3

Changes in the number of servers at data center No. 2

No	Servers	Electricity power cost
2	35000	0.2
2	40000	0.2
2	50000	0.2
2	60000	0.2
2	70000	0.2

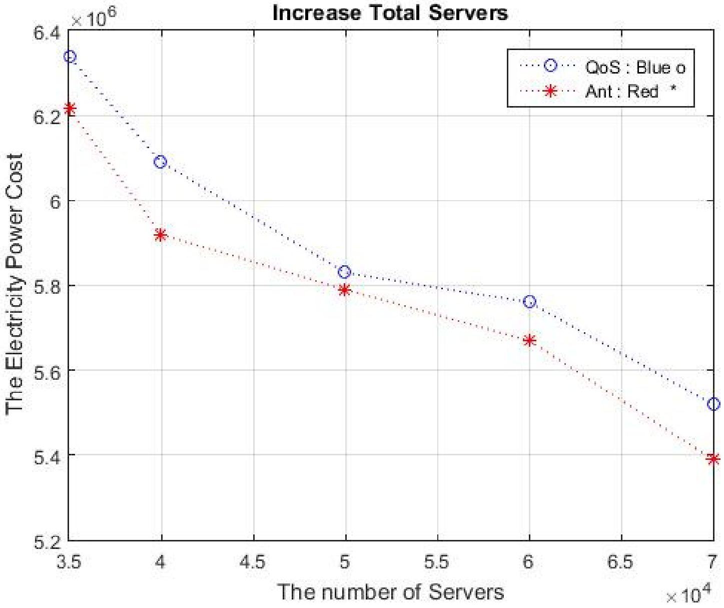


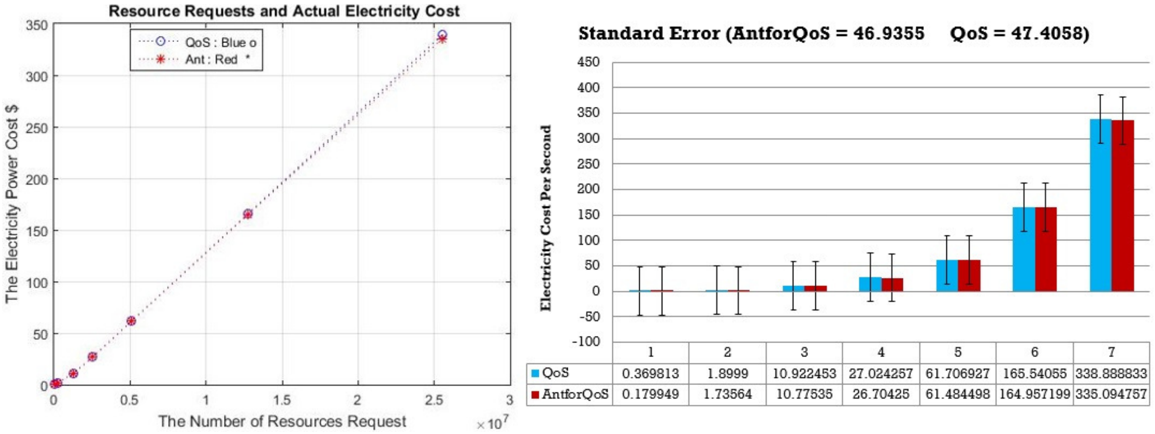
Figure 10 – Increase in servers

Experiment 3 was conducted to see how QoS and AntforQoS will perform with large data sets. In this experiment, the number of users range from 1,000 to 1,000,000 and with users request ranging from 25,473 to 25,508,481. Table 2 show the result of the experiment, as also depicted in figure 11. The results indicates that AntforQoS performs better in assigning users' requests to data centers to reduce energy cost. Figure 12 shows the distribution of users' requests by the two algorithms to reduce energy cost.

Experiment 3 – 3 geo-distributed data centers. (Number of users – 10,000 to 1,000,000)

Number of users	Total resources request	QoS Processing Cost	AntforQoS Processing Cost
1000	25473	0.369813	0.179949
10000	254986	1.899900	1.735640
50000	1274560	10.922453	10.775350
100000	2551785	27.024257	26.704250
200000	5104816	61.706927	61.484498
500000	12748122	165.540550	164.957199
1000000	25508481	338.888833	335.094757

Table 2 – Cost for processing users request (Large Scale Input).



	1	2	3	4	5	6	7
■ QoS	0.369813	1.8999	10.922453	27.024257	61.706927	165.54055	338.888833
■ AntforQoS	0.179949	1.73564	10.77535	26.70425	61.484498	164.957199	335.094757

Figure 11 – QoS and AntforQoS energy cost on Large Scale Input with 3 data centers.

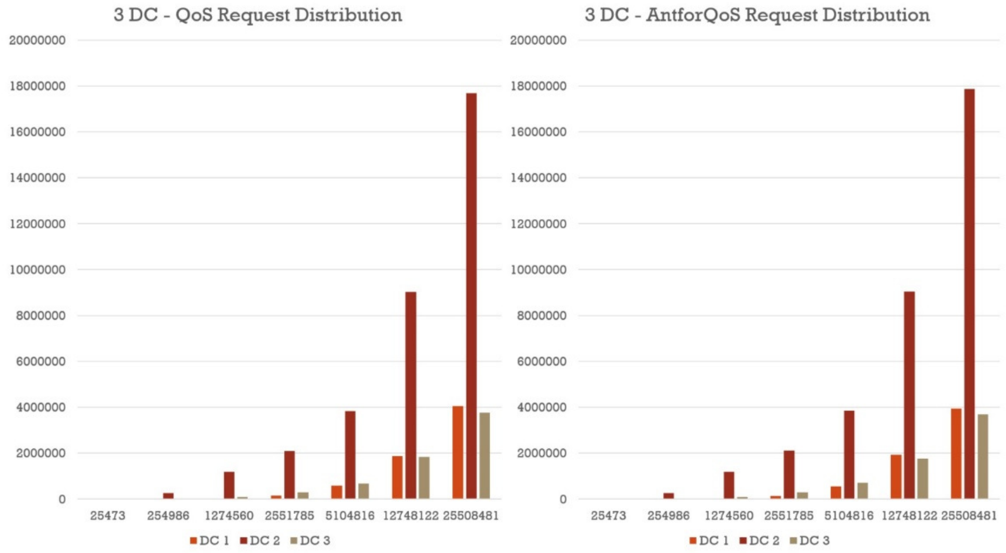


Figure 12 – QoS and AntforQoS users request distribution of Large Scale Input.

Experiment 4 was conducted to compare QoS and AntforQoS to see how they will perform with an increase in datacenters. In this experiment, just as experiment 3, the number of the geo-distributed data centers where changed from 3 to 5. Table 3 show the output of the simulation. As showed in figure 13, in this experiment, both algorithms performed almost at the same rate until users request was greater than 3 million. Figure 14 shows the distribution of users request to the geo-distributed data centers.

Experiment 4 – 5 geo-distributed data center. (Number of users – 10,000 to 1,000,000)

Number of users	Total resources request	QoS Processing Cost	AntforQoS Processing Cost
10000	254986	1.474925	1.499299
50000	1274560	6.768908	7.492913
100000	2551785	14.067337	14.595963
200000	5104816	34.164615	33.468033
500000	12748122	111.871618	105.802473
1000000	25508481	249.821441	227.477992

Table 3 – Cost for processing users request (Large Scale Input with 5 Data centers)

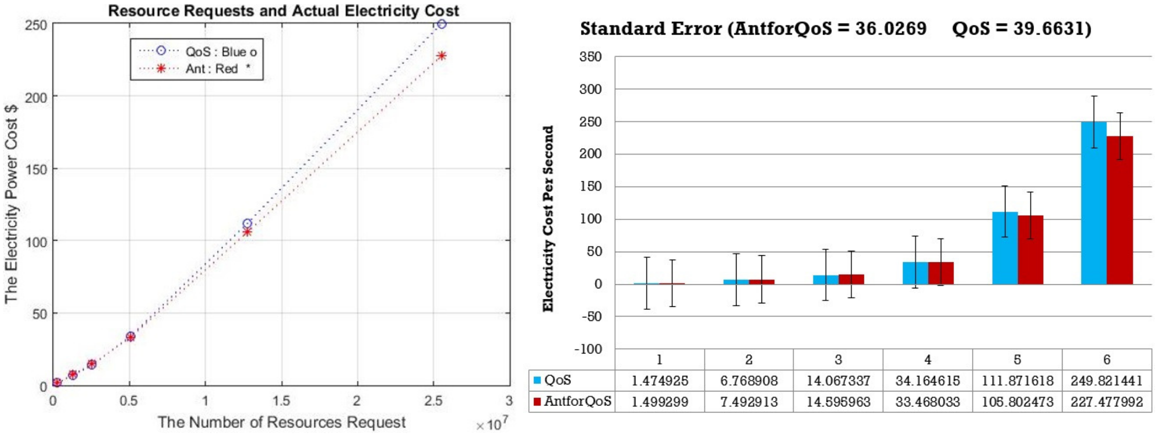


Figure 13 – QoS and AntforQoS energy cost on Large Scale Input with 10 data centers.

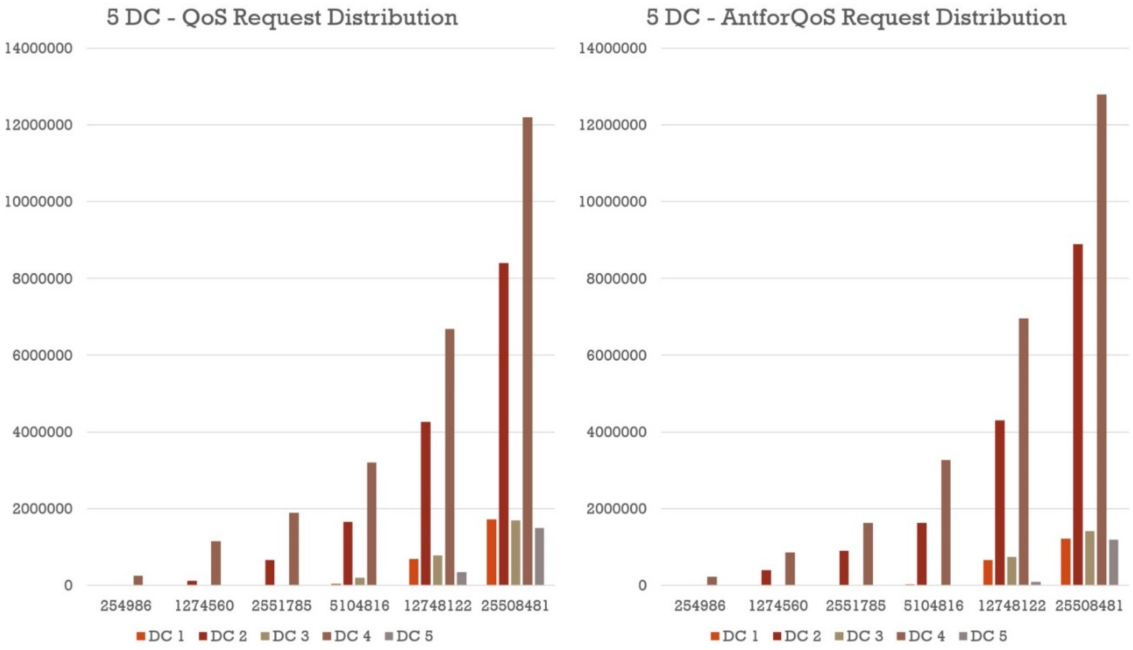


Figure 14 – QoS and AntforQoS users request distribution of Large Scale Input with 5 data centers.

Experiment 5 was conducted similarly to experiment 4, but with more data centers. The number of data centers in this simulation is 10, and table 4 shows the output results from the simulation. As the results indicated in figure 15, AntforQoS perform worse compared to QoS when there are more data centers. This is because the data size remains the same while the number of data center increases. The AntforQoS algorithm spread users request across the data centers while in the QoS, continues to send request to the data centers with low energy cost, just like in experiment 4. Figure 16 shows the distribution of the users' requests to the geo-distributed data centers for both algorithms. This simulations has shown that, the ACO can optimize an Optimal Server Activation and Task Placement Algorithm, and it performs better as the input data increases.

Experiment 5 – 10 geo-distributed data center. (Number of users – 10,000 to 1,000,000)

Number of users	Total resources request	QoS Processing Cost	AntforQoS Processing Cost
10000	254986	1.274930	2.046548
50000	1274560	6.510078	9.924022
100000	2551785	13.534817	22.942775
200000	5104816	29.448777	36.275398
500000	12748122	86.063679	98.541214
1000000	25508481	202.405709	276.95351

Table 4 – Cost for processing users request (Large Scale Input with 10 Data centers)

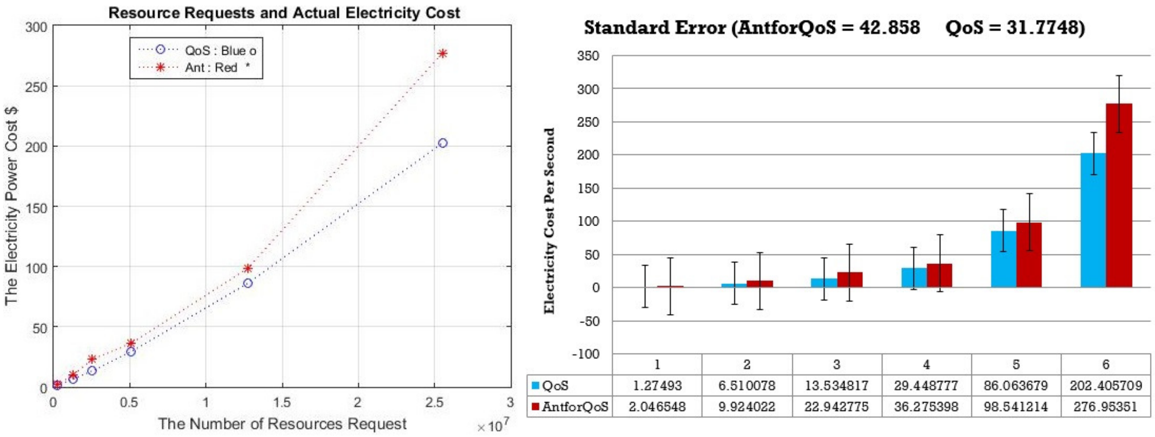


Figure 15 – QoS and AntforQoS energy cost on Large Scale Input with 10 data centers.

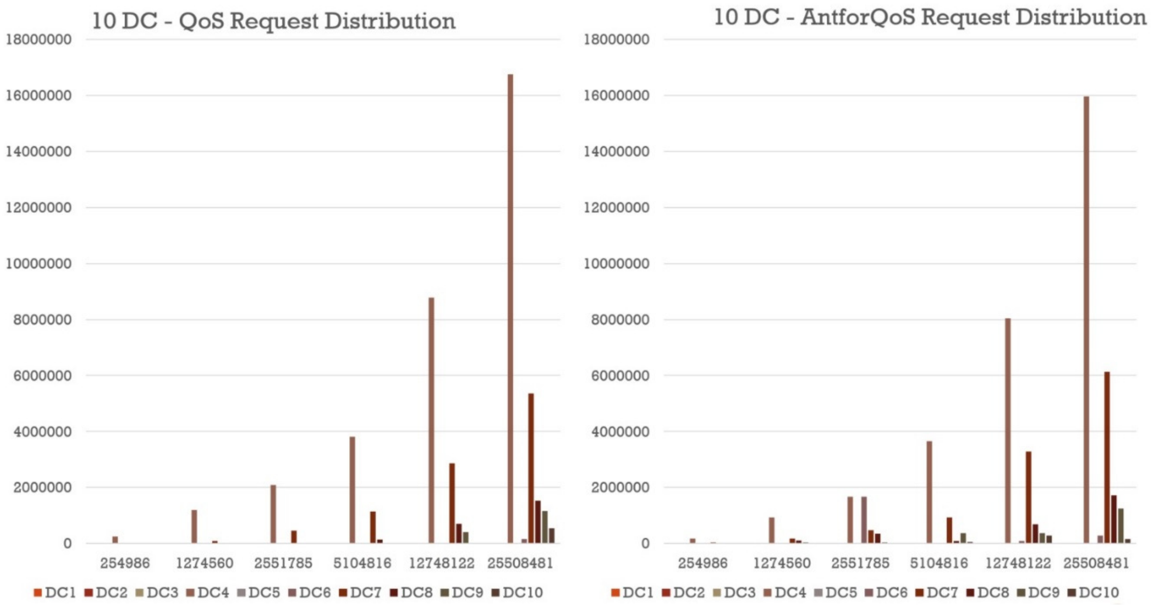


Figure 16 – QoS and AntforQoS users request distribution of Large Scale Input with 10 data centers.

The experimental results demonstrate that using Ant Colony Optimization on a Quadratic Assignment Problem like Optimal Server Activation and Task Placement reduces the energy consumption in geo-distributed data centers. This is because using ACO algorithm finds a near optimal solution in a reasonable computational time for routing users' resource request in a cost effective way to data centers. And the results has also show that AntforQoS work best with fewer geo-distributed data centers when compared to the QoS algorithm and its performance advantage increases with the input data.

CHAPTER 5

CONCLUSION

Electricity cost has become the leading operational expenditure at data centers due to the increase in demand for cloud computing. This paper has explored another approach to reduce energy cost by using the Ant Colony Optimization algorithm for scheduling users' requests in geo-distributed data centers. The ACO was formulated based on Optimal Server Activation and Task Placement Algorithm, which is a DVFS-aware data center management and request scheduling algorithm. In this paper, ACO is used in place of the scheduling algorithm to increase energy cost savings. The experimental results show an improvement for using ACO on a QAP such as minimizing energy cost through effective scheduling for geo-distributed data centers. Based on the experiments conducted, ACO (AntforQoS) performance advantage increases as the input data grows.

Although, the experimental results do not show a substantial difference in the two algorithms, taking in account that it represent a per-second energy cost for processing users' requests makes a huge difference in energy cost savings.

REFERENCES

[1] Liu, Shuo, et al. “Power minimization for data center with guaranteed QoS.” *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition*, pp. 1347-1352, March 2015.

[2] Wu, Caesar, and Rajkumar Buyya. *Cloud Data Centers and Cost Modeling: A Complete Guide To Planning, Designing and Building a Cloud Data Center*. Morgan Kaufmann, 2015.

[3] Google “Data Center Locations.” *Google Data Centers*. 2015. Web. 18 June 2015. http://www.google.com/about/datacenters/inside/locations/index.html

[4] Natural Resources Defense Council, “Data Center Efficiency Assessment.” *America's Data Centers Consuming and Wasting Growing Amounts of Energy*. August 2014. Web. 18 June 2015. http://www.nrdc.org/energy/data-center-efficiency-assessment.asp.

[5] Shan, You-cheng, Chao Lv, and Wen-bo Cui. "A Pilot Study on the Application of Cloud CRM in Industrial Automation." *Proceedings of the 21st International Conference on Industrial Engineering and Engineering Management 2014*. Atlantis Press, pp. 177-180, 2015.

[6] Gu, Lin, et al. "Optimal Task Placement with QoS Constraints in Geo-distributed Data Centers using DVFS." *Computers*, Vol. 64, No. 7, pp. 2049-2059, August 2014.

[7] Callau-Zori, Mar, et al. "MERCi-MIsS: Should I Turn off My Servers?. "*Distributed Applications and Interoperable Systems. Springer International Publishing,* Vol. 9038, pp. 16-29, Grenoble, France, June 2015.

[8] Wang, Yanzhi, et al. "Resource allocation and consolidation in a multi-core server cluster using a Markov decision process model." *Quality Electronic Design (ISQED), 2013 14th International Symposium on*. IEEE, March 2013.

[9] Meisner, David, Brian T. Gold, and Thomas F. Wenisch. "PowerNap: eliminating server idle power." *ACM Sigplan Notices*. Vol. 44. No. 3, pp. 205-216, March 2009.

[10] Goudarzi, Hadi, and Massoud Pedram. "Energy-efficient virtual machine replication and placement in a cloud computing system." *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*. pp. 750-757, June 2012.

[11] Goudarzi, Hossein, and Massoud Pedram. "Geographical load balancing for online service applications in distributed datacenters."*Cloud Computing (CLOUD), 2013 IEEE Sixth International Conference on*. IEEE, 2013.

[12] Zheng, Weiye, Kwan-Liu Ma, and Xiongfei Wang. "TE-Shave: Reducing Data Center Capital and Operating Expenses with Thermal Energy Storage." *Computers, IEEE*, Vol. PP, No. 99, January 2015.

[13] Chen, Gong, et al. "Energy-Aware Server Provisioning and Load Dispatching for Connection-Intensive Internet Services."*NSDI*, Vol. 8, pp. 337-350, 2008.

[14] Liu, Lu, Osama Masfary, and Nick Antonopoulos. "Energy performance assessment of virtualization technologies using small environmental monitoring sensors." *Sensors*, Vol. 12, No. 5, pp. 6610-6628. May 2012.

[15] Borgetto, Damien, et al. "Energy-aware service allocation." *Future Generation Computer Systems,* Vol. 28, No. 5, pp. 769-779, May 2012.

[16] Wu, Chia-Ming, Ruay-Shiung Chang, and Hsin-Yu Chan. "A green energy-efficient scheduling algorithm using the DVFS technique for cloud datacenters.” *Future Generation Computer Systems,* Vol. 7, pp. 141-147, July 2014.

[17] Google “Data Center Locations.” *Data Center Management in the U.S. and China: Trends and Challenges*. 2015. Web. 20 June 2015. http://www.seagate.com/tech-insights/data-center-management-master-ti/

[18] Islam, Saiyedul, et al. "A highly scalable solution of an NP-complete problem using CUDA." *Parallel Computing in Electrical Engineering (PARELEC), 2011 6th International Symposium on*, pp. 93-98, Luton, England, April 2011.

[19] Ahmed, Zakir H. "Genetic algorithm for the traveling salesman problem using sequential constructive crossover operator." *International Journal of Biometrics & Bioinformatics (IJBB)*, Vol. 3, No. 6, pp. 96-105, January 2010.

[20] Matai, Rajesh, Murari Lal Mittal, and Surya Singh. “Traveling salesman problem: An overview of applications, formulations, and solution approaches.” *INTECH Open Access Publisher*, 2010.

[21] Drezner, Zvi. "The Quadratic Assignment Problem." *Location Science. Springer International Publishing*, pp. 345-363, January 2015.

[22] Marco Dorigo and Thomas Stutzle. *Ant Colony Optimization*. 1st ed. A Bradford Book, 2004. Print.

[23] Gambardella, Luca Maria, and Marco Dorigo. "HAS-SOP: Hybrid ant system for the sequential ordering problem." 1997.

[24] Gambardella, Luca Maria, and Marco Dorigo. "An ant colony system hybridized with a new local search for the sequential ordering problem." *INFORMS Journal on Computing*, Vol. 12, No. 3, pp. 237-255, March 2000.

[25] Gambardella, L. M., T. Taillard, and G. Agazzi. "MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows. *Mcgraw-Hill,* pp. 63-76, 1999.

[26] Gambardella, Luca Maria, and Marco Dorigo. "Solving Symmetric and Asymmetric TSPs by Ant Colonies." *International conference on evolutionary computation*, Nagoya, Japan, May 1996.

[27] Gambardella, Luca Maria, E. D. Taillard, and Marco Dorigo. "Ant colonies for the quadratic assignment problem." *Journal of the operational research society*, Vol. 50, No. 2 pp.167-176, February1999.

[28] Özkale, Celal, and Alpaslan Fığlalı. "Evaluation of the multiobjective ant colony algorithm performances on biobjective quadratic assignment problems." *Applied Mathematical Modelling* Vol. 37, No. 14, pp. 7822-7838, August 2013.

[29] Zhang, XiaoLi, et al. "Intelligent fault diagnosis of rotating machinery using support vector machine with ant colony algorithm for synchronous feature selection and parameter optimization." *Neurocomputing*, Vol. 167, pp. 260-279, May 2015.

[30] Saremi, Hamed Qahri, Babak Abedin, and Amirhosein Meimand Kermani." Website structure improvement: Quadratic assignment problem approach and ant colony meta-heuristic technique." *Applied Mathematics and Computation,* Vol. 195, No. 1, pp. 285-298, January 2008.

[31] Tosun, Umut. "A new recombination operator for the genetic algorithm solution of the quadratic assignment problem." *Procedia Computer Science*, Vol. 32, pp. 29-36, 2014.

[32] Tosun, Umut. "Distributed database design using evolutionary algorithms." *Communications and Networks,* Vol. 16, No. 4, pp. 430-435, August 2014.

VITA

Richard Osei

Candidate for the Degree of

Master of Science

Thesis: USING ANT COLONY OPTIMIZATION ON THE QUADRATIC ASSIGNMENT PROBLEM TO ACHIEVE LOW ENERGY COST IN GEO-DISTRIBUTED DATA CENTERS.

Major Field: Computer Science

Biographical: Born in Kumasi, Ghana, lived in USA since 1999.

Education:

Received Bachelor of Arts/Science in Computer Science & Mathematics from Langston University, Langston Oklahoma 2008.

Completed the requirements for the degree of Master of Science with a major in Computer Science at Oklahoma State University, Stillwater, Oklahoma December 2015.

Professional Memberships: Work for Langston University as Learning Management System Coordinator.