ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

ENERGY EFFICIENT RESOURCE MANAGEMENT IN CLOUD DATA CENTERS

Ph.D. THESIS

İlksen ÇAĞLAR

Department of Computer Engineering

Computer Engineering Programme

JULY 2023

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

ENERGY EFFICIENT RESOURCE MANAGEMENT IN CLOUD DATA CENTERS

Ph.D. THESIS

İlksen ÇAĞLAR
(504102501)

Department of Computer Engineering

Computer Engineering Programme

Thesis Advisor: Prof. Dr. Deniz Turgay ALTILAR

JULY 2023

ISTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

BULUT VERİ MERKEZLERİNDE ENERJİ VERİMLİ KAYNAK YÖNETİMİ

DOKTORA TEZİ

İlksen ÇAĞLAR
(504102501)

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Deniz Turgay ALTILAR

TEMMUZ 2023

İlksen Çağlar, a Ph.D. student of İTU Graduate School student ID 504102501, successfully defended the thesis/dissertation entitled "ENERGY EFFICIENT RESOURCE MANAGEMENT IN CLOUD DATACENTERS", which she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Prof. Dr. Deniz Turgay ALTILAR**.....
İstanbul Technical University

Jury Members : **Prof. Dr. Nadia ERDOĞAN** .....
Istanbul Technical University

Dr. İsmail ARI .....
Özyeğin University

Dr. Ayşe YILMAZER .....
Istanbul Technical University

Doç. Dr. Didem UNAT .....
Koç University

Date of Submission : 05 January 2023
Date of Defense : 14 July 2023

To my parents, spouse, son & daughter

FOREWORD

This thesis was written for my Ph.D. degree in Computer Engineering at the Istanbul Technical University. The subject of this thesis is related to Energy Efficient Resource Management in Cloud Data Centers. This is a very fascinating research topic as it has some impacts such as minimizing the damage to nature and reducing the energy consumption.

I would like to express my deepest appreciation to my supervisor Prof. Dr. Deniz Turgay ALTILAR for providing guidence, his invaluable patience and feedback. I also could not have undertaken this journey without my defense committee, who generously provided knowledge and expertise. I am also grateful to TÜBİTAK B3 Lab for their real Cloud environment support, and feedback sessions. Thanks also to my family, especially my parents, spouse and children. Their belief in me has kept my spirits and motivation high during this process.

July 2023

İlksen ÇAĞLAR
(Bilgisayar Mühendisi)

TABLE OF CONTENTS

Page

ABBREVIATIONS

ACF : Auto Correlation Function
ACPI : Advanced Configuration and Power Interface
APR_{total} : Approximated Total Processing Requirements
ARIMA : Auto Regressive Integrated Moving Average
CR : CPU Requirement
DVFS : Dynamic Voltage and Frequency
EC : Energy Consumption
ER : Real Energy Consumption
ET : Execution Time
HDDs : Hard Disk Drives
HW : Holt Winters
IaaS : Infrastructure as a Service
LAA : Look-ahead Energy Efficient VM Allocation
LAA-HW : Look-ahead Energy Efficient VM Allocation – Holt Winters
LAA-O : Look-ahead Energy Efficient VM Allocation – Optimal
LR-MMT : Local Regression-Minimum Migration Time
MAPE : Mean Absolute Percentage Error
MCR : Mean CPU Requirements of Submitted Workloads
MET : Mean Execution Time of Submitted Workloads
MIPS : Million Instruction Per Second
NoHS : Number of Hosts Shutdown
NoRW : Number of Received Workloads
NoSW : Number of Submitted Workloads
Na : Active Number of Servers
NAS : Network Attached Storage
NICs : Network Interface Cards
NoM : Number of Migrations
NPA : Nonpower Aware Algorithm
Nr : Required Number of Processing Unit

PaaS **:** Platform as a Service

PACF **:** Partial Auto Correlation Function

PR_{total} **:** Total Processing Requirement

PUE **:** Power Usage Effectiveness

SaaS **:** Software as a Service

SAN **:** Storage Area Network

SLA **:** Service Level Aggreements

SSDs **:** Solid State Drives

Th **:** Optimal Utilization Threshold

VM **:** Virtual Machines

SYMBOLS

S : Set of servers where S, $S_i \in S$, $i = \{1, \dots, n\}$, $i \in N$

RC[S_i] : Remaining capacity of S_i

UR[W_{new}] : Utilization requirement of W_{new}

TU[S_i] : Total utilization of S_i

$W_{i,j}$: $W_j \in W$ running on S_i $j = \{1, \dots, m\}$, $m \in N$

$VM_{i,k}$: $VM_k \in VM$ on S_i $k = \{1, \dots, z\}$, $z \in N$

max($RT_{i,j}^{W}$) : Maximum remaining time of existing workloads

min($RT_{i,j}^{W}$) : Minimum remaining time of existing workloads

ET[W_{new}] : Execution time of W_{new}

E_OO : Energy consumption for switching the server off and on

E_CSi : Cumulative energy consumption of a server

E_Seri : Energy required for the actual service time

E_CDC : Total energy consumption of the data center

SC_{total} : Total Capacity of a Server

LIST OF TABLES

Page

LIST OF FIGURES

Page

ENERGY EFFICIENT RESOURCE MANAGEMENT IN CLOUD DATACENTERS

SUMMARY

Cloud computing provides services to address the consumers' requirements such as accessing resources for storage, processing and platforms from anywhere based on the service level agreements between the providers and the consumers. Generally clouds give customers three levels of access: Software-as-a-Service, Platform-as-a-Service, and Infrastructure-as-a-Service. Performance improvement is one of the most important goals in order to provide of computing systems. However, by growing the number of resources for offering the users' requirements, the energy consumption of data centers in the cloud is increasingly a key challenge. Therefore, in recent, several researches are investigated a way to increase energy efficiency. Improving energy efficiency of datacenters has received significant attention in the past years. Many existing studies have proposed to run resources at full capacity to increase energy efficiency but, it effects on the system performance. In order to solve performance degradation, researches defines over utilization threshold and when the utilization rate of resource exceeds the threshold, the resource becomes overloading. It causes some problems such as increasing of response time, decreasing of throughput, cache conflicts, context switching. Otherwise, when the utilization rate is lower than the threshold, the resources cannot be used efficiently because of the idle time. However, a single threshold is not enough parameter for decision of allocation. The possible requests in the future are also important as the current requests to the system. Therefore, the costs of both overloading and turning on a new server should be evaluated well in terms of energy efficiency and performance. In addition, under certain condition keeping a server at idle state instead of turning it off can be better to supply incoming requests. Another approach is the server consolidation which reduces the number of active physical machines through collocating Virtual Machines (VM) to a small set of physical machines called VM migration. However, VM migration and server consolidation cause performance degradation as well as energy overheads. Decreasing the number of servers decreases the energy consumption of the system by switching off the idle servers. However, switching on and off servers causes considerable amount of energy consumption in some cases. Therefore, prediction of future workloads becomes an important issue for decision of turning on and off servers to save power. Idle and running time periods of servers and workload execution time should be observed and resource usage pattern should be derived from this knowledge with ignorable error rate. To address these issues, we propose an energy efficient resource allocation approach that integrates Holt Winters forecasting model for optimizing energy consumption while considering performance in a cloud computing environment. The approach is based on adaptive decision mechanism for turning on/off machines and detection of over utilization. By this way, it avoids performance degradation and improves energy efficiency. The proposed model consists of three functional modules, a forecasting module, a workload placement module along with physical and virtual machines, and a monitoring module. The forecasting module

determines the required number of processing unit (Nr) according to the user demand. It evaluates the number of submitted workloads (NoSW), mean execution time of submitted workloads in interval and mean CPU requirements of them to calculate approximately total processing requirement (APRtotal). These three values are forecasted separately via forecasting methodologies namely Holt Winters (HW) and Auto Regressive Integrated Moving Average (ARIMA). The Holt Winters gives significantly better result in term of Mean Absolute Percentage Error (MAPE), since the time series include seasonality and trend. In addition, the interval is short and the long period to be forecasted, the ARIMA is not the right choice. The future demand of processing unit is calculated using these data. Therefore, the forecasting module is based on Holt Winters forecasting methodology with 8.85 error rate. Therefore, the forecasting module is based on the Holt Winters. Workload placement module is responsible for allocation of workloads to suitable VMs and allocation of these VMs to suitable servers. According to the information received from forecasting module, decision about turning a server on and off and placement for incoming workload is making in this module. The monitoring module is responsible for observing system status for 5 min. The consolidation algorithm is based on single threshold whether to decide that the server is over utilized. In other words, if the utilization ratio of CPU exceeds the predefined threshold, it means that the server is over utilized otherwise, the server is under load. If the utilization of server equals the threshold, the server is running at optimal utilization rate. Unlike other studies, overloading detection does not trigger VM migration. Overloading is undesirable since it causes performance degradation but, it can be acceptable under some conditions. To decide allocation of incoming workloads, this threshold is not only and enough parameter. Beside the threshold, the future demands are also considered as important as systems current state. The proposed algorithm also uses different parameters as remaining execution time of a workload, active number of servers (Na), required number of servers (Nr) besides efficient utilization threshold. The system can be instable with two cases; (1) Na is greater than Nr that means there are underutilized servers and it causes energy inefficiency (2) Nr is greater than Na, if new servers are not switched on, it causes over utilized servers and performance degradation. The algorithm is implemented and evaluated in CloudSim which is commonly preferred in the literature since, it provides a fair comparison between the proposed algorithm with previous approaches and it is easy to adapt and implementation. However, workloads come to the system in a static manner and the usage rates of the works vary depending on time. Our algorithms provide dynamically submission. Therefore, to make fair comparison, the benchmark code is modified to meet dynamic requirement by working Google Cluster Data via MongoDB integration. The forecasting module is based on Holt Winters as described before. Therefore, the approach is named Look-ahead Energy Efficient Allocation – Holt Winters (LAA-HW). If we knew the actual values instead of forecasted values, the system would give the result as Look-ahead Energy Efficient Allocation –Optimal (LAA-O). The proposed model uses Na and Nr parameters to decide the system's trend whether the system has active servers than required. If Na is greater than the Nr, incoming workloads are allocated on already active servers. It causes bottleneck for workloads with short execution time and less CPU requirement as the Google Tracelog workloads. The mean cpu requirement of a day and the mean execution time of a day are 3% and 1,13 min 32 respectively. It gives the small Nr value and it causes less number of received workload than Local Regression-Minimum Migration Time (LR-MMT). The number of migration is zero in our approach. The energy consumption for switching on and off in our model is less in comparison with the migration model. As

described in section 3, the energy consumption is sum of the required energy for compute the workload, migration and energy on and off. Each migration consumes 0,05 kwh and each Eoo consumes 0,0019 kwh. LR-MMT has not forecasting module. Therefore, the number of host shut down is greater in LR-MMT than LAA. The proposed model focuses on the open issues of the researches in the literature. The VM migration and the unnecessary switching on and off servers cause extra energy consumption. In addition, completing a workload with the migration overhead consumes more energy since it takes more time. Therefore, the migration can be avoided by optimizing placement of new requests well and to avoid unnecessary switching on and off servers, the proposed approach uses prediction methodology. Holt Winters is preferred as a forecasting technique because of its suitability to the time series. Furthermore, the most of approaches in the literature propose to use resources at optimal utilization rate since over and underutilization cause energy inefficiency. However, it means increasing the number of required resources and it causes more energy consumption. By this motivation, we have proposed an adaptive approach for VM placement without VM migration. The most important contribution is preventing VM migrations while considering remaining time of running workloads. The optimum utilization is a significant factor in order to provide energy efficiency. However, even if its load will exceed the optimum utilization rate, allocation a workload to an already active host instead of allocation the workload to a new server should be preferred under some circumstances. We proposed an adaptive decision making approach in order to energy efficient allocation without migration. We have considered not only history but also future demands and the remaining time of running workloads. In order to see the systems behavior for workloads with longer execution time, sensitivity analysis is done. Real execution times of workloads are extended by multiplying with 10, 100 and 1000 to make the sensitivity analysis. The short execution time means less processing requirement and small value of Nr. It causes bottleneck and less number of received workloads than the Lr-mmt. When the execution times of workloads are increased, the Nr is increased. Thus, the number of received workloads are increased compared to the Lr-mmt. In this manner, the energy consumption is decreased with the proposed algorithms and VM migration overheads are avoided. In addition, the system performance in terms of energy efficiency and throughput is better than Lrmmt for the longer execution time.

BULUT VERİ MERKEZLERİNDE ENERJİ VERİMLİ KAYNAK YÖNETİMİ

ÖZET

Bulut bilişim, müşterilerine servis sağlayıcalar aracılığı ile müşterilerin ihtiyaç duyduğu farklı seviyede hizmetleri, servis seviyesinde anlaşma gereğine uymayı taahhüt ederek sunar. Bulut bilişimle sunulan hizmetler çeşitlilik gösterir. Öyle ki bu hizmetler, depolama veya hesaplama ihtiyacına istinaden fiziksel kaynaklar olabileceği gibi yazılım ya da platform hizmetleri gibi uygulama seviyesinde de olabilir. İnternetin de gelişimi ile birlikte bulut bilişim sayesinde müşteriler, ihtiyaç duydukları hizmetlere, bulundukları yer ve zamandan bağımsız olarak erişebilme kolaylığına sahiptirler. Genellikle bulut, müşterilerine üç seviyede servis erişimi sağlar. Bunlar yazılım, platform ve alt yapıdır. Müşteriler aldıklara hizmetlere bağlı olarak bazı sorumlulukları servis sağlayıcılara bırakırlar. Bu da bulut bilişimin, kullanıcılara bi takım avantajları beraberinde getirdiği anlamına gelir. Örneğin, uygulama seviyesinde hizmet alan kullanıcı, uygulamanın versiyonlarıyla, lisanslamasıyla ve zaman zaman da çıkan yamalarıyla uğraşmak zorunda kalmaz. Bu sorumluluklar hizmeti sunan servis sağlayıcılarına aittir. Ya da dönemsel olarak yoğun hesaplama gücüne ihtiyaç duyan bankalar, bu ihtiyaçlarını hesaplama kaynaklarını satın almak yerine, dönemsel olarak bulut bilişimden kaynak kiralayarak giderdiklerinde maddi olarak kara geçecek ve yılın geri kalanında kullanmayacakları kaynaklar için yatırım ve bakım maliyetinden kurtulacaklardır.

Bulut bilişim, kullanıcılarına, kullandığın kadar öde modeli, servislere kolay erişim ve kaynakların yüksek ölçeklenebilirliği gibi avantajlar sağlar. Tüm bu avantajları nedeniyle, bulut bilişimin sağladığı bu servislere olan ilgi arttıkça veri merkezlerindeki enerji tüketimi de günden güne artmaktadır. Kullanıcılar aldıkları hizmetin kalitesinin düşmesini ya da hizmet bedelinin artmasını istemezken; servis sağlayıcılar da yasalar ve düzenlemelerin de zorlamasıyla enerji tüketimini azaltmaya çalışırken, bir yandan maliyeti arttırmamanın bir yandan müşterilere aynı kalitede hizmet vermenin yollarını aramaya başladılar. Literatürde, son zamanlarda bir çok çalışma enerji verimliliğini arttırmaya yönelik çözümler araştırmaktadır. Enerji verimliliği bir çok seviyede sağlanabilir. Daha enerji verimli veri merkezleri inşa etmek, ağ seviyesinde enerji verimliliğini sağlamak bir yol olabilir. Ya da donanım seviyesinde enerji verimliliğinin sağlanmasına yönelik çalışmalar mevcuttur. Soğutma sistemlerinde enerji verimli çözümlere gitmek ya da veri merkezlerini doğrudan soğutma maliyetini düşürecek lokasyonlara kurmak da enerji verimliliğine yönelik çalışmalardandır. Ancak, bu tez kapsamında veri merkezlerindeki düşük kaynak kullanım oranlarından kaynaklı enerjinin verimsiz kullanılmasına odaklanılıp, bunu çözmek üzerine yöntemler geliştirilmiştir. Literatürdeki çalışmalar gösteriyor ki sunucular, boşta veya düşük yük kaynak ile çalıştırıldıklarında en fazla miktarda enerji tüketiyorlar.

Bu tez kapsamında, enerji verimli kaynak yönetimini amaçlayan adaptif bir yaklaşım ileri sürülmüştür. Bu yaklaşım, enerji verimliliğinin yanında performans kaybını da minimize etmeyi amaçlar. Sunucudaki tüm kaynakların (CPU, RAM, Disk, vb.) enerji

verimliliği açısından optimum kullanımı değerleri vardır. Literatürdeki bazı çalışmalar, enerji verimliliğinin kaynak kullanımını arttırmak ile sağlanabileceğini düşünerek kaynakları maximum seviyede doldurmaya çalışırken, bir takım çalışmalar da optimum kullanım değerini önemseyerek bunu statik bir eşik değeri olarak çalışmalarının merkezine koymuşlardır. Oysaki sistemin sadece mevcut durumu değil olası gelecek ihtiyaçlar da göz önünde bulundurulduğunda, iş yerleştirme kararı için statik eşik değeri kullanmak tek başına yeterli değildir. Sistem, olası gelecek tahmin değerlendirmesine göre küçülmeye gidecekse bu durumda yeni gelen iş yükünü atamak için sırf eldeki aktif sunucular eşik değerini aşacak diye yeni bir sunucu ayağa kaldırmak yerine eldeki kaynakları optimum kullanım değerini aşacak şekilde çalıştırmayı seçebilir. Sistemin küçülme eğiliminde olduğu ve kaynakların yeni gelen iş yükünün atanması ile birlikte eşik değeri aşacağı durumda, iş ataması yapılacak sunucu kararı mevcutta çalışan iş yüklerinin kalan zamanı göz önünde bulundurularak verilir. Burada önemli olan eşik değerin aşıldığı süreyi minimize etmektir. Mevcut çalışan iş yüklerinden kullanım oranı yüksek ve kalan süresi az olan iş yükünün çalıştırıldığı sunucu, yeni gelen iş yükünün atanması için seçilir. Yine sistemin küçülme eğiliminde olduğu ve yeni gelen iş yükünün atanması ile birlikte düşük kullanım oranında kalmaya devam edecek sunucular içinden iş yükü ataması için sunucu belirlenmesi kararında da kalan süre önemli bir parametredir. Sistem küçülme eğilimindeyken, boşa düşen sunucular (üzerindeki mevut işleri bitiren ve yeni bir iş yükü almamış olan) kapatılır. Bu nedenle yeni gelen iş yükünün yürütülme süresi, sunucu üzerinde çalıştırılmakta olan iş yüklerinden kalan süresi en fazla olan ile karşılaştırılır. Eğer iş yükünün yürütülme süresi, en fazla kalan süreden daha küçükse bu durumda, en fazla kalan süresi olan iş yükünün yürütüldüğü sunucu iş yükü ataması için seçilir. Böylece, sunucunun kapatılacağı süre uzatılmamış olur. Sistem, olası gelecek tahmin değerlendirmesine göre genişlemeye gidecekse bu durumda, sunucuları optimum kullanım oranında çalıştırmak birincil hedeftir. Bu yaklaşım, konsolidasyon olarak değerlendirilebilir.

Konsolidasyon, aynı işi daha az sunucu ile tamamlamak olarak düşünülebilir. Enerji tüketimini azaltmayı hedeflerken performans kaybını da minimize etmeye çalışır. Konsolidasyon, iş yüklerinin yerleştirilmesi esnasında olabileceği gibi işlerin yeniden yerleştirilmesi esnasında da uygulanabilir. Yaygın olarak da kaynakların az kullanımını engellemek için sanal makine göç ettirme tekniği önerilir. Böylelikle çok yüklü bir makine üzerinden bir veya birden fazla sanal makine bir başka fiziksel makineye göç ettirilerek optimum kaynak kullanımı sağlanması hedeflenmektedir. Az yüklü bir makine üzerindeki sanal makineler bir başka fiziksel makineye göç ettirilerek az yüklü makinenin boş duruma geçmesi de tercih edilen bir diğer tekniktir. Bu sayede az yüklü makinenin enerji verimsiz çalışmasının önüne geçilir. Boştaki makine de kapatılarak optimum sayıda fiziksel makine kullanımı amaçlanmaktadır. Ancak; bu yaklaşım sanal makine göç ettirme maliyetine sebep olacaktır. Eğer göç ettirilen sanal makine üzerinde koşan iş yükünün kalan çalışma süresi yeterince uzun değilse; bu fiziksel makinenin yük durumunu koruması - az yüklü veya çok yüklü durumda kalması – göç ettirme ile yeni duruma geçmesinden daha enerji verimli olabilir. Bir başka dikkat çekilmesi gereken konu da; makine açma ve kapama kararı sadece mevcut makinenin yük durumuna bakılarak karar verilecek bir karar olmamalıdır. Kullanıcı istekleri göz önüne alınmalı, zamansal örüntülerin farkına varılmalı ve sistemin mevcut durumunun yanında gelecekteki olası istekler değerlendirmelidir.

İleri sürülen LAA ile LAA'nın başarısını göstermek için kıyasladığımız sanal makine göç ettirme tekniğini temel alan LR-MMT, CloudSim kullanılarak gerçekleştirilmiş ve

değerlendirilmiştir. CloudSim literatürde sıklıkla tercih edilen, kapsamlı ve kullanımı kolay bulut simülatörlerindendir. Dinamik iş yüklerine cevap verecek şekilde mevcut göç ettirme paketi de genişletilmiştir. Ayrıca ileri sürülen LAA modeli de gerçekleştirilmiştir. Testler simülatör üzerinde koşulmuş ve sonuçlar elde edilmiştir.

İleri sürülen LAA modelini, Sanal makine göç ettirme yöntemiyle kıyaslamak için gerçek dünya verileri kullanılmıştır. Google'ın clusterlarına ait bir aylık kullanım bilgilerini içeren veriler kullanılmıştır. İş yüklerinin CPU kullanımı, yürütülme süresi ve başlama zamanı bilgileri beş dakikalık aralıklarla toplanmış ve her biri beş dakikalık aralığı gösterecek şekilde veri tabanında ayrı tablolara yerleştirilmiştir.

Sistemin gelecekteki durumunun tahmini için literatürde çok çeşitli zaman serisi tahminleme metodolojileri mevcuttur. Bu tez kapsamında Holt Winters ve ARIMA yöntemleri uygulanmış ve kıyaslanmıştır. Holt Winters seçilen iş yükü zaman serisi için daha düşük hata oranı vermişir. Bu nedenle tahminleme modülü, Holt Winters methodolojisini kullanmaktadır. Sistemde üç tane zaman serisi bulunmaktadır. Bunlardan ilki beş dakikalık aralıkta gelen iş yükleri sayısıdır. İkincisi, beş dakikalık aralıkta gelen iş yüklerinin ortalama yürütülme süresidir. Üçüncü zaman serisi ise beş dakikalık aralıkta gelen iş yüklerinin ortalama CPU ihtiyacıdır. Tüm bu tahmin değerlerinin çarpımı, kaynak ihtiyacını hesaplamada kullanılır. Bir sunucunun optimum CPU kullanım oranı, 70% olarak ele alınmıştır. Toplam kaynak ihtiyacının, bir sunucunun 70% kullanım oranı ile sunduğu kapasiteye bölünmesi ile toplam sunucu ihtiyacı hesaplanır. Mevcut aktif sunucu sayısı ile tahminlenen sunucu sayısı karşılaştırılarak sistemin genişleme ya da küçülme eğiliminde olduğu saptanır. Algoritmalar da sistemin eğilimine göre adaptif karar verir.

Sunucu açma kapama sayısı, simülasyon süresi bittiğinde tamamlanmış olan iş yükleri sayısı, enerji tüketimi ve sanal sunucu göç ettirme sayısı, sonuçları değerlendirirken kullanılan performans ölçütleridir. LAA modeli tahminleme modülüne sahiptir ve sunucu açma kararı da tahminleme modülüne dayalı verilir. O nedenle sunucu açma kapama sayısı LR-MMT yöntemine göre daha düşüktür. Yine LAA modelinde, sanal makine göç ettirme olmadığı için sanal makine göç ettirme sayısı metriği LAA için her zaman sıfırdır. Bir gün içinde gelen iş yüklerinin ortalama CPU ihtiyacı ve ortalama yürütme süresi sırasıyla 3% ve 1.13 dkdır. Bu da tahminlenen sunucu sayısının düşük olmasına sebep olur ki biten iş yükü sayısı da yine LR-MMT'ye kıyasla düşüktür. Kısa yürütme süreli ve düşük CPU ihtiyaçlı iş yükleri, ileri sürülen LAA yaklaşımında dar boğaza sebep olmaktadır. Sistemde tüketilen enerji, açma kapama için gerekli enerji, sanal makine göç ettirme için gerekli enerji ve iş yükünü yürütmek için gerekli enerji miktarlarının toplamı olarak ifade edilmektedir. Tüm bunlar göz önünde bulundurulduğunda, toplam enerji tüketimi de yalnızca bir işi bitirmek için gereken enerji tüketimi de LAA yaklaşımında LR_MMT'ye kıyasla daha iyi sonuç vermektedir. LAA'nın doğruluğundan emin olmak için hassasiyet analizi yapılmıştır. Yürütme süreleri, 10, 100 ve 1000 ile çarpılmış ve deneyler tekrar yürütülmüştür. Böylece daha uzun yürütme süresi ile LAA yaklaşımının tahminlenen sunucu sayısı ihtiyacının küçük değerlerinden kaynaklı çıkan dar boğazı aşılmış ve daha enerji verimli sonuçlar elde edilmiştir.

Bu tez kapsamında sadece CPU tabanlı iş yüklerine odaklanışmıştır. İleri sürülen yaklaşımın diğer sistem kaynaklarını da gözetecek şekilde geliştirilmesi planlanmaktadır.

1. INTRODUCTION

Cloud Computing provides services to consumers through service providers to meet the growing services requirements such as storage and computing needs of commercial or scientific projects. These services offered by service providers to the consumers through data centers are based on the pay per use model. Service providers offer customers basicly three services, Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS), through data centers. A Service Level Aggreements (SLA) is determined between service providers and users and it is guaranteed that service will be provided in accordance with this agreement. Consumers can be individuals as well as companies that do not want to have their own data centers or need extra resources to meet seasonal needs. Based on the services types providers offered, services include some application level functionality such as accessing software, license updates as well as infrastructure, storage or computing functionality such as backup, data storage and big data computing.

Thus, consumers leave the responsibility of owning the resource and its maintenance to the service providers, within the scope of the service received and the SLA. The cloud also allows consumers to use services as per demand. The advantages of cloud computing can be summarized as follows.

- provides a pay-as-you-go model
- resources can be easily added and removed on demand, thus reducing cost
- provides highly scalability
- offers easy access to the resource.
- reduces business risk and maintenance cost.

Thanks to the progress of processing and storage technologies and the success of the internet, cloud computing provides more powerful, cheaper, and easier to access resources. However, it also brings along issues that need to be researched [1].

As the demand for cloud computing services has increased, CO2 emissions and energy consumption have also increased. It is predicted that the total energy consumption and carbon dioxide absorption in the world will increase by 48% and 34% between 2010 and 2040 [2].

Infrastructure providers also try to reduce the cost of energy in data centers due to both laws and regulations and standards. On the other hand, they aim to reduce the cost of the service they provide and increase the profit rate. Users, on the other hand, want to have the same service at an acceptable quality with less cost. For this reason, it is necessary to consider the performance of the service offered to the users while aiming to reduce energy consumption and reduce costs. In recent years, the issue of energy efficiency in data centers has gained importance for all these reasons. Energy efficiency in Cloud Computing can be achieved at many levels.

It is estimated that 43% of total energy consumption in data centers is belong to powering and cooling. It may be possible to develop and prefer energy-efficient equipment, to go for energy-efficient solutions at the cooling level, or to achieve energy efficiency with new cloud architectures. In addition, the low usage rates of the machines in the data centers and the inefficient use of the machines cause the energy consumption to increase, thus increasing the cost and decreasing the performance. Assigning resources by requirement is one way to reduce cost. With this; Thanks to virtualization, cost can be reduced by using the Virtual Machine Migration technique, by migrating the workloads on the low-loaded machines to a smaller number of less-loaded machines together with their situation, and by shutting down or sleeping the idle machines on which the workloads have been migrated.

In this thesis, we proposed an energy efficient job placement approach. So far, we have examined and categorized the studies in the literature in detail. We analyzed the strengths and weaknesses of existing studies. In some researches, optimum utilization rate has been proposed to achieve energy saving and performance according to the types of resources where workloads are placed. Filling the resources up to optimal utilization rates instead of using 100% enables them to work more energy efficient. However; it is only to make a decision to place a job according to the resource utilization rate and to focus on the current state of the system. By looking at the pattern, seasonality and trend of the workload and considering the possible future situation, a decision can be made to use resources more efficiently. Also, taking into account the

general trend of the system, unnecessary machine on/off decision is avoided. Virtual machine migration also causes performance degradation and extra energy consumption. In our study, the virtual machine migration method was not used. An adaptive decision mechanism work placement approach, which minimizes the number of machines used and converges the utilization rate to the optimal level, is presented.

The contributions made within the scope of this thesis can be summarized as follows;

An energy efficient resource allocation algorithm (Look-ahead Energy Efficient VM Allocation - LAA) has been proposed. Incoming requests are placed on the resources based on the adaptive decision mechanism.

The proposed algorithm also takes into account performance loss while trying to minimize energy consumption. Unlike other approaches, the algorithm takes into account not only the current state of the system, but also possible future requests. Holt Winters was used as the estimation method.

Real-world workload was used to evaluate the algorithm. Google Cluster data was used and the implemented algorithm was run on a simulator named CloudSim.

1.1 Problem Definition

The amount of energy used in data centers has increased with the increase in the interest in data center services, the increase in the number of customers and software that need more computing or storage. According to [3], between 2010 and 2018, total energy consumption in data centers increased by 6% from 194 TWh to 205 TWh. While the need for data storage has increased 25 times, the increase in energy consumption has been limited to 3 times thanks to hardware improvements. On the other hand, the increase in IP traffic increased 10 times, while energy use increased marginally. In data centers, the number of instances running computational workloads increased more than 6 times. According to [4], amount of computational workloads in data centers increased by 550%. While the electricity need of global data centers is 1% of the global electricity need, 30% of the data center servers are located in the USA. 1.8% of electricity consumption in America is used by data centers.

Data centers provide the services they offer to their users with different types of devices. While providing the storage or stored data needs of the users with storage devices, they use network devices for incoming and outgoing data flow and internet

needs. Server needs are required for data processing. The energy used by all these devices is released as temperature, and the cooling systems are operated to guarantee the healthy operation of the devices at a certain temperature. All these devices and systems cause energy consumption at different rates. [5] the energy consumption rate belong the fundamental components of data centers are summarized in Table 1.1.

Table 1.1 : Fraction of U.S. data center electricity use.

Components	**Energy Consumption Rate**
Cooling and power provisioning systems	43
Servers	43
Storage	11
Network	3

In addition to causing an increase in electricity consumption, it also has negative effects on global warming by causing an increase in the CO2 footprint. The 0.5% value of America's Greenhouse Gas Emission is caused by data centers in America [4].

There are studies in the literature to reduce the energy consumed by servers and to use resources more efficiently. While trying to reduce the energy consumed by the computing units, the quality of service promised to users should also be considered. To handle energy inefficiency, common resource provisioning and running servers at an optimal utilization rate through VM migration have been proposed to reduce energy consumption on cloud infrastructures. However, running at the optimal utilization rate may require turning a new server on to meet the requirement of incoming workloads, and it consumes more energy than is consumed due to the performance degradation from allocating the incoming workload on already active servers that are at optimal utilization rates. Moreover, VM migration also consumes energy and causes execution delays since time is needed for VM migration.

1.2 Motivation

The motivation of this thesis is to reduce the cost of Cloud Computing services and to minimize their negative effects on the environment. Data centers account for 1% of the world's electricity consumption. With the increase in computational workloads running in data centers, the increase in IP traffic and the increase in the need for storage, it is predicted that the increase in energy consumption and the energy gain from energy efficiency studies will not be at the same rate. According to [6], even if

the growth factor is taken the same, data centers that needed 292TWh of energy in 2016 will need 353 TWh of energy by 2030. According to their study, with the increase in the growth factor, they predict that the energy need will be 1287 TWh in 2030.

In this study, green cloud computing is aimed and the negative effects of cloud computing on the environment are tried to be minimized by taking into account the performance loss. The relevant studies in the literature have been extensively researched and analyzed and an energy efficient workload allocation approach has been proposed. Due to its effects on global warming and the consumption of energy resources, it seems inevitable to reduce the amount of resources used. Resource and energy efficiency should be ensured at the allocation of workloads on servers while considering the performance loss. It is important to minimize the negative impact on the environment by using the optimal number of machines, avoiding unnecessary server startup costs or the additional burden of migrating virtual machines.

1.3 Research Question

Consolidation is a commonly used technique during allocation and migration for saving energy by using minimum number of resources and increasing resource utilization rate. Consolidation can be achieved during both allocation and migration by using heuristic solutions. The objective is to use fewer resources to run the same workload at low cost and energy efficiently through consolidation. However, when the migration is preferred for consolidation, it causes low throughput and extra energy consumption since it is a time- and resource-consuming process. We broadly explored the literature and propose an Energy-efficient VM allocation approach. The aim of this research is to answer the following questions.

1. Is it possible to provide an energy efficient workload allocation model that will also consider performance and cost?
2. Is it possible to provide energy efficiency with only VM allocation technique with avoiding VM migration?
3. Is it enough to use only one static threshold whether the host is available of not to run the incoming workload?
4. When the switch on/off decision is based on system's possible future demand, it is possible to save more energy than the method based on only current state?

5. Which prediction methodology is the most suitable for the choosen workload type?

I focus on an adaptive approach of VM placement without VM migration.In particular, the following issues are explored and examined in this research:

- Energy efficient VM Allocation: Datacenters consists of homogenous servers. Power consumption ratio is defined with spec for the related server. Each workload request is allocated on a VM. The VM types are Amazon EC2 instance types. Initially, the most appropriate VM is allocated for the incoming workload with the resource requirements defined by the VM types. Then, VMs utilize less resources according to workloads [7]. The most appropriate server selection is based on not only the optimal utilization threshold but also existing workloads' remaining times which are running on the same server. Moreover, the predicted future demand is also consider while allocation process. (RQ1)
- VM migration: VM migration causes performance degradation and additional cost. In addition, completing a workload with the migration overhead consumes more energy since it takes more time. Therefore, to avoid migration, the allocation approach aims optimizing the placement of new requests through using prediction methodology. (RQ2)
- CPU is the component with the highest proportion in terms of power consumption of a physical host [8]. Moreover, the running machines at idle states or low utilized cause energy inefficiency. Power consumption of CPU in idle state is more than 50% of the fully loaded state [9-12]. According to [13], optimal CPU utilization is set as 70% and if the usage exceeds the optimal value, it means the server is overloaded. However, it is not the only parameter used to decide whether to turn a new server on to host the newly arrived workload. The trend of the system is as important as the threshold during the allocation decision process. If the number of already active servers meets the forecasted future requirement and the current overutilization situation occurs temporarily, then the proposed algorithm makes the allocation decision for the newly arrived workload by considering the remaining time for already running workloads on active servers. (RQ3, RQ4)
- When the server is idle state, the common approach is switching of this server to save energy. However, beside that the current state of the server, future

demand also is considered. If the system trend is enlarging, it means the number of existing active servers are not enough to meet incoming requests and additional servers are required. Because of additional active servers requirement, switching off the idle server is an unnecessary process to cause energy overhead and time consuming. (RQ4)

- Prediction Methodology: Suitable prediction methodology can be differ for workload types. Training data set definition, parameters and method selection are important steps of forecasting. Right monitoring interval and prediction methodology give more accurate result in terms of minimum error rate. (RQ5)

1.4 Objectives

The aim of this thesis is to propose an energy efficient resource allocation while considering the minimization of performance degradation. The model deals with the optimum number of servers relying on the adaptive optimal utilization of hosts with respect to the consumed energy to meet dynamic workload requirements based on prediction. In order to achieve the aim, resource management approaches in the literature are investigated in detail and research questions are explored and the following objectives are defined.

- To determine the place of the approach put forward in the thesis in the literature by examining and categorizing the literature in detail. (RQ1)
- Comparing the results of our study with one of the most effective VM migration methods (RQ2)
- Developing an adaptive decision mechanism by including the prediction method, taking into account the possible future state of the system, as well as the instantaneous state of the system. (RQ3, RQ4)
- Prediction methodologies are compared in terms of predicting the Google Cluster tracelog data with Mean Absolute Percentage Error. (RQ5)

1.5 Research Methodology

Following research methodologies are followed to achive objectives:

Resource management techniques are investigated in detail and categorized to deeply understand KPIs, contributions and challenges. In this manner, the literature survey gives a broad perspective to show open points and improvement areas. (Objective 1)

Real world trace data is used to evaluate the algorithm in terms of energy efficiency and performance. The proposed approach is based on dynamic workload submission. Therefore, the existing code of simulation is extended to meet dynamic workload requirements and make a fair comparison with the proposed algorithm. The algorithm is implemented and run on CloudSim [14], which is a commonly used Cloud simulator. (Objective 2)

Prediction Methodologies: In order to avoid migration by optimizing the placement of new requests well and to avoid unnecessarily switching servers on/off, the proposed approach uses prediction methodology. After data analysis and comparison with another time series analysis methodology called ARIMA, HW gave the better result in terms of the minimum error rate. (Objective 3, 4)

1.6 Research Hypothesis and Contributions

The amount of energy used in data centers has increased with the increase in the interest in data center services, the increase in the number of customers and software that need more computing or storage. Servers are the components of datacenter which have main portion of electricity usage in datacenter. Therefore, many researches focus on decreasing the number of active servers for running the same workload to save the energy. To minimize the number of active servers, workloads are consolidated into a subset of active servers. Consolidation can be applied in two phases; workload placement and scheduling. During the placement phase, maximizing resource usage is aimed. When a new workload is arrived, target resource for allocation is seeked among already active servers. If remaining capacity of one of the active servers is enough to allocate the workload on it, there is no need to switch a new server on. In this way, it is avoided increasing the number of active servers at placement phase. In a simply way, consolidation is achieved at placement phase by maximizing usage of already active resources. Scheduling is the other phase to be able to apply consolidation.

Contraty to the workload placement phase, workloads have already been allocated on servers in the scheduling phase and consolidation is achieved through migration of already allocated workloads to fewer servers. After the consolidation, some servers become idle since the workloads are migrated from them. To save energy, servers at idle state are switched off. However, both unnecessary switching idle servers off and unnecessary migration cause extra energy consumption compared to initial state of the system. Migration is a time and resource consuming process. Switching servers off and on is also time consuming process and extra energy is consumed to switch them on again to be ready for newly arrived workloads. It has been observed that an adaptive approach is needed to solve this problem instead of a static approach, since too many parameters have to be taken into account.

1.6.1 Hypothesis

VM migration and the unnecessarily switching of servers on/off cause additional energy consumption. In addition, completing a workload with the migration overhead consumes more energy since it takes more time. When the predicted future demand is considered beside that the current state of the system, the optimized resource management could be achieved. Thus, unnecessary switching on/off processes could be avoided in such a way that the system's trend is analyzed. Moreover, the remaining time of existing workloads on a server is also another important parameter for allocating new incoming workload on that server. In addition to using constant optimal utilization threshold during allocation decision, the systems trend and the remaining time parameters help to decide more efficient way compared to the single threshold decision.

1.6.2 Contributions

The key contributions of this research are as follows:

- A taxonomy and detail analysis of existing papers in the literature about energy efficienct resource management in cloud (RQ1)
- We propose an energy efficient resource allocation algorithm called Look-ahead Energy Efficient Resource Allocation (LAA). LAA facilitates adaptive allocation of the incoming user requests to computing resources. A single threshold is used for CPU overutilization detection, but it is not the only parameter used to decide whether to turn a new server on to host the newly

arrived workload. The trend of the system is as important as the threshold during the allocation decision process. If the number of already active servers meets the forecasted future requirement and the current overutilization situation occurs temporarily, then the proposed algorithm makes the allocation decision for the newly arrived workload by considering the remaining time for already running workloads on active servers. (RQ2, RQ3, RQ4)

- The proposed algorithm minimizes energy consumption while preventing performance degradation. The algorithm is based on not only the current state of the system but also future demand as determined through Holt Winters forecasting to make adaptive decisions. Google has published the trace data of their clusters. These data are used to evaluate the performance of the proposed algorithm. After data analysis and comparison with another time series analysis methodology called ARIMA, HW gave the better result in terms of the minimum error rate. (RQ5)

2. LITERATURE REVIEW

Energy efficiency in cloud computing is a very broad topic. Energy efficiency can be achieved at any level of the cloud. Energy efficiency can be achieved at the network level used in the data center or at the hardware level by building more energy efficient data centers. You can use energy efficient CPU, RAM, or you can design the system with smaller data centers, reducing maintenance, repair, installation costs and designing energy efficient way. Energy efficient solutions in cooling systems can be chosen or data centers can be set up in locations that will reduce cooling costs. However, energy inefficiency due to low resource usage in data centers has been focused on this thesis. Running the CPU at 10% usage rate causes more than 50% power consumption [15]. Servers consume the highest amount of energy when they are running idle or with a light load. Another remarkable fact is the total power consumed in data centers is much higher than the power consumed by infrastructure components such as cooling systems and power distribution. Efficient resource usage has a significant effect on reducing power consumption and thus energy consumption. The relationship between the total power consumption and the power consumption of IT components is a good indicator of the energy efficiency of data centers. Green Grid – a global consortium of various members from the IT industry for the energy efficiency of data centers, has defined power usage effectiveness (PUE) to measure this relationship. The ratio of the data center power consumed in total to the power consumed by the IT equipment gives the PUE value. It is recommended that this value be at levels 1.4. [15]. Within the scope of this thesis, it is aimed to reduce the energy consumed by servers to run applications Although studies aiming at energy efficiency by reducing the power consumed by IT equipment are mentioned, studies on resource management, efficient use of resources, workloads resource compatibility are examined, and an adaptive work placement algorithm is proposed. In this way, it is aimed to allocate workloads on resources in an energy-efficient way by using resources efficiently and considering performance. In the rest of this section, different approaches to achieve energy efficiency are explained. These approaches are presented in the scope of the taxonomy shown in Figure 2.1

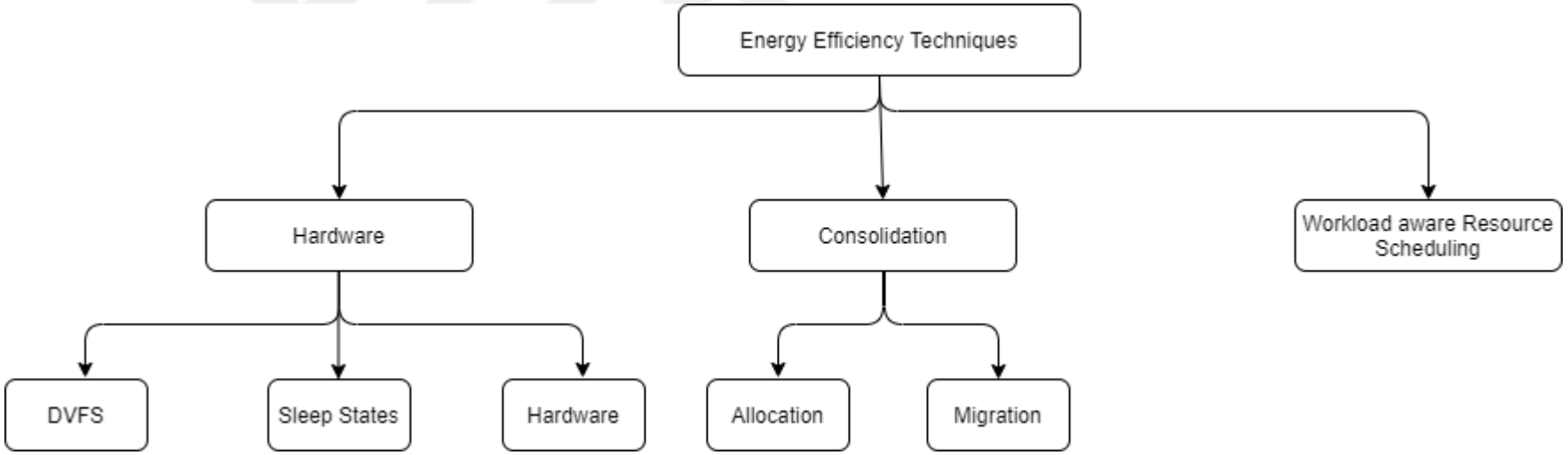

Figure 2.1 : Taxonomy of energy efficient techniques in computing systems.

2.1 Hardware

To save energy, a compute node/server has many components whose power consumption can be optimized. Energy savings can be achieved by turning off the hard disk, network card, CPU, motherboard, RAM, fan and PCI slot components or switching them to lower power consumption modes. Table 2.1 shows the energy consumption amounts of the components of a typical server [16]. Power management of devices and operating system interfaces is defined in the Advanced Configuration and Power Interface (ACPI) specs. Server-level power savings can be achieved primarily at the processor level. There are several ways to control power usage at the processor level; server idle power states (C-states) which control static energy consumption, and different CPU frequencies (P-states) which use dynamic voltage and frequency technique technology (DVFS), and CPU throttling (T-states). The method that uses the ability to turn off components that are not in use to conserve power is known as C-states. The higher-numbered C state indicates deeper sleep, saving more power and taking longer to get the component up. (e.g. the CPU is fully powered on at C0 state while CPU internal clocks are completely stopped at C3 state) [17]. P-states refer to process states. It refers to the different values of the frequency and voltage of the CPU. Higher-numbered P-state means slower processor speed and therefore less power consumption; lower-numbered P-state means that the processor performs better while consuming a higher amount of power. T-states refer to the throttling of the processor. It can be expressed as throttling the processor clock to lower frequency values to reduce the thermal effect. It means that the processor is forced to idle for a certain percentage of each second's cycle. It is only aimed to reduce the thermal effect without throttling voltage. It does not reduce power consumption, it causes more energy consumption as it prolongs the process.

Table 2.1 : Power consumption breakdown of components of a typical server.

Component	Peak Power	Count	Total Power	Percentage
CPU	40W	2	80W	37.0
RAM	9W	4	36W	16.9
Disk	12W	1	12W	5.6
PCI Slots	25W	2	50W	23.5
Mainboard	25W	1	25W	11.7
Fan	10W	1	10W	4.7
Total			213W	

The CPUfreq engine is used to dynamically change the processor frequency at runtime. In this way, the processor is operated at different frequencies and voltages, thus gaining power. The pre-configuration of power schemes of CPUs are referred to as kernel governors. Performance governor, Powersave governor, On-demand governor and Conservative governor are the governors provided by the CPUfreq infrastructure. The performance governor sets the CPU frequency to the highest possible performance and does not focus on power gain. On the other hand, the powersave governor sets its frequency to the lowest value without considering the performance of the processor, as the opposite of the performance governor, its focus is power gain. The on-demand governor checks the CPU usage rate every second and determines the system clock frequency and voltage according to the CPU usage rate information. It defines the optimum CPU utilization rate as a threshold and changes the CPU frequency so that the CPU is always running at the optimum utilization rate. Conservative governor is similar to On-demand governor but waits for the next cycle instead of performing the frequency change immediately [18]. The DVFS technique can be used when the start of a task depends on the completion of the other two predecessors, and when two parallel and independent precursor tasks have different end times. Since the next job cannot start before the two tasks are finished, the ending times of the predecessor tasks are changed so that the frequency is the same. Energy efficient job placement algorithms using DVFS are static algorithms where tasks and resources are tightly coupled [19, 20]. Storage area network (SAN) and network-attached storage (NAS) are network storage systems that do not cause energy problems to a certain extent. In addition, Solid State Drives (SSDs) are more energy efficient than traditional storage systems such as hard disk drives (HDDs). There are also studies on reducing energy consumption in Network Interface Cards (NICs) and Hard Disks components in the literature[18]. Many companies reduce the power used for cooling by installing their data centers in regions and countries with cold climates or using seawater, such as Google's data center in Hamina, Finland. This technique, called the free cooling technique, aims to reduce the cooling power consumption, which constitutes about 33% of the power consumed in data centers, by using renewable energy or reduce to zero[21].

2.2 Consolidation

Consolidation is a commonly used technique during allocation and migration for saving energy by using minimum number of resources and increasing resource utilization rate. Consolidation can be achieved during both allocation and migration by using heuristic solutions. Allocation and placement step of migration are handled as a Bin Packing optimization which places items into the minimum number of PMs using heuristic techniques which are enough fast for large-scale system but not guaranteeing optimal results [22]. Bin packing approach is applied in different ways in the literature as shown in Table 2.2. Finding suitable host for both incoming workload and already running workload to be replaced is the basic step of the bin packing approach. Studies differ from each other according to methods used for finding suitable host such as First Fit, Worst Fit and incoming request types such as VM, task or container [23]. Other differentiating point of studies is meaning of bin capacity as optimal resource utilization rate of a resource or fully loading.

Table 2.2 : Bin packing application differences.

Terminologies	Aspects
Problem	VM Allocation
	VM Placement
Bin Type	CPU
	Multi Dimensional
Bin Size	Full Capacity
	Optimal Utilization Rate
Item	Task
	VM
	Container
Method	Heuristic
	Meta-heuristic

Running resources at full capacity affects performance negatively. To solve the performance problem, some studies suggest defining a threshold value that determines the optimum static resource utilization rate [23, 24]. Threshold usage differences of VM allocation and migration source/target host detection are shown in Table 2.3.

Table 2.3 : Threshold usage differences of energy efficiency techniques.

Technique	Threshold		Papers
Allocation	Up to fully loaded	Single resource	[25], [11], [12]
		Multiple resources	[26]
	Up to optimal utilization rate	Single resource	[10]
		Multiple resources	[27], [28]
Migration	Single threshold	Single resource	[9]
	Multiple thresholds	Single resource	[29], [30], [31]

Studies are also divided into two categories in terms of bin. In some studies, CPU is considered as a bin since [8] shows that CPU is the component with the highest proportion in terms of power consumption of a physical host. Moreover, the running machines at idle states or low utilized cause energy inefficiency. Power consumption of CPU in idle state is more than 50% of the fully loaded state [9-12]. However, [25, 26, 28] propose multi-dimensional bin packing solutions because CPU is a sufficient parameter alone for only CPU-bounded workload and based on the workload requirements, other resource types also should be considered.

2.2.1 Allocation

VM allocation is the process of provisioning physical resources for a task or a VM. In order to reduce the energy consumption and optimize the resource utilization in data center, there are several studies in the literature which focus on improving allocation .

[32] aims to find node workload pair in terms of the most energy efficient way. When the server is not needed any more, it is sended to sleep state to save energy. The approach provides different power management policies which define different amount of time before switching the node sleep state. Moreover, the approach provides different computer selection policies , cluster termination policies and management policies. These policies are combined as required and provides the most energy efficient way for allocation, termination and management . The technique only considers the current state of a system. [24] is also another approach which aims optimizing energy consumption during task consolidation. The idea is underlined that the relationship between the CPU utilization and energy consumption is not linear. Different levels of utilizations are presented according to the energy consumption . According to the optimal level of utilization, servers are tried to be filled with tasks. When the proposed algorithm Energy Aware Task Consolidation (ETC) is compared with MaxUtil which aims to maximize resources, ETC provides 17% improvement over MaxUtil. [33] focuses on minimizing the number of servers in order to reduce energy consumption. It aims to find the most efficient server for the task. They show that the number of servers required to achieve a minimum task waiting time is bounded. [28] is an another study that underlines the relationship of utilization and energy consumption . At low utilization , idle power is not amortized efficiently. On the other hand, at high utilization, it causes performance degradation. The paper provides a solution based on Euclidian Distance to find the distance between optimal point and the actual utilization level. [25] takes into account different types of resources; memory and disk in addition to CPU. The study focuses on resource leak while aiming of reducing power consumption.The problem is obtained as a multi dimensional bin packing problem and it aims to find optimal solution in terms of not only CPU utilization rate but also other resources. [10] proposes Power Aware Load Balancing Algorithm to reduce power consumption. The algorithm is based on optimal utilization rate of combined resources; CPU and RAM. There are three states; balance, upscale and downscale. In the balance mode, if all machines are active and utilization rate belong all machines exceed 75%, then the most underutilized server among them is selected for allocation. If there is any machine with utilization rate under the threshold, the most utilized server is selected. In the upscale mode, there are inactive servers. If all active servers exceed the threshold, in that case a new server is turned on. In the downscale mode, idle VMs and servers are detected and turned off to save

power. [12] is another research focusing on energy while resource provisioning based on prediction of future demand and the number of required PMs. It is similar with the previous work in that case the threshold is a combination metric of multi dimensional resources. [27] consists of prediction module based on Holt Winters. They underline that frequently switching off and on servers consume more energy than keeping idle host in the standby mode since switching on/off hosts take almost 3 minutes and also cause the delay of task execution. They focus on the importance of prediction methodology and constant predicted required number of servers has lower energy consumption than on demand service approach which shuts down hosts immediately. [11] is another approach which considers the relationship between energy consumption and resource utiliziation is highly coupled. Therefore, the paper aims maximizing resource utilization to reduce energy consumption. They have also considered that the impact of parallel running time on energy consumption. During cost function calculation in their algorithm, energy consumptions of running alone and running in parallel are calculated separately. Overlapping tasks execution times is preferred to reduce energy consumption.

Table 2.4 : Summary of some allocation approaches in cloud computing.

Paper	Aim	Technique	KPI	Workload	Environment	Advantages	Open Points
[32]	Reduction of power consumption for any computer system.	Presenting a number of policies; machine level, cluster level and virtualized environment. sleep state	QoS, Throughput, PUE	Combination of real data and synthetic	Simulation	Different level policies are not mutually exclusive and combination of these policies, the overall energy saving can be inreased	No prediction technique
[24]	Minimizing energy consumption	BF, restricting CPU usage below a threshold, consolidation	Network lantency	Low loading medium loading high loading	Simulation low resources – 5 nodes medium resources – 10 nodes high resources – 15 nodes	Different levels of energy consumption util rates are divided into 6 levels.	No sleep or shut down mode. Only current state is taken into account during allocation.

Table 2.4 (continued): Summary of some allocation approaches in cloud computing.

Paper	Aim	Technique	KPI	Workload	Environment	Advantages	Open Points
[33]	Minimize energy consumption	The most efficient server first greedy task scheduling algorithm.	Average task waiting time. Assumption: enough bandwidth to avoid delays in transmission of data.	Two types or tasks: immediately assigned tasks and backlog tasks. exponentially distributed task arrivals, homogeneous tasks	Matlab simulation	The number of servers required to achieve a minimum task waiting time is bounded. Determine the largest number of servers that can be turned on for a given workload	Algorithm time complexity is not considered. Energy comsumption is evaluated for each server for each task.
[28]	Finding optimal points while energy performance trade-offs	Problem is modeled as a modified bin packing consolidation – eucladian distance based	Performance degradation because of internal conflicts such as cache, CPU, disk scheduling	CPU bounded Disk bounded 4 mixes of applications	Real environment consists of 4 physical servers.	2 different types of resources are considered. Optimal utilization rate is proposed and performance degradation is considered.	Calculation complexity is high. It can be ignorable for 4 machines but increasing number of servers, increasing time complexity.

Table 2.4 (continued): Summary of some allocation approaches in cloud computing.

Paper	Aim	Technique	KPI	Workload	Environment	Advantages	Open Points
[25]	Reducing power consumption while minimizing the number of PMs and avoiding resource leak.	Two alogorithms: greedy (fully loaded, standby) balanced (resource limited, fully loaded, resource leak, standby)	Resource leak three types of resources; cpu, memory and disk.	4 different datasets with different sample size – uniform distribution	Simulation	Focus on continuous VM placement avoiding resource leak	Only current status is considered
[10]	Reducing the total power consumed by local cloud.	Power aware load balancing algorithm with three states; balance, upscale, downscale.	Optimal utilization rate CPU and RAM utilizations are considered	Synthetic workload	Simulator	Static utilization is stretched based on the systems state.	Some decisions are not clear such as composed utilization concept, 75% util. threshold, predefined m value for total number of PCs and 6 hours for the maximum time period of idle state to decide shutting VM down .

Table 2.4 (continued): Summary of some allocation approaches in cloud computing.

Paper	Aim	Technique	KPI	Workload	Environment	Advantages	Open Points
[12]	Energy aware resource provisioning framework to reduce energy consumption	Modified BFD – estimating the number of needed PMs during the next prediction window. data clustering – k means traces decomposer	Defining prediction window, the energy consumed by transition for sleep state is considered.	Google tracelog	Real environment	Prediction of future VM requests with the amount of CPU and memory. prediction of the number of PMs that the cloud needs in the future.	Maximum utilization is aimed and performance degradation is not considered

Table 2.4 (continued): Summary of some allocation approaches in cloud computing.

Paper	Aim	Technique	KPI	Workload	Environment	Advantages	Open Points
[27]	Power saving approach in cloud computing environment.	Modified knapsack algorithm: forecasting demands of next period with HoltWinter	Cost of switching on/off hosts	Real world data trace – three types of resource demans	Cloudsim	Avoiding frequently switching on/off hosts. Underlines that constant number of servers consume less energy than ondemand service shut down.	Resource allocation is based on number of cores, amount of memory. There is no optimal utilization rate. Resource wastage is not considered.
[11]	Maximizing resource utilization without performance degradation.	ECTC, maxUtil	Remaining time of workloads	Synthetic workload Service requests arrive in a poisson process and the requested processing time – exponential distributed.	Homogeneous resources	During energy consumption calculation, time period of workloads running in parallel and alone are considered.	Optimal utilization rate is not considered and if forecasting methodology is used, switching on unnecessary server can be avoided.

2.2.2 Migration

Server consolidation through migrating workloads into a small set of servers and switching idle servers off is a commonly used technique to save energy. Migration is a process of replacement VMs from the source hosts to target hosts to reduce the number of active servers via following 4 fundamental steps: (i) threshold definition, (ii) source host detection, (iii) VM selection and (iv) VM placement. Threshold is used to classify hosts as source and target. Threshold value can be set for single resource or multiple resources. It depends on the problem formulation defined as above (i.e. unidimensional and multidimensional). Threshold value can be set before the runtime statically or at the runtime dynamically. According to [13], optimal CPU utilization is set as 70% and if the usage exceeds the optimal value, it means the server is overloaded. Therefore, some selected VMs should be migrated out to reduce performance degradation and energy consumption. [7, 29, 34] propose auto-adjustment of threshold by using statistical analysis of historical data such as Median Absolute Deviation, Interquartile Range, Local Regression belong VMs because of the motivation which fixed values of thresholds are unstable for unpredictable workloads. Underloading is also undesired state in terms of energy efficiency. Through migration all VMs from underutilized server to target hosts, underutilized server's state is changed to idle, and it can be switched off to save energy. In [30], K-means clustering algorithm is used to define three threshold T_a, T_b, T_c values to classify hosts as little loaded, less loaded, normally loaded and overloaded. Underutilization is divided into two sub states as little loaded and less loaded. Besides overloaded hosts, little loaded hosts also are candidates of source hosts while target hosts are selected from less loaded hosts during VM placement step [31] uses power performance ratio called gear. Tere are four types of gears as preferred, best, underutilized and overutilized and 11 gears value corresponding to 11 distinct utilization rates from 0 to 100%. If the gear which the server is working at is higher than the preferred gears, it means the server is overutilized. Otherwise, it is underutilized. Unlike the previous studies, based on the characteristics of the computing node (i.e., 4 diferent computing nodes Fujitsu, Inspur, Dell, and IBM are evaluated), there may not be a utilization rate that means overloading. Treshold usage diferences of VM allocation and migration source/target host detection are shown in Table 2.3. The objective is to use fewer resources to run the same workload at low cost and energy efficiently through consolidation. However,

when the migration is preferred for consolidation, it causes low throughput and extra energy consumption since it is a time- and resource-consuming process. [35] aims finding best placement via a central decision module called Arbitrator which receives the related information belong servers from three managers; performance, power and migration managers while considering migration cost. They proposes a detail information about the impact of migration time manner. [36] monitors the system's state in 20 minutes interval to decide source and destination servers according to two thresholds for migrating VMs from source to destination. The approach is based on prediction methodology to avoid unnecessary switching on/off servers. [13] balances the system load in different levels; memory, I/O and compute.They propose the idea that allocated workloads with heterogeneous characteristics on a PM is more efficient than workloads with homogeneous characteristics. Unbalance resource utilization causes high power consumtion and performance degradation. For example, unbalance of memory utilization causes cache conflicts. Migration decision is taken based on predefined thresholds for each resource type. [37] emphasizes that frequently migration causes network vibration. Moreover, the impact of migration on energy consumption is taken into account in terms of two aspects; energy consumption during migration transition period and energy consumption during VM migration further.

Table 2.5 : Summary of some migration approaches in cloud computing.

Paper	Aim	Technique	KPI	Workload	Environment	Advantages	Open Points
[13]	Energy efficiency and performance	4 steps: MPC balance, IPC balance, util. balance, DVFS. Thresholds are defined for each resource type.	Memory, I/O, CPU resource types, performance, cache conflict	Spec CPU 2000 suite eon (low memory, high IPC) and mcf (high memory, low IPC)	Xen client server model (vgnode, vgserv)	The idea is proposed that allocated workloads with heterogeneous characteristics on a PM is more energy efficient than workloads with homogeneous characteristics	Migration cost and remainin time of migrated task are not considered.
[37]	Energy efficient scheduling of VM reservations in cloud data center.	Finding optimal solution with the minimum number of job migrations LLIF, turn off or put into sleep mode.	Network vibration result of migration	Synthetically generated data, poisson arrival process, log data of parallel workloads archive (PWA)applications	Real AWS EC2 instances and VMs	VM migration cost and further results are considered. Job execution time is also considered.	The selected workload is CPU intensive but there is no optimal utilization rate belong that resource. It causes performance degradation and energy inefficiency.

Table 2.5 (continued): Summary of some migration approaches in cloud computing.

Paper	Aim	Technique	KPI	Workload	Environment	Advantages	Open Points
[35]	Power minimization while considering performance constraints	Framework consists of three managers; performance, power and migration. Arbitrator is responsible of gathering related information from these managers and deciding VM sizing, server throttling and live migration.	Performance, SLA	Two HPC apps daxpy and fma	Simulation	This is entire solution which provides different levels of power saving methodologies	No prediction technique.
[36]	Minimizing energy consumption while considering QoS	PreAntPolicy consists of monitoring model, prediction model based on fractal mathematics, allocation model and scheduler	CPU and disk usage, SLA	Google Tracelog	CloudSim	Turning on/off decision is based on prediction. Heterogeneous of tasks are considered.	Monitoring window is defined as 20 mins. The reason behind that is not cleared. It can be too large window.Only current state is taken into account during allocation.

2.3 Workload Aware Resource Scheduling

[2] propose a scheduler which aims to find best node for the task in terms of minimum energy requirement to complete the task within the deadline and budget constraint. The proposed scheduler consists of three phases; Task submission and analysis phase which is responsible for analyzing tasks in terms of deadline and budget, Node management phase which is responsible for decision of switching nodes sleep/active states and GCRS module which also consists of sub modules – power consumption calculation, pricing unit and CGCS to find best node combination. [38] has two direction; one of them is energy reduction for cloud providers and the second is payment saving for cloud users. The main goal of the study is guaranteeing the deadline of tasks while avoiding VM migration overheads. The approach is based on laxity of tasks. It means that how long the execution of the tasks can be postponed within the deadline. The goal is minimizing the power consumption while avoiding waking up new PMs thanks to postpone tasks if it is possible. [39] aware of the workload type and resource requirements of each workload. According to the type of resource requirements, it aims saving energy with provisioning the most suitable hosts for incoming workloads. The study provides a heuristic scheduling method for specifically MapReduce workloads on heterogeneous Hadoop clusters which consists of low power (wimpy) and high performance (brawny) nodes. While provisioning wimpy Intel Atom nodes for I/O bounded workloads and brawny Intel Sandy Bridge for CPU bounded workloads, energy efficient scheduling is aimed. [40] is another workload aware resource provisioning approach. Deadline is main constraint to meet during the resource provisioning decision process. [41] provides Particle Swarm Optimization (PSO) It consists of two phases; resource provisioning and scheduling to minimize cost, meet the deadline of workflows. The study also presents VM boot time and performance variation.

Table 2.6 : Summary of some workload aware approaches in cloud computing.

Paper	Aim	Technique	KPI	Workload	Environment	Advantages	Open Points
[2]	Finding best nodes with least energy consumption and deadline budget fulfilling capacity for task allocation process	Forward only counter propagation network (CPN) based intelligent scheduler	Cost performance, CPU and memory	Synthetic workload random uniform distribution	CloudSim	Entire solution which consists of three phases; task submission and analysis phase, node management phase and GCRS module is proposed	Directly switch the host sleep state, time requirement of awaken is not considered
[38]	Reducing energy consumption and payment while considering deadline	Energy and deadline aware with non-migration scheduling (EDA-NMS) algorithm. aims to postpone the execution of the tasks to avoid waking up new PMs. turning on/off	Deadline, cost, laxity, CPU usage	Synthetic Randomly generated tasks length poisson distribution at the arrival time uniformly distributed deadline compute intensive	CloudSim	Using the laxity of submitted workloads focus on finishing all submitted tasks before the user defined deadline while considering cost. It avoids migration overhead and unnecessary PMs	How to decide the optimal number of PMs is not so clear.

Table 2.6 (continued): Summary of some workload aware approaches in cloud computing.

Paper	Aim	Technique	KPI	Workload	Environment	Advantages	Open Points
[39]	Energy aware scheduling heuristics for MapReduce workload on heterogeneous Hadoop clusters	Energy efficient scheduler Energy efficient scheduler with locality run reduce phase on wimpy nodes	Memory, I/O, CPU resource types, performance, cache conflict	Spec CPU 2000 suite eon (low memory, high IPC) and mcf (high memory, low IPC)	Xen client server model (vgnode, vgserv)	The idea is proposed that allocated workloads with heterogeneous characteristics on a PM is more energy efficient than workloads with homogeneous characteristics	Migration cost and remainin time of migrated task are not considered.

Table 2.6 (continued): Summary of some workload aware approaches in cloud computing.

Paper	Aim	Technique	KPI	Workload	Environment	Advantages	Open Points
[40]	Resource provisioning to support the deadline requirements of data-intensive applications	Since the number of resources is limited in private cloud, to meet the deadline requirement extra resources are used from the public cloud	Deadline, data transfer time, bandwidth, data locality and data size	55 tasks test app walkability index for 4 neighbours.	Aneka Real + Microsoft Azure	Considering data transfer time, data localition and the network bandwidth, deadline constraint is met.	Energy efficiency is not considered.
[41]	Finding a schedule to execute a workflow on IaaS comp. resources such that the total execution cost is minimized, and deadline is met	PSO static cost-minimization, deadline constrained heuristic for scheduling scientific workflows	I/O CPU and Memory Intensive workloads VM performance	A set of interconnected tasks Montage LIGO SIPHT CyberShake	CloudSim	Perform better with smaller sized forkflows. It considers VM boot time performance variation	Data transfer cost is not considered.

3. MODELS AND TECHNIQUES

As the demand for cloud computing services has increased, energy consumption in cloud datacenters has also increased. Energy efficiency in cloud environments has received significant attention in the past few years because of the increasing usage of system resources with developing technology and decreasing prices. To achieve energy efficiency in cloud datacenters, there are several researches aim energy saving at different levels. In this section, common models and techniques which are used for evaluation of the proposed LAA approach are mentioned.

3.1 Resource Management

In order to reduce energy consumption in cloud data centers, there are several techniques in the literature as discussed in Ch. 2. One of the common ways is switching servers off or running servers at lower energy states, when these servers are not needed because of the low demand compared to the number of active servers can meet. However, switching servers on and off and scaling resources in different energy states cause operational cost and increase workloads execution time. Moreover, because of the scheduling delay, the performance degradation would not be suprised result. [27, 36] aims energy efficiency in cloud computing while considering performance degradation and avoiding unnecessary switching servers on and off.

3.2 Virtualization

Virtualization allows creating multiple different or the same types of VMs on a single cloud server. Each VM has own OS running on top of primary OS of the server. Virtualization enables creating isolated compute environments for ensuring availability, protecting sensitive data, sharing files if needed. In terms of cloud computing and datacenters, virtualization is considered as the most promising approach to save energy by consolidation, VM re-sizing and isolation abilities.

Table 3.1 : First Fit VM Allocation Algorithm.

First Fit VM Allocation

Input: hostList, VM Output: allocation of VM
allocatedHost ← NULL
foreach host *in* hostList **do**
  if host *has enough resources for* VM **then**
  allocatedHost ← host
  add (allocatedHost, VM) to allocation
  break;
if allocatedHost = *NULL* **then**
  "no suitable host found for VM"
return allocation

3.3 Scheduling

VM allocation problem while considering performance and energy efficiency is modelled as Bin Packing Problem. More details about the differences and common characteristics of existing researches which takes the problem as Bin Packing can be finded in Ch. 2. The following VM placement algorithms are selected for evaluation of the performance and the energy efficiency in this thesis scope due to their widespread usage and popularity.

3.3.1 First fit

First fit shown in Table 3.1 is one of the oldest approaches to allocate the first free partition large enough which can accommodate the VM. The algorithm is ended after finding the first suitable free partition. To place the VM on the first PM, the total remaining capacity of the PM is compared with the resource requirement of the VM. If the remaining capacity of the PM is enough for allocation of the VM, the PM is selected for allocation. If none of PMs has enough capacity to accommodate the VM, the new PM is activated and the VM is allocated on the newly activated PM. It is the fastest algorithm compared to the other memory management algorithms. The disadvantage of this approach is the resource imbalance due to remaining unused resources left after allocation [42].

3.3.2 Best Fit

In the best fit approach, the scheduler deals with allocating the smallest free partition which meets the requirement of the VM as shown in Table 3.2. The algorithm searches the entire list of free partitions and place the VM to the first PM that has the enough smallest hole. Memory utilization is much better than first fit as it searches the smallest free partition first available. However, this algorithm is slower since it tend to fill up memory with tiny useless holes [42].

Table 3.2 : Best Fit VM Allocation Algorithm.

Best Fit VM Allocation

Input: hostList, VM Output: allocation of VM
allocatedHost ← NULL
lessFree ← MAX_CAPACITY
foreach host *in* hostList **do**
  if host *has enough resources for* VM and hostRemainingCapacity<lessFree **then**
  allocatedHost ← host
  lessFree ← hostRemainingCapacity
if allocatedHost != *NULL* **then**
  add (allocatedHost, VM) to allocation
return allocation

3.3.3 Worst Fit

The worst fit algorithm is the reverse of Best-Fit algorithm in terms of reducing the rate of production of small gaps. The algorithm aims to allocate the VM on the PM which has largest available free portion as seen in Table 3.3. Thus, the remaining capacity after allocation will be big enough to be useful. The problem of this algorithm is about availability for the VM which arrives with large resource requirements at a later stage, the largest hole is already splitted and occupied [42].

Table 3.3 : Worst Fit VM Allocation Algorithm.

Worst Fit VM Allocation

Input: hostList, VM Output: allocation of VM
allocatedHost ← NULL
mostFree ← 0
foreach host *in* hostList **do**
  if host *has enough resources for* VM
and hostRemainingCapacity>mostFree **then**
  allocatedHost ← host
  mostFree ← hostRemainingCapacity
if allocatedHost != *NULL* **then**
  add(allocatedHost, VM) to allocation
return allocation

3.4 Consolidation

Server consolidation [2, 38-40], which reduces the number of active physical machines through VM migration or by collocating VMs to a small set of physical machines. It is aimed to be reduced energy consumption by minimizing the number of active servers while running them at predefined utilization rate. The predefined utilization rate can be either maximum or an optimal value. According to the utilization rate, newly arrived workloads are allocated on one of the suitable servers. The utilization rate is also used for categorizing the servers according to their usages such as over-utilized, under-utilized. Thus, the performance degradation and energy wastage are aimed to minimize by the solutions according to categorizes servers in. Performance degradation due to the over-utilization is minimized by migrating some of the workloads from them to others while energy wastage due to the under-utilization is minimized by switching them off after migrating workloads on these servers to others.

3.5 Migration

Migration is a process of replacement VMs from the source hosts to target hosts to reduce the number of active servers as discussed in Ch. 2. Migration process consists of three main decision points; whether the migration is needed or not, which VM should be migrated and which place the VM will be migrated to. These are similar to the components of load distributing algorithms; transfer policy, selection policy and location policy. The following algorithms which are proposed as a part of CloudSim.

3.5.1 Host overloading detection

As discussed in Ch. 2, in order to provide energy efficiency and minimize performance degradation, keeping the total utilization of the CPU by all the VMs at the optimal utilization threshold is one of the preferred approach. If the CPU utilization of the server exceeds the predefined threshold, it means some VMs should be migrated from this server to reduce the total utilization in order to reduce performance degradation. Overloading detection also is used during VM allocation decision. A newly incoming workload will be placed on a server in which the utilization of the server will not exceed the predefined optimal utilization rate after the placement. [7] has proposed an adaptive threshold instead of a fixed value to decide overloading. The idea is based on adjusting the threshold value according to the taking the last ten utilization rates of a machine on average. In the paper, 4 different static techniques; Median Absolute Deviation, Interquartile Range, Local Regression and Robust Local Regression are proposed. According to the results of comparison in the paper, there is a statistically significant difference between the local regression based algorithms and the other algorithms. Energy and SLA Violations (ESV) which is a combination metric which captures energy consumption and SLA violation is used for comparison. Best results accoring to the ESV metric, LR-MMT gives the best results compared to the other combinations of algorithms. Detail of the MMT algorithm will be explained in the next VM selection section.

3.5.2 VM selection

After the overloading detection, the second step is the decision of the VM will be migrated from the overloaded server. According to [7]**,** the Minimum Migration Time Policy gives the best result with the LR compared to the other combinations as mentioned in previous section. Since the migration time of a VM is calculated as a ratio of ram requirement of the VM to the bandwidth as seen in Table 3.4, the VM which has minimum ram requirement compared to the other VMs host on the same server is selected for migration in this policy shown in Table 3.5.

Table 3.4 : Maximum VM Migration Time Algorithm.

Maximum VM Migration Time

Input: host, vmList Output: maximumVMMigrationTime
maxRam ← Integer.MIN_VALUE
foreach vm *in* host.getVMList **do**
  ram = vm.getRam()
  if ram > maxRam **then**
  maxRam ← ram
return maxRam/hostBW

Table 3.5 : Minimum migration time policy algorithm.

Minimum Migration Time Policy

Input: host, migratableVMList Output: vmToMigrate
minMetric ← Double.MAX_VALUE
foreach vm *in* migratableVMList **do**
  vmToMigrate = *NULL*
  metric = vm.getRam()
  if metric < minMetric **then**
  minMetric ← metric
  vmToMigrate ← vm
return vmToMigrate

3.5.3 VM placement

Final step of the migration process is the decision of place to host the selected VM. The selected placement server should not exceed the optimal utilization rate after the migration. Moreover, the server has enough capacity to meet the requirements of the VM. [7] proposes an algorithm based on Best Fit Decreasing algorithm named the Power Aware Best Fit Decreasing algorithm as seen in Table 3.6.

Table 3.6 : Power Aware Best Fit Decreasing (PABFD) algorithm.

Power Aware Best Fit Decreasing (PABFD)

```
Input: hostList, vmList Output: allocation of VMs
vmList.sortDecreasingUtilization()
foreach vm in vmList do
    minPower ← MAX
    allocatedHost ← NULL
    foreach host in hostList do
        if host has enough resources for VM then
            power ← estimatePower (host, vm)
            if power < minPower then
                allocatedHost ← host
                minPower ← power
    if allocatedHost != NULL then
        allocation.add(VM, allocatedHost)
return allocation
```

3.6 Key Findings

Key findings of both Ch. 2 and Ch. 3 include:

1) Switching off idle resources is more energy efficient than keeping these resources at idle state under some conditions

2) Migrations may affect energy consumptions due to that the migration is time and energy consuming process.

3) Approaches including prediction module gives better results in terms of energy efficiency due to considering not only the current state of the system but also the future demand during allocation/consolidation process.

4) Remaining time of already running workloads on the system is an important parameter for the decision of allocation and migration.

5) Single threshold usage is not a sufficient for allocation decision. The combination of system's trend and utilization threshold gives more optimal decision compared to the single threshold usage.

4. EXPERIMENTAL METHODOLOGY

The workload placement problem while considering both energy consumption and performance has been introduced in this chapter. To find an energy efficient solution for VM allocation, a comprehensive approach has been proposed. It consists of monitoring module, forecasting module and allocation module. Monitoring module is responsible for observation of the systems current status in terms of usage of resources, the number of resources and categorizing them according to their usages. Forecasting module is based on HW. ARIMA and HW have been applied into the same time series and since HW gives the better results compared to ARIMA for these time series, HW is decided to be used. Allocation module uses data comes from both forecasting module and monitoring module for allocation decision while considering systems current state and future demands. Responsibilities of each modules have been explained in detail to provide a broad perspective about the solution in the system model sub section. Moreover, the energy model has been built with several equations in the Energy model sub section. The proposed algorithms of LAA have been explained with example use cases scenarios. CloudSim is used as a simulation environment to apply the proposed approach and to evaluate the performance of it. The existing power package provides migration algorithms has been extended to be suitable for running dynamic workload with. An additional package has also been implemented to provide LAA approach in CloudSim. LAA has been evaluated with real workload data named Google Tracelog. To ensure about the robustness of the selected forecasting algorithm, it is decided that the Google Tracelog usage instead of ITU IT department logs. Forecasting results belong both workloads have been provided under the implementation sub section.

4.1 Problem Definition

The objective of this thesis is to propose an energy efficient resource allocation while considering the minimization of performance degradation. The model deals with the optimum number of servers relying on the adaptive optimal utilization of hosts with respect to the consumed energy to meet dynamic workload requirements based on prediction. Many studies in the literature aim to run resources at the maximum utilization rate to provide energy efficiency. Note that the utilization of resources can

be managed by introducing new processes to the servers running preemptive schedulers. However, under- or overutilization of servers causes performance degradation. Each resource type has its own optimum utilization rate. In addition, turning servers on and off unnecessarily causes performance degradation. Workloads can be classified according to resource needs. In this study, we address computationally intensive workloads. The CPU needs of this type of workload are more intense than those of other resource needs. The threshold for the optimal utilization rate of the CPU is set at 70% [13]. Through an adaptive decision mechanism, the unnecessary turning of servers on/off is avoided to increase energy efficiency.

4.2 System Model

The proposed model consists of three functional modules: a monitoring module; an allocation module, with physical and virtual servers; and a forecasting module as shown in Figure 4.1.

It is assumed that the users' requests come to the system with the CPU requirements and execution time information. Based on the resource requirements of tasks, tasks are mapped with the most suitable VMs. VM types are from Amazon EC2 instance types. Extra Large Instance (2000 MIPS, 3.75 GB); Small Instance (1000 MIPS, 1.7 GB); and Micro Instance (500 MIPS, 613 MB). Initially, the most appropriate VM is allocated for the incoming workload with the resource requirements defined by the VM types. Then, VMs utilize less resources according to tasks requirement [7]. The allocation module is responsible for finding the suitable host to allocate the VM while considering resource capacity and energy efficiency. The allocation module gets the system's current state from the monitoring module and the prediction of the system's trend from the forecasing module. Physical servers are categorized based on their CPU utilization rates by the monitoring module. All physical servers are passive at the initial. To allocate the workloads, some of servers are switchied on accoring to the decision of the allocation algorithm. When the utilization rate of a server exceeds the optimal utilization rate, this server is evaluated as overutilized. If the utilization rate is less than the optimal utilization rate, it means that the server is underutilized. The optimal utilization rate of the server means that the optimal utilization rate of the CPU in this thesis scope. It is defined as 70%. In addition, if the server's utilization rate is

in the range between 10% up and down values of the threshold, it means the server is in well utilized server list. If all of the workloads are executed and finished on any server, it means the server is considered as in idle server list. Therefore, there are five server lists; passive, under-utilized, over-utilized and well-utilized. The monitoring module is responsible for monitoring the servers and put them to the related resource list according to their utilization. The allocation module uses these five server lists in the algorithms while deciding the allocations of VMs.

Beside that the monitoring module, the forecasting module is also other module which provides data to the allocation module. System's trend is an other important factor which is used by monitoring module. Forecasted future demand is provided to the allocation module by the forecasting module.

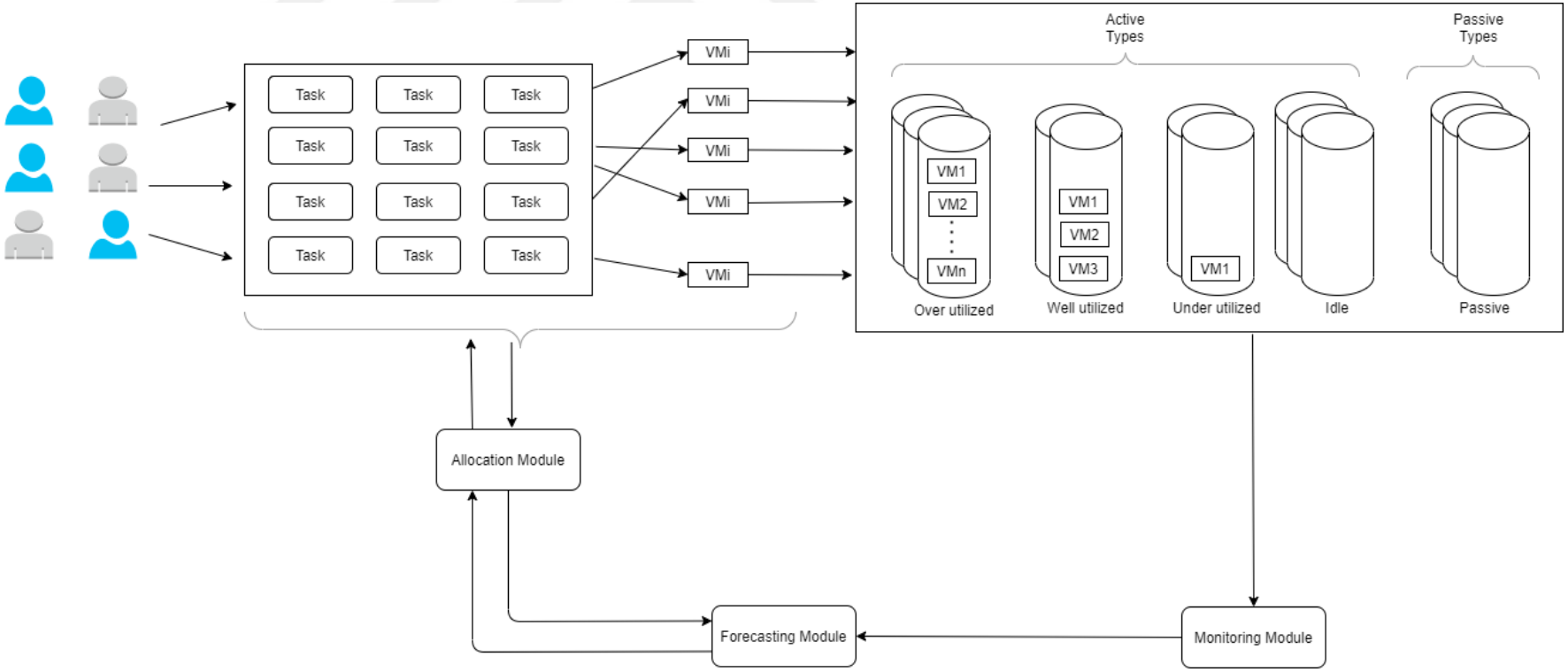

Figure 4.1 : System model.

5 mins interval is used as a monitoring window of both the forecasting and allocation modules. Tasks in the same window are handled at the same time by the allocation module. It means all the tasks coming in 5 mins are put in the queue and these tasks are allocated to the VMs and PMs respectively. The forecasting module also uses 5 mins interval. The interval is defined as 5 minutes according to the specs of the machine which is used in this thesis. This interval may differ according to the different machines. It is assumed that the cloud data center consists of homogeneous servers in the scope of this thesis. Initially, only CPU is considered in term of resource type consumes energy. HP Proliant XL170R G9 Xeon 2670 G4 includes 2x128 GB Ram, Intel Xeon 2670, 2x12 cores x 2300 MHz. Power consumption of selected server with respect to utilization rate are shown in Figure 4.2.

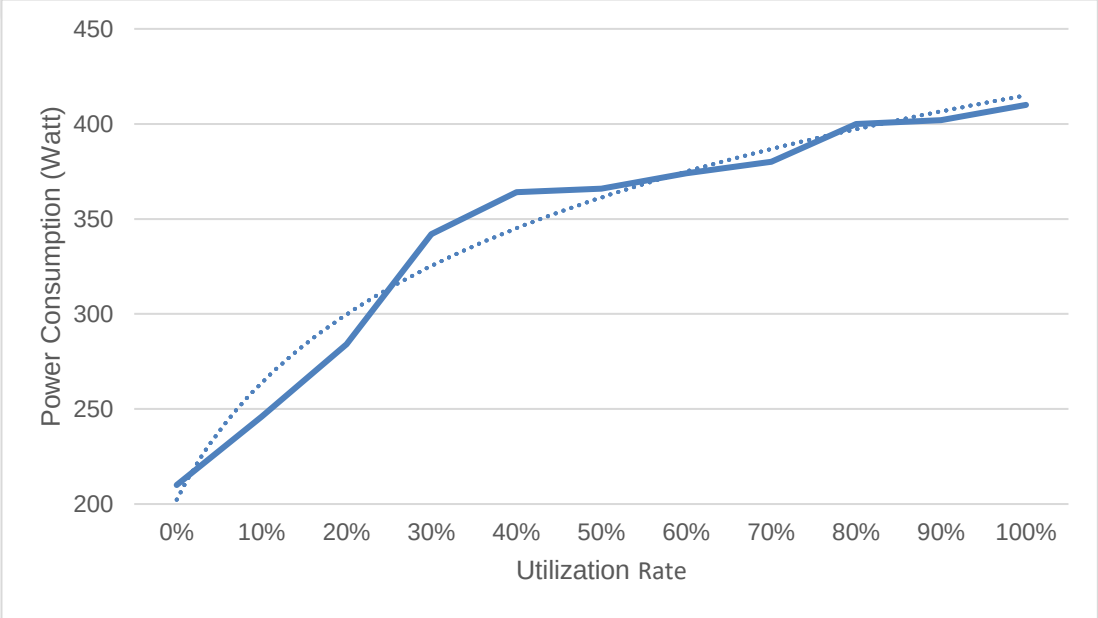

Figure 4.2 : Power consumption of HP Proliant G9 at different load level.

In addition, based on the experiments conducted with TÜBİTAK B3 Laboratory, the energy consumption for switching the server off and on (E_OO) is measured as 0,019kwh. The result is obtained as an average of 5 times measurement as shown in Figure 4.3. Shut down period takes 36 seconds and consumed 3600ws. The time it takes for the server to be ready to operate after the power has been turned on is 248 second and energy consumption for boot time is around 64000ws. As shown in Figure 4.3, during boot time, power consumption is increased until 336 watt which is almost the power at 30% utilization of the server according to the server's spec and then it is

decreased to the power at idle state of the server. The model presented in Equation 1 can be used to calculate approximately value of E_OO for any server.

$$E_{OO} = 1/2(t_2 - t_1) * P_{idle} + 1/2(t_2 - t_1) * P_{20\%} + (t_3 - t_2) * (P_{20\%} + P_{idle}) \qquad (4.1)$$

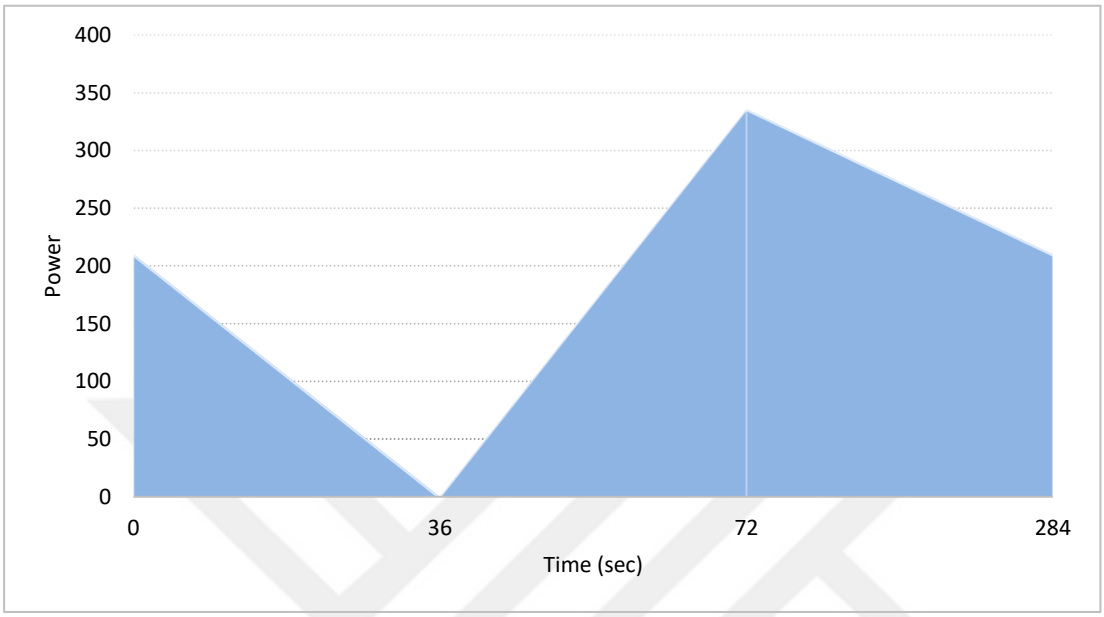

Figure 4.3 : Power consumption for switching the server off and on.

4.2.1 Prediction method

Time series can be expressed as past values of a variable observed in a time period. For example, the observed time series is X1, X2, X3, …, Xn. At time n, k steps ahead is wanted to be predicted. In this case, the value to be predicted is Xn+k. There are many forecasting methodologies in the literature. There is no simple answer to the question of which one is the best [43].

There are many factors to consider, such as the characteristics of the data, the historical value length of the time series while deciding the forecasting method. In this thesis scope, Holt Winters and ARIMA are applied for the selected workloads and compared with each other in terms of error rates.

Holt Winters generates short-term forecasts for data with trend and seasonality patterns. There are two types, incremental and multiplicative. The methodology is based on three basic equations [44-47].

L represents level, T represents trend, and S represents seasonality. The smoothing coefficients α , β and γ are used in these equations. The period in which seasonality is

seen is expressed by the cycle. Incremental type of Holt Winters is used in this thesis scope and equations of incremental type are seen in below;

$$L_t = \alpha(Y_t - S_{t-c}) + (1-\alpha)(L_t - 1 + T_{t-1}) \quad (4.2)$$

$$T_t = \beta(L_t - L_{t-1}) + (1-\beta)(T_{t-1}) \quad (4.3)$$

$$S_t = \gamma(Y_t - L_t) + (1-\gamma)(S_{t-c}) \quad (4.4)$$

$$F_t = L_{t-1} + T_{t-1} + S_{t-c} \quad (4.5)$$

The following equations can be used when finding the initial values;

$$Lc = \frac{1}{c}\sum_{i=0}^{c} yi \quad (4.6)$$

$$T_c = \frac{1}{c}\left[\frac{y_{c+1}-y_1}{c} + \frac{y_{c+2}-y_2}{c} + \ldots + \frac{y_{2c}-y_1}{c}\right] \quad (4.7)$$

$$S_i = Y_i - L_c, i = 1,2,\ldots,c \quad (4.8)$$

In the Holt Winters method, determining the smoothing coefficients at values suitable for the data to be estimated affects the success of the algorithm and reducing the error rate. Various methods can be used to determine these values such as Cplex, Excel solver. In this thesis scope, Excel solver is used for determining smoothing coefficients.

The estimation method is applied on the sample observation values after deciding possible models. Error rates are found by comparing the estimation results with the actual observation values. The model that gives the least error rate is decided. There are many methods in the literature for calculating error rates. Within the scope of this thesis, MAPE method is used [48].

$$M = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{At-Ft}{At}\right| \quad (4.9)$$

ARIMA describes time series with its historical values and probabilistic error term [3]. ARIMA takes three parameters; p, I and q. The auto regressive is based on the relationship the time series with previous past values up to p paramater AR(p).

$$Y_t = \alpha_1 Y_{t-1} + \alpha_2 Y_{t-2} + \ldots + \alpha_p Y_{t-p} + u_t \rightleftarrows \quad (4.10)$$

I is the stationary paramater. If the mean and covariance are not change in the time and the covariance between two periods depends on the distance between the periods, not the period viewed, the time series is considered as stationary time series. If the time series is not stationary, I paramater is used to make it stationary. The moving average is based on the relationship of the time series with error rates. How much history to go to is also shown with the q parameter MA(q).

$$Y_t = \mu + u_t + \beta 1 u_{t-1} + \ldots + \beta u_{t-q} \rightleftarrows \quad (4.11)$$

Depending on the time series, the appropriate method to be used for the forecasting may be only the moving average or only the auto regressive model. Box Jenkins is the whole of this family of models. It is difficult to find out which model is the most appropriate one for the selected time series. Auto Correlation Function (ACF) and Partial Auto Correlation Function (PACF) can be used to find the appropriate model. As a result of the interpretation of these functions, one or more Box Jenkins models are decided. This model makes predictions. Comparisons are made with the available observation values. The model that gives the least error rate is determined and this method is used for the estimation.

The monitoring module is responsible for observing the CPU requirement (CR) and execution time (ET) of each incoming workload during the last 5 min. The processing requirement of each workload W_i is calculated by multiplying CR_i and ET_i. The total processing requirement (PR_{total}) is the sum of submitted workloads' processing requirements in that 5 min interval. The monitoring module transfers the obtained knowledge to the forecasting module and the workload placement module.

The forecasting module determines the required number of processing units (*Nr*) according to user demand. Previously, the forecasting module used PR_{total} time series to predict future demand. Since PR_{total} does not have seasonal patterns and trends, forecasting methodologies, such as HW, ARIMA, support vector regression and nonlinear regression, give results with a high error rate. The approximated total processing requirements (APR_{total}) are used to calculate *Nr*. APR_{total} uses the mean value of both the CPU requirement and execution time instead of exact values. The noise of the PR_{total} time series is filtered by using the mean value of the parameters. APR_{total} is calculated by multiplying the number of submitted workloads (*NoSW*), mean execution time of submitted workloads (*MET*) and mean CPU requirements of submitted workloads (*MCR*) in intervals. *NoSW, MET,* and *MCR* are forecasted

separately through forecasting methodologies, namely, HW and ARIMA. HW gives significantly better results than ARIMA in terms of the Mean Absolute Percentage Error (MAPE) for each time series since the time series includes seasonality and trend. In addition, because the interval is short and the period being forecast is long, ARIMA is not the right choice. To use optimal system resources (SC_{total} indicates the total capacity of a server and *Th* indicates the optimal utilization rate), the following equations are used to forecast the required number of processing units (SC_{total} indicates the total capacity of a server and *Th* indicates the optimal utilization rate).

$$APRtotal = NoSW * MCR * MET \tag{4.12}$$

$$Nr = \frac{APR_{total}}{SC_{total} * Th} \tag{4.13}$$

The forecasting module based on the Holt Winters forecasting methodology has an 8.85 error rate.

4.3 Energy Model

The cumulative energy consumption of a server (E_CS_i) is equal to the sum of the energy required the switching the server off and on (E_OO_i) and the energy required for the actual service time (E_Ser_i). It is given by:

$$E\_CS_i = E\_OO_i + E\_Ser_i \tag{4.14}$$

The sum of the energy consumption of servers in a data center is equal to the total energy consumption of the data center (*E_CDC*), as shown in Equation 15, where m indicates the number of servers in the data center.

$$E\_CDC = \sum_{i=0}^{m} E\_CS_i \tag{4.15}$$

During the service time, the server may run at different utilization levels that consume different amounts of power. Power consumption figures are published by computer producers. Total energy consumption during service time is shown in Equation 16, where *t* indicates the time spent at a particular level of processor usage given by *Power[j]*. Note that the power model of the server is based on the utilization level which is split into eleven rates from 0% to 100%. For example, with the placement of a newly incoming workload, the utilization level of the server can be increased from 10% to 30%. It means that the server has never run at 20% utilization level and t equals

0 for the Power[2]. The discrete power model is used to be compliant with the simulator.

$$E\_Ser_i = \sum_{j=0}^{10} t * Power[j] \quad (4.16)$$

In addition, the migration cost is not an ignorable parameter when calculating the energy consumption of a system. The real energy consumption (ER) can be obtained by adding costs of migration and switching on to this result, as seen in Equation 17.

$$ER = E\_CDC + NoM * \text{MigrationCost} \quad (4.17)$$

PI_i is a special representation of $\text{Power}[0]$ of a Server i. t_{idle} shows the idle time of Server i. If the required energy for switching a server on/off is more than the required energy for running the server at idle state for a period, then keeping the server in an idle state during that period is more energy efficient. The monitoring window is decided based on Equation 18. In the calculated window, keeping a server in the idle state is more energy efficient than switching a server on/off to meet the incoming request. According to the spec of the selected server in this research, the monitoring interval is 5 min.

$$PI_i * t_{idle} < E\_OO_j \quad (4.18)$$

4.4 VM Consolidation Algorithm

The workload placement module is responsible for the allocation of workloads to suitable VMs and the allocation of these VMs to suitable servers. According to the information received from the monitoring module and forecasting module, decisions about turning a server on/off and the placement of incoming workloads are made in this module.

The consolidation algorithm is based on a single threshold to decide whether the server is over utilized. In other words, if a server's CPU utilization ratio exceeds the predefined threshold, then the server is over utilized; otherwise, the server is underutilized. If the utilization of the server equals the threshold, then the server is running at the optimal utilization rate. Unlike other studies, overloading detection does not trigger VM migration. Overloading is undesirable since it causes performance degradation, but it can be acceptable under some conditions. To decide the allocation of incoming workloads, this threshold is not a sufficient parameter. In addition to the

threshold, future demands are also considered to be as important as the system's current state. There are two essential trends in the VM Consolidation Algorithm: Shrinking and Enlarging. Since the algorithm starts with no active server, the default trend in which the system is going to enlarge to serve incoming workloads by introducing new servers is enlarging. The second trend is shrinking, in which the number of active servers will be reduced because the number of expected workloads can be completed with a smaller number of servers than the number of existing ones.

All the servers in the system are in the passive (i.e., available in the system to be activated upon request) list initially. When a request arrives, one of the passive servers is turned on to meet the requirement. When all the allocated workloads on a server are completed, the server is added to the idle host list. Then, the server is turned off and moved from the idle host list to the passive host list. According to the load of the server when it is in an active state, it is either on the underutilized server list or overloaded server list.

The proposed algorithm also uses a new set of parameters, such as the remaining execution time of a workload, active number of servers (N_a), required number of servers (N_r) and efficient utilization threshold. The system can be unstable in two cases: (1) N_a is greater than N_r, which means that there are underutilized servers, causing energy inefficiency, or (2) N_r is greater than N_a, causing overutilized servers and performance degradation when new servers cannot be switched on.

The proposed model to find the most appropriate server in terms of energy efficiency, as shown in Algorithm 1, is based on the shrinking or enlarging trend of the system. In the shrinking trend, the system has more active servers than required for foreseen demand. There are three different types of servers from the perspective of utilization: idle, underutilized and utilized. If the number of idle servers equals the difference between the number of active servers and the number of required servers, then the servers will be switched off. Moreover, if the number of idle servers to be switched off is not sufficient for optimal solution, then a subgroup of underutilized servers, and even utilized ones for some extreme cases, will also be switched off as soon as their respective ongoing workloads are completed. To support the shrinking trend when a workload is submitted, the workload is assigned to the most suitable active server instead of activating a new server for the workload. The most suitable server selection

algorithm starts with finding the longest remaining execution time of the running workloads among underutilized servers.

The selection of the server with the highest remaining time for the existing workloads (seen in Table 4.2) makes it possible to overlap the execution time of the newly arrived workload with it. Therefore, the newly arrived workload is expected to be completed after the completion of the existing workloads in the worst case. If there is more than one suitable server, then an additional parameter called minimum violation is used to decide the most appropriate one. The efficient utilization rate of the CPU is determined to be 70% for energy efficiency. Given that 70% is the optimal value, a range between 65% and 75%, whose mid-value is 70%, is accepted as the energy efficiency range. Minimum violation is the most approximate utilization rate to the efficient utilization rate when the incoming workload is assigned to them.

Table 4.1 : Look ahead VM allocation algorithm.

Look Ahead VM Allocation

Input: Set of Workloads W, underutilizedServerList, passiveServerList
Output: Set of Provisioned VMs and Allocated Workloads
for $W_j \in W$ **do**
  if active server list is empty **then**
    select and switch on a server S_i from passiveServerList;
    create VM_1 in S_i;
    allocate W_j on $VM_{i,1}$;
    update Server Lists according to utilization rate of S_i;
    continue;

  if Nr > Na **then** //Enlarging Trend
    if underutilized server list is not empty **then**
      select a server S_i from underutilizedServerList;
      create VM_k in S_i;
      allocate W_j on $VM_{i,k}$;
      update Server Lists according to utilization rate of S_i;
      continue;

  else //Shrinking Trend
    $S_i \leftarrow$ call HRT
    if S_i is not null **then**
      create VM_k in S_i;
      allocate W_j on $VM_{i,k}$;
      update Server Lists according to utilization rate of S_i;
      continue;

    else
      $S_i \leftarrow$ call LRT
      if S_i is not null **then**
        create VM_k in S_i;
        allocate W_j on $VM_{i,k}$;
        update Server Lists according to utilization rate of S_i;
        continue;

  if W_j is not allocated on any VM **then**
    select and switch on a new server S_i from passiveServerList;
    add S_i to active server list;
    create VM_1 in S_i;
    allocate W_j, on $VM_{i,1}$;

Table 4.2 : Highest Remaining Time algorithm.

HRT

Input: underutilizedServerList, W_{new} ,RC, UR
Output: targetServer

```
threshold ← 0.70;
minViolation ← 0.70;
highestRT ← ET[W_new] ;
targetServer ← null;
foreach server S_i in underutilizedServerList
    if  RC[S_i]>= UR[W_new] then
        TU[S_i] = TU[S_i] +UR[W_new] ;
        violation = threshold –TU[S_i] ;
        if minViolation > violation
                && max(RT_{i,j}^W)  > highestRT then
                minViolation = violation;
                highestRT = max(RT_{i,j}^W) ;
                targetServer = S_i;
return targetServer;
```

In the example in Figure 4.4, the number of active servers is more than the future demand; therefore, the number of active servers should be decreased. Assuming that these two servers from the active server set have been selected to be turned off, the server with the highest remaining time is selected for the incoming workload W_4. If W_4 is allocated on Host 1, then it prolongs the time in standby of the server. Conversely, when W_4 is allocated on Host 0, it does not cause the same issue. The cost of running the same server with the same resource requirement is not considered in this study.

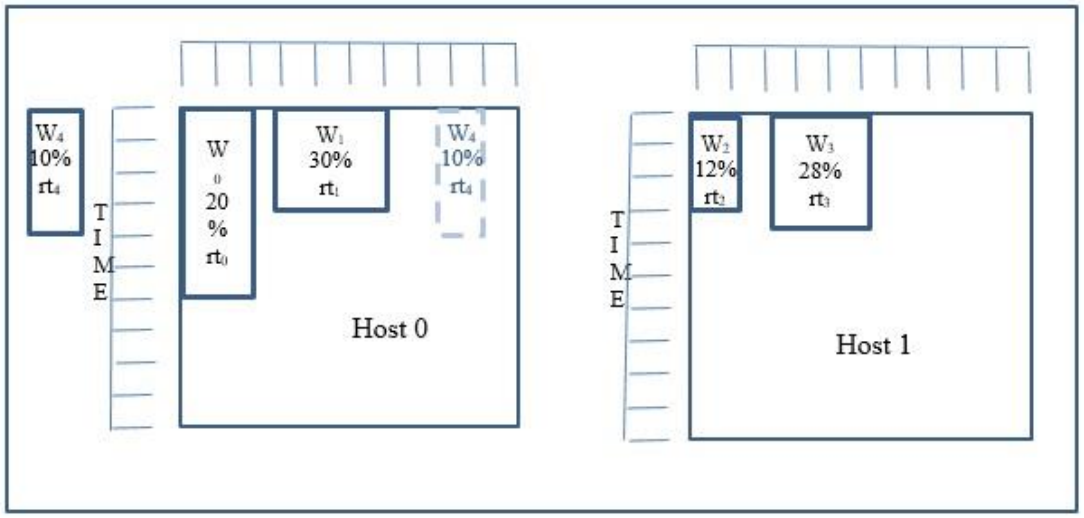

Figure 4.4 : Example allocation of HRT.

If there is not a suitable server to host the new workload according to the parameters of the highest remaining time and minimum violation, the search algorithm is conducted among servers and will be overloaded after the assignment. Running in the overloaded state causes performance degradation in terms of response time and throughput. In addition, switching a new server on to meet short-term needs can cause redundant energy consumption when the systems trend is considered. Therefore, the proposed approach is based on the selection of the most suitable server from the set of active servers instead of switching a new server on. The temporarily overloaded state is acceptable when the system has a number of active servers that is greater than the required number of servers for future demand, i.e., the system is in a shrinking trend. To reduce the running time of the selected server in the overloaded state, the workload with the shortest remaining execution time according to the optimal utilization rate after assignment should be run, as described in Table 4.3. The shortest remaining time of existing workloads is not the only parameter used. The utilization rate of the server with the shortest remaining time is another parameter to be considered. When the workload is finished, the utilization rate of the server should approximate the optimal utilization rate. In addition, the minViolation parameter is used to ensure that the server is up to 100% utilization.

In Figure 4.2, W_0 and W_1 are running in Host 0, while W_2 and W_3 are running in Host 1. W_4 is newly arrived with a 30% utilization requirement, and rt_4 shows that the remaining time of W_4 is equal to its execution time. When the new workload is allocated on either Host 0 or Host 1, it causes performance degradation since the utilization rate will exceed the optimal utilization threshold. The least remaining time on Host 0 is rt_1 of W_1, and rt_3 is the shortest remaining time of Host 1. The servers are

Table 4.3 : Least Remainin Time algorithm.

LRT

Input: underutilizedServerList, W
Output: targetServer

threshold ← 0.70;
minViolation ← 0.30;
leastRT ← ET[W_{new}] ;
targetServer ← null;
foreach server S_i in underutilizedServerList
  if RC[S_i] >= UR[W_{new}] **then**
    TU[S_i]= TU[S_i] + UR[W_{new}] ;
    violation = TU[S_i]- threshold;
    if minViolation > violation
      && min($RT^W_{i,j}$) < leastRT **then**
      minViolation = violation;
      leastRemainingTime = min($RT^W_{i,j}$)
    ;
      targetServer ← S_i;

return targetServer;

compared to decide which server is the most appropriate in terms of the least performance degradation. However, in the example shown in Figure 4.5, the shortest remaining times on Host 0 and Host 1 are equal. Therefore, the decision is made according to the utilization rate. For this case, looking at the further utilization rate

when W_1 and W_3 are finished, Host 1 will be well utilized while Host 0 will be overutilized. Therefore, Host 1 is selected for the placement of W_4.

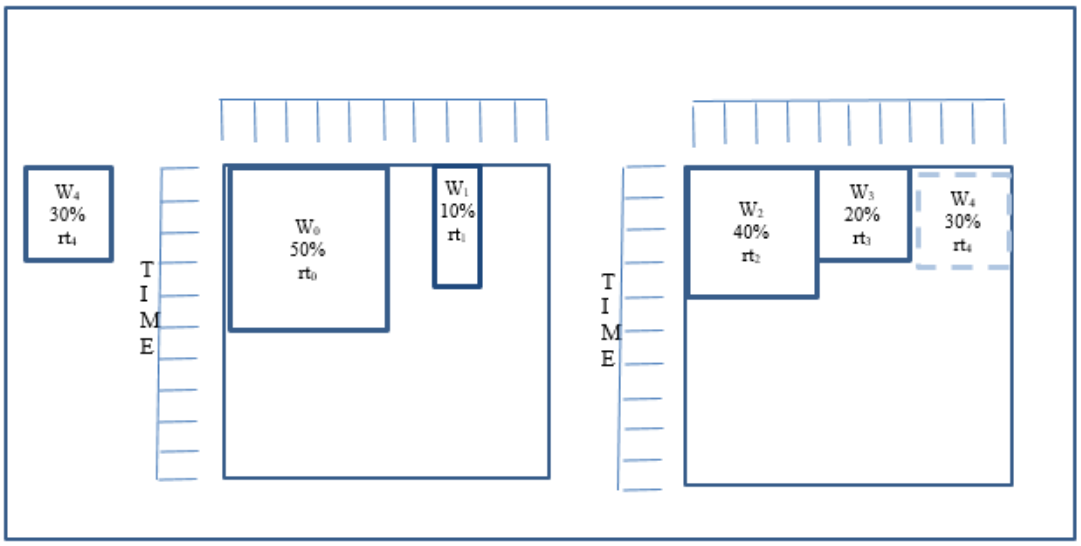

Figure 4.5 : Example allocation of LRT.

In the enlarging trend, if N_a is less than N_r, then a destination host is investigated from the underutilized server list to be well utilized when the server is allocated for the workload. If there is no suitable server among the underutilized servers, then the idle server list is checked to determine whether the list is empty. If the idle server list is not empty, then one of the idle servers is used to run the workload. Otherwise, a server from the passive server list is switched on.

The algorithm is based not only on the current state of the system but also on future demands, as described below. The number of incoming workloads is forecasted by using different methodologies: Holt Winters and ARIMA.

4.5 Implementation

One of the most preferred simulator is used for the implementation and evaluation of the proposed approach. It is easy to use and extend. Moreover, CloudSim is mostly preferred by the previous related researches in the literature. CloudSim provides VM migration approaches under the power package. It is suitable to work with static workload. It is extended to be work with dynamic workload as well. In this manner, it has become more similar to the real world. Beside that the extension of existing power package, another package has been implemented for LAA. These two packages are explained in CloudSim section. Workload selection is important for the evaluation of

the achievement. In the beginning of this thesis, ITU IT Department user logs have been used to simulate the workloads in cloud. Then, It is decided that to use Google Cluster Tracelog since it provides longer data trace with more detail such as resources requirements compared to ITU IT Department. The proposed approach includes a prediction module which contributes energy saving thanks to avoid unnecessary switching servers off. Therefore, selection of the prediction methodology effects the success of the proposed approach. All the prediction methods implementations and workloads details are expressed in the following sub sections.

4.5.1 CloudSim

CloudSim [14] is one of the most commonly used cloud simulation toolkits. CloudSim provides several VM allocation and migration policies and is mainly focused on IaaS-related operations. However, the provided allocation algorithms run with static workloads. All VMs are created at the beginning and terminated at the end of simulation. There is no capability to add or remove VMs during simulation runtime. The existing code of CloudSim is extended to meet dynamic workload requirements and make a fair comparison with the proposed algorithm as shown in Figure 4.3. Therefore, core classes of CloudSim are extended to be able to create and terminate VMs at runtime. In addition, to validate the accuracy of the proposed model, it is important that a sufficient number of data is available. Therefore, Google Cluster data [49] is used to evaluate the algorithm. The existing code of CloudSim is extended to meet the required capabilities: (i) creating VMs at runtime; (ii) read the Google Cluster tracelog data; (iii) VM allocation according to the proposed algorithm; and (iv) extend the Power Package classes to be able to same functionality with the proposed algorithm.

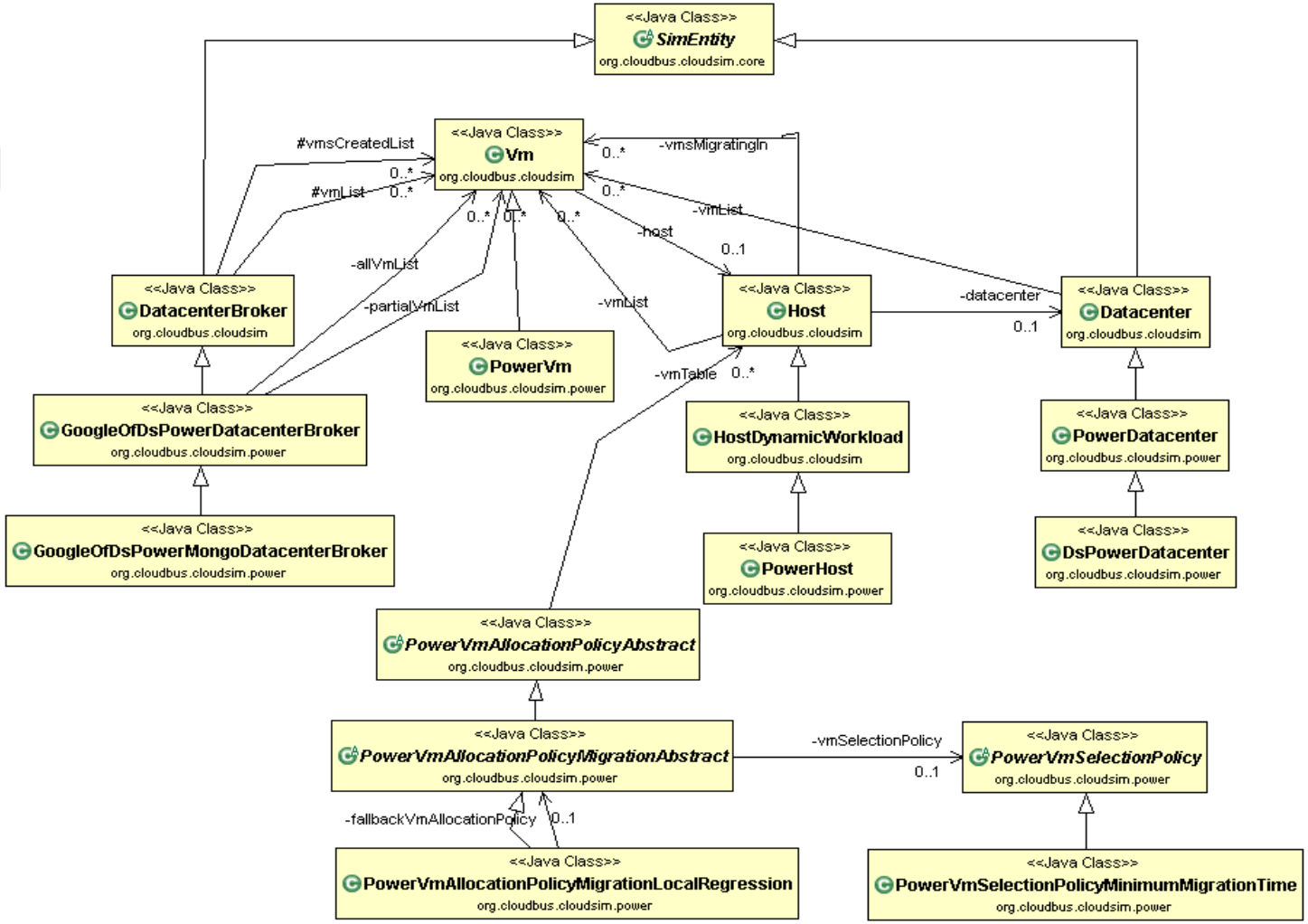

Figure 4.6 : Class diagram of extended Power package.

DatacenterBroker class is extended to create VMs dynamicaly and read Google Cluster tracelog. DatacenterBroker class is also responsible of workload and VM mapping. To implement Look Ahead approach, a new package named LAA is created and related class diagram is shown in Figure 4.4. Datacenter class is extended to implement VM allocation algorithm and calculation of the N_r. N_r is calculated as a sum of constant and a ratio of the forecasted Million Instruction Per Second (MIPS) for the specific time slot to 70% of the capacity of the PM. DatacenterCharacteristic class is another class we extended to keep 4 types of Host lists namely underloaded host lists, overloaded host lists, idle host lists and passive host lists. Host class is extended with the capability of the host utilization calculation as a ratio of total allocated MIPS (a sum of the allocated MIPS to each VM on the host) to the total MIPS of the host. Beside that, when the host is idle, there is an assumption that the host does not consume energy. However, the power consumption of the CPU in idle state is more than 50% of the fully loaded state [9-12]. It is fixed in both PowerHost and LAHost classes. LAHost is also responsible for returning the VMs with highest remaining time and least remaining time. To terminate the VMs which are completed, returning the VMs to remove list is also a function provided by this class. Cloudlet is a job/task to be run on VM. The Cloudlet class is also extended with the constructer including utilization rate and a get method to return cloudlet utilization. This value is gathered from the Google tracelog CPU requirement column. Cloudlet length is also set with the execution time column of the Google Cluster tracelog. VMAllocationPolicySimple is extended to provide the algorithms described in Section 4.4. Broker class and Datacenter class in the power package are also extended with the capability of dynamic VM creation/termination and reading Google Cluster data in order to make fair comparison between the existing approach and the proposed approach. Cloudlet is created by setting the execution time and the required CPU usage through reading the Google Data in the extended DatacenterBroker class. VM creation responsibility is also given to this class. The energy efficiency calculation is also harmonized within the proposed model.

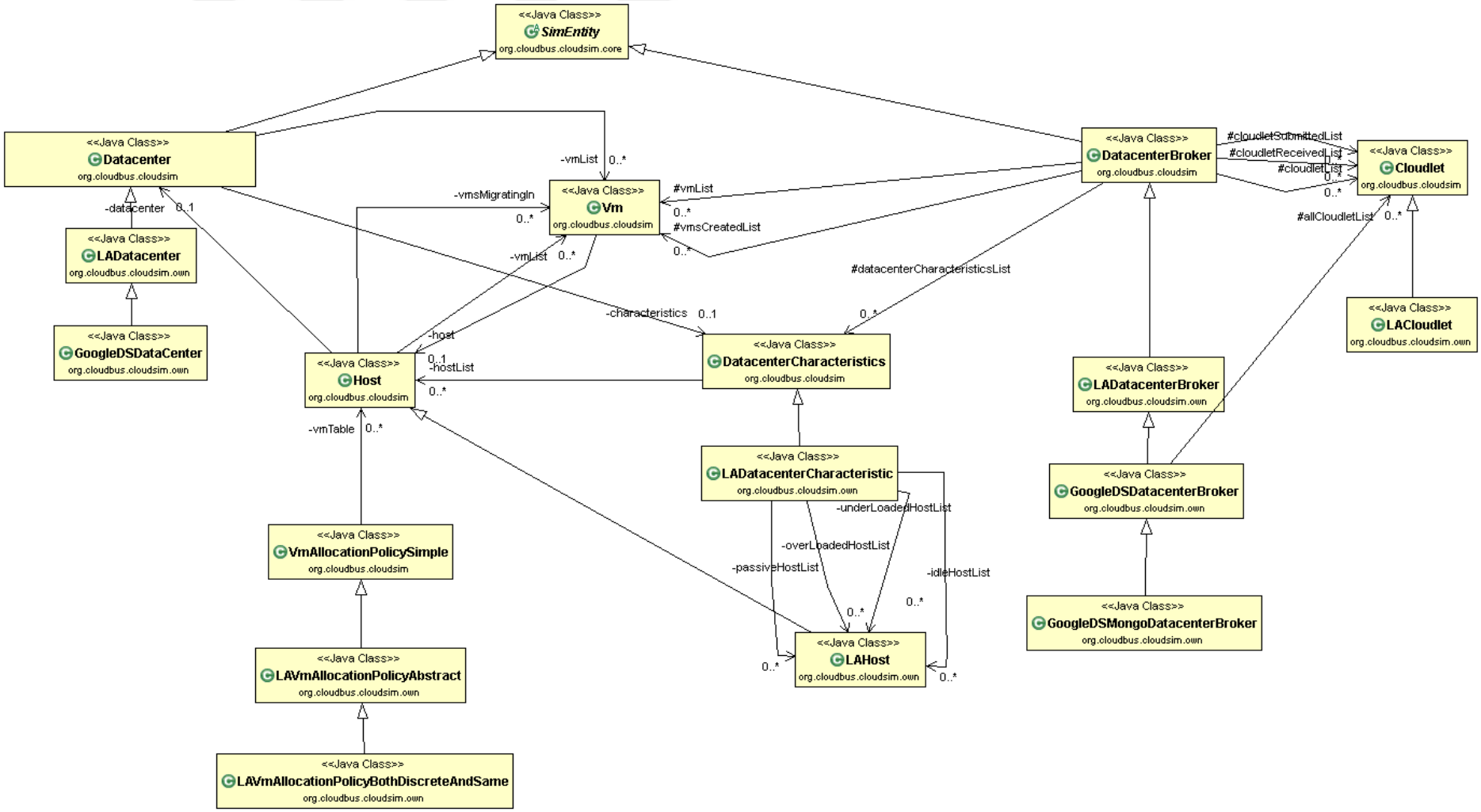

Figure 4.7 : Class diagram of the LAA approach.

4.5.2 ITU IT Department Tracelog

Istanbul Technical University IT Department 3 days user entrance tracelog is used as a dataset. In the beginning of the implementation of the forecasting module, the Holt Winters forecasting method is applied in the ITU IT Department dataset. First day data is used to calculate initial seasonal values. To calculate the trend and level values, second day user entrance data is used. User requests are forecasted by using the level, trend and seasonal information obtained from the first two days. Actual third day data and and forecasted third day data are compared. Thus, Holt Winters model which considers mathematical, seasonal and trend analysis is used to predict the user requests. However, there are a lot of various forecasting methodologies in the literature. For validation of convenient of Holt Winters for this workload, some of other forecasting approaches are also applied and compared error rate with Holt Winters's. Therefore, Auto Regressive Moving Average, Auto Regressive, Moving Average and Auto Regressive Integrated Moving Average models are conducted separately and compared with Holt Winters's results. ARMA(p,q) gives the best result compared to the other variation of the ARIMA (namely; AR(p), MA(q), ARIMA(p,I,q)). Holt Winters gives the better result than ARIMA(1,0,1) with the error rates 0.26 and 0.37 respectively as shown in Figure 4.8.

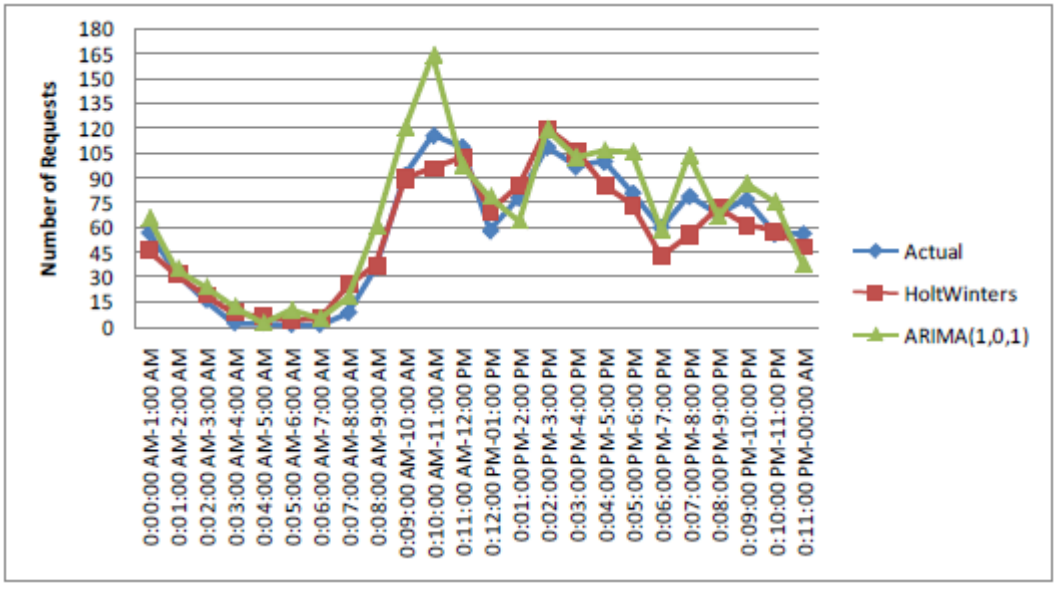

Figure 4.8 : Holt Winters and ARIMA results for ITU IT department tracelog.

Although ITU IT Department time series is useful and forecasting methods with this tracelog gives satisfactory results, since the algorithm uses predictive data as an

important parameter when deciding on VM placement, it is desired to develop the dataset and ensure that the appropriate estimation method is used for the dataset. ITU IT Department time series consists of the number of user entrance within 1 hour interval during 3 days period with 1 day cycle. To simulate the workload's resource requirement, random type of VM is created for each user entrance. The forecasting module only estimates the number of user entrance to the site, not the actual resource requirement from the system. Moreover, to make prediction more accurate, more data with the longer period than 3 days is required. Therefore, the google cluster tracelog is decided to be used.

4.5.3 Google Cluster Dataset

Google published trace data on a cluster of approximately 12.5 k machines including 29 days of cell information from May 2011. A cell means a set of machines sharing a common cluster-management system. A job consists of one or more workloads. Google shared the trace data through six tables: machine events, machine attributes, job events, workload events, workload constraints, and resource usage. In the scope of this paper, the workload event table is used. The workload event table consists of timestamps, missing info, job ids, workload indexes, machine ids, event types, usernames, scheduling classes, resource requests for CPU cores, RAM and local disk space and diferent-machine constraints. Timestamps are in microseconds. Event types have diferent values, such as submit, schedule, fnish and fail. When the task is seen as the first time with status type of event as "submit", it is put into the map; the value submission time and the key consists of the concatenation of the job id and task id. If the key already exists and the event type equals one of the four event types (i.e. "fail", "finish", "kill", "lost") which show the completion status, the tasks execution time is calculated by the time elapsed between two timestamps. The record is inserted into the mongodb LA database. Google published the tracelog as csv format with semicolon delimeter. It is converted to the excel format. MongoDb has a support to migrate data from excel to itself. These data have been migrated from excel to MongoDB by this feature. The database consists of 8352 tables and each one includes jobs-related information which is submitted in 5 minutes intervals. Therefore, there are 288 tables for a single day. Each table consists of submissionTime, jobId, taskId, executionTime, CPUReq columns. Execution time is kept in minutes.

In the beginning of the integration of google tracelog, number of submitted jobs in 1 hour interval is used as similar with the integration of ITU IT Departmant workload. Same as the previous work, random VM is selected to represent each job which is submitted in the interval. The forecasting module is responsible for prediction of the number of jobs will be submitted in the next period. It is shown in Figure 4.9.

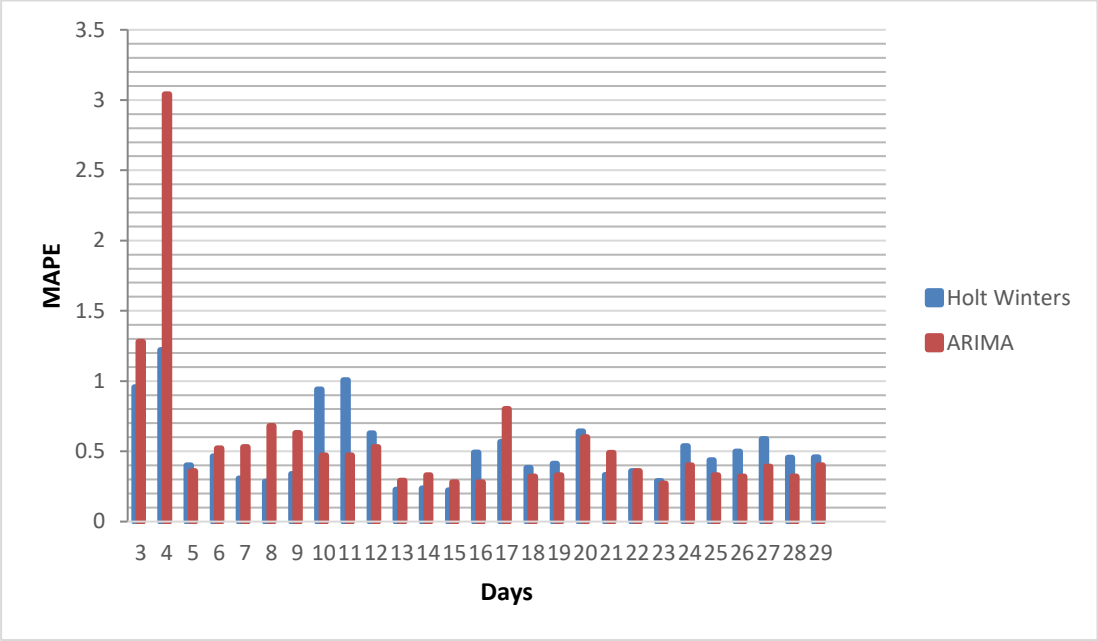

Figure 4.9 : MAPE results of Holt Winters and ARIMA on Google Tracelog data with 1 hour interval.

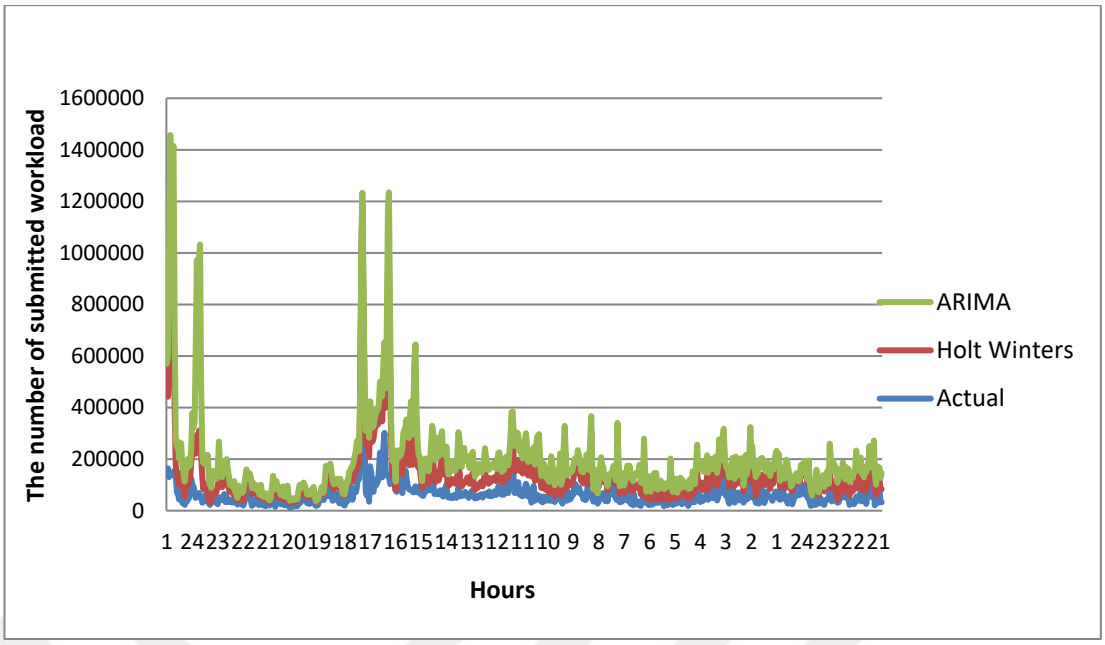

Figure 4.10 : The number of submitted workload results of Holt Winters and ARIMA on Google Tracelog data within 1 hour interval.

Then, it is decided to use job execution time and CPU requirement which are already known in Google tracelog to make the analysis of the proposed allocation algorithm more accurate. Forecasting module is responsible for estimating PR_{total} for the next 1 hour interval. Processing requirement of a job is calculated by multiplying the execution time and CPU requirements of the job. Total processing requirement means that the sum of the processing requirements of jobs in the same 1 hour interval.

Then, it is decided that 1 hour is so large window for monitoring and forecasting for this case. Because, keeping the server in the idle state during 1hour causes energy wastage. On the other hand, if the required energy for switching a server on/off is more than the required energy for running the server at idle state for a period, then keeping the server in an idle state during that period is more energy efficient. According to the spec of the selected server in this research, the monitoring interval is 5 min. However, when 5 min interval is used instead of 1 hour, since PR_{total} does not have seasonal patterns and trends, forecasting methodologies, such as Holt Winters, ARIMA, support vector regression and nonlinear regression, give results with a high error rate. Therefore, the APR_{total} are used to calculate Nr. APR_{total} uses the mean value of both the CPU requirement and execution time instead of exact values. The noise of the PR_{total} time series is filtered by using the mean value of the parameters. APR_{total} is calculated by multiplying mean execution time of submitted workloads (shown in

Figure 4.11), submitted workload (shown in Figure 4.12), and mean CPU requirements of submitted workloads (shown in Figure 4.13) in intervals. Holt Winters is applied to the workloads by using excel solver add-in while ARIMA is applied by using Minitab.

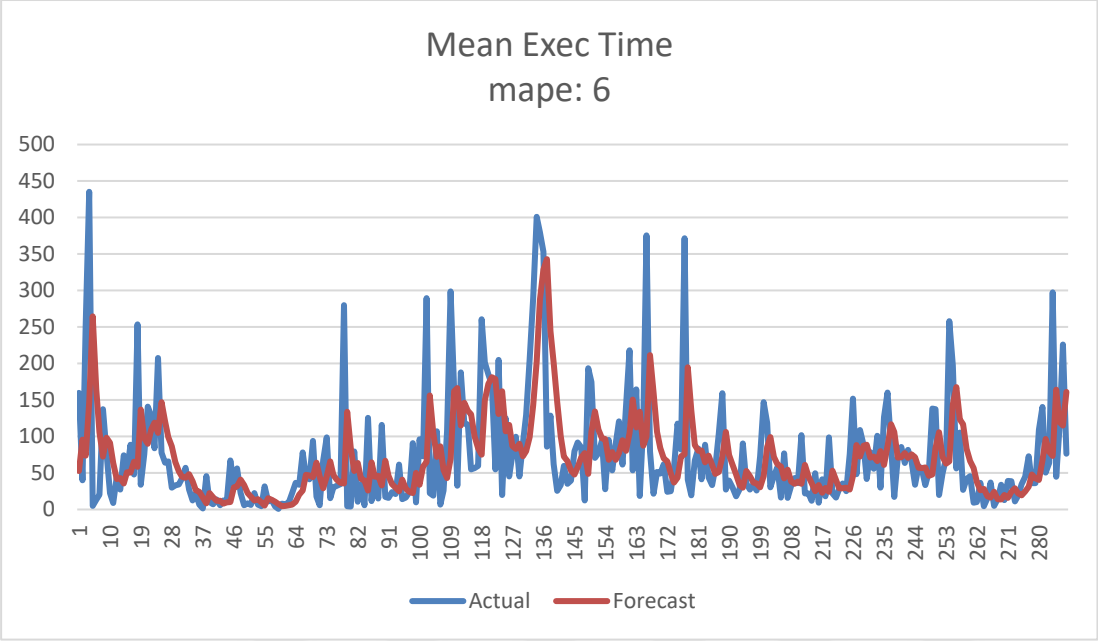

Figure 4.11 : Mean execution time Holt Winters result – alpha: 0.39, beta: 0.022 and gama: 1.

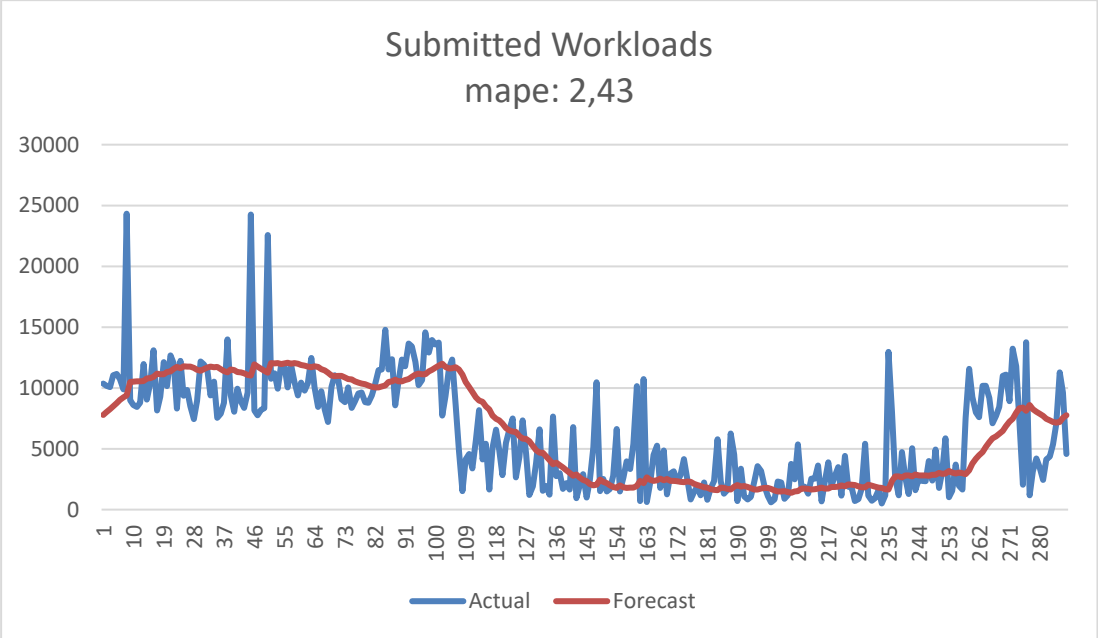

Figure 4.12 : Submitted workloads Holt Winters result – alpha: 0.06, beta: 0.03 and gama: 0.29.

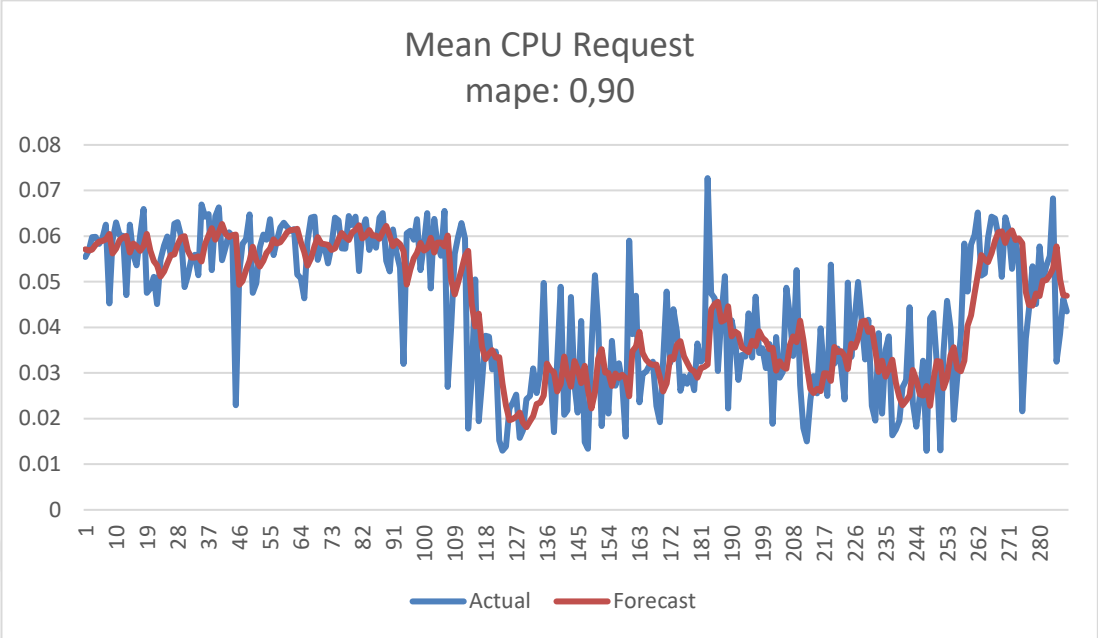

Figure 4.13 : Mean CPU Request Holt Winters result – alpha: 0.29, beta: 0.01 and gama: 0.

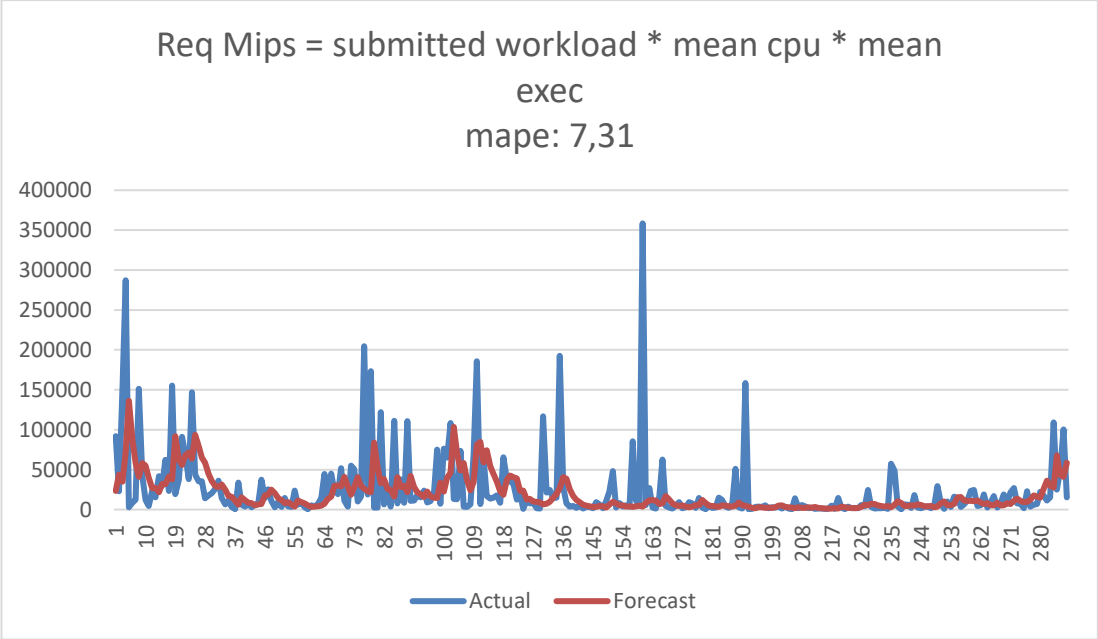

Figure 4.14 : Mean computation requirement result of Holt Winters.

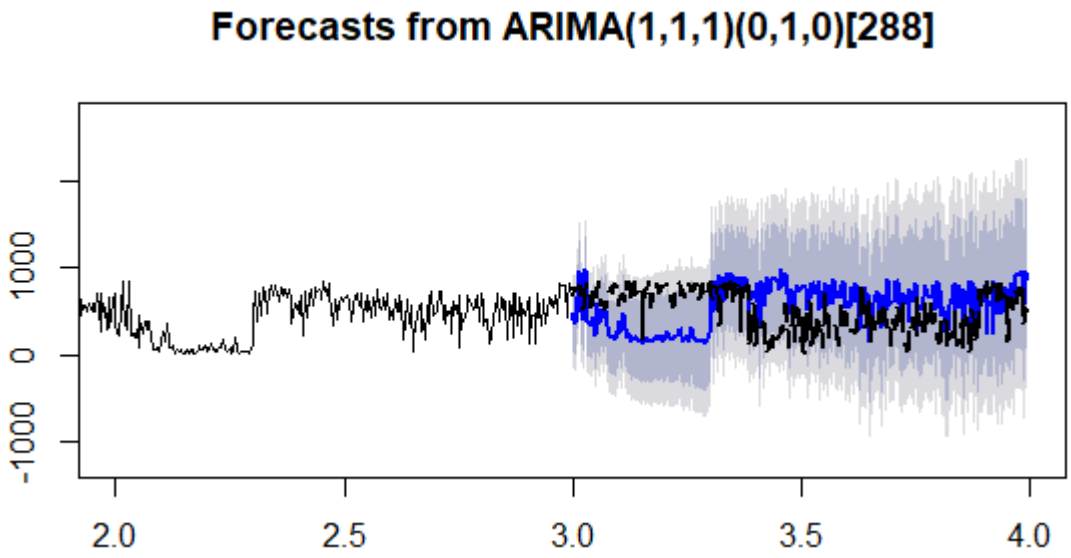

Figure 4.15 : Mean CPU request ARIMA result – MAPE: 68.89.

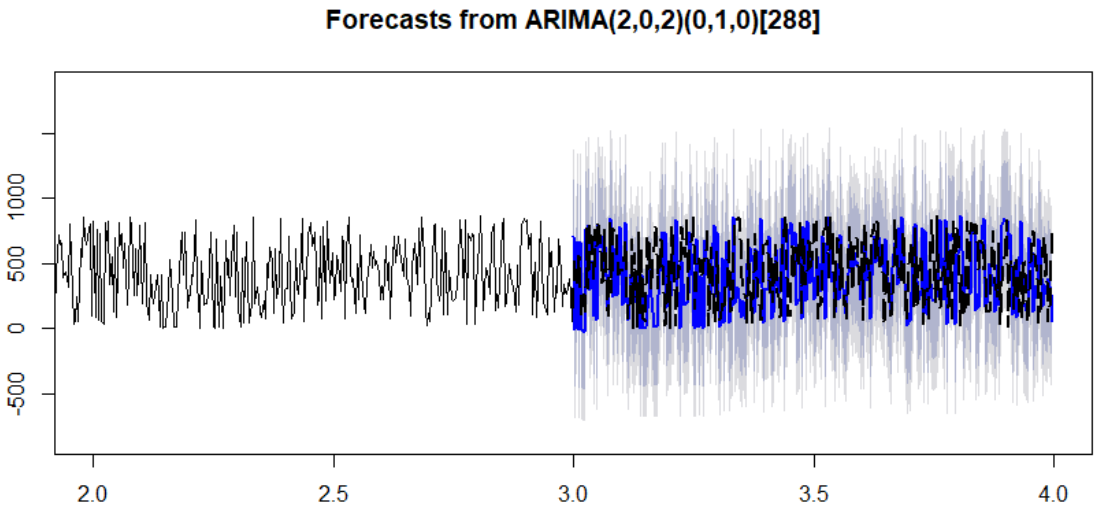

Figure 4.16 : Mean execution time ARIMA result – MAPE: 168.

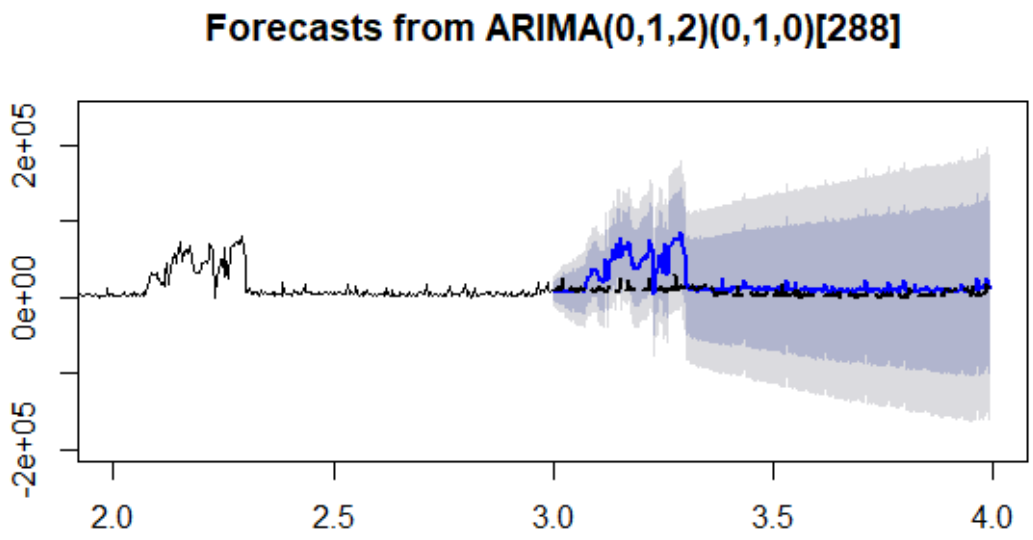

Figure 4.17 : Submitted workload ARIMA result – MAPE: 142.

5. RESULTS ANALYSIS

This chapter expresses the results details under two subsections; performance metrics and simulation results and analysis. One of the important points underlined in this thesis, migration causes extra energy consumption due to it is time and resource consuming process. So that, during the calculation of energy consumption in a datacenter, the number of migrations has a portion. In addition, the another important point is avoiding unnecessary hosts shutdown to prevent energy wastage. Minimizing energy consumption is the main goal of this thesis. Therefore, it is one of the key performance indicator of this thesis. Lastly, the number of received workloads within the simulation time is another metric for evaluation of LAA's performance.

To evaluate energy efficiency results of LAA, LR-MMT is selected for comparison since it is the best approach provided by CloudSim based on VM migration. LAA provides a comprehensive appoach consists of different modules. Prediction module is one of them. It is based on HW methodology. While evaluating the performance of LAA, it is handled as two separated models; LAA-HW and LAA-O. LAA-O indicates the optimal with using the actual input instead of the forecasted value. Moreover, to be ensure about the robustness of the proposed approach, sensitivity analysis is made by increasing execution time of submitted workloads. All the metrics used for evaluation and results analysis have been explained with details in following subsections.

5.1 Performance Metrics

The number of Hosts Shutdown (NoHS), Number of Received Workloads (NoRW), Energy Consumption (EC) and Number of Migrations (NoM) are used as key performance metrics in the migration model to which we compare our proposed approach. Our approach does not contain a migration step. Therefore, the number of VM migrations becomes zero in this approach. The number of active servers is less than the migration model according to the experimental results. If there is already a place on an active server, then a new server will not be awakened. Moreover, energy consumption due to the number of active servers and the number of unnecessary migrations can be reduced through the proposed approach.

5.2 Simulation Results and Analysis

The forecasting module is based on Holt Winters, as described before. Therefore, the approach is named LAA-HW. If we knew the actual values instead of forecasted values, then the system would give the optimal result as LAA-O. The proposed model uses N_a and N_r parameters to decide whether the system has more active servers than required. If N_a is greater than N_r, then incoming workloads are allocated on already active servers. The mean CPU requirement of a day and the mean execution time of a day in the Google Trace logs are 3% and 1,13 min, respectively, leading to a small Nr value and causing less received workloads in LAA compared to LR-MMT. These requirements cause bottlenecks for workloads with short execution times and fewer CPU requirements. The number of migrations is zero in our approach. The energy consumption for switching on/off in our model is less than that in the migration model. As described in section 3, the energy consumption is the sum of required energy for computing the workload, migration and switching on/off decision. Each migration consumes 0,05 kwh [50]. Each E_OO consumes 0,019 kwh according to the lab experiments. The comparison results are shown in Table 5.1. LR-MMT does not have a forecasting module. The host shutdown decision is made by considering only the system's current state. Therefore, the number of hosts shut down in LR- MMT is greater than in any type of LAA. LAA-O gives the best result in terms of energy consumption, and it also gives better results for NoRW than LAA-HW. This is because LAA-O knows the incoming CPU requirements of the workload, instead of forecasted the values belonging to the workloads.

Table 5.1 : Comparison results.

Algorithm	NoRW	EC (kwh)	NoM	NoHS
LR-MMT	34838	3344,26	6036	8563
LAA-HW	23945	1300,92	-	3125
LAA-O	26538	1200,59	-	2716

To ensure the robustness of the proposed algorithm, sensitivity analysis is performed by increasing the execution time by multiplying by 10, 100 and 1000. According to the results, the experiment with the longest execution time gives the best result since a longer execution time requires a higher server open time and a higher number of active servers. The longest execution time overcomes the bottleneck of the small value of N_r.

LR-MMT gives a similar number of received workloads under different execution time values, unlike LAA-HW and LAA-O, as seen in Figure 5.1. The numbers of received workloads by both LAA-HW and LAA-O are increased under extended execution time since longer execution times require a greater number of active servers.

Figure 5.1 : Sensitivity analysis results for number of received workloads.

In Figure 5.2, the mean energy requirement to complete a workload is shown. The mean energy requirement is calculated by dividing the total energy consumption by the number of received workloads. LAA gives the optimal result with workloads of approximately 10 minutes since both high throughput and energy efficiency can be provided without turning a new server on. When the workloads remain as they are, the forecasted Nr value is small. Because of the lower number of active servers, the received number of workloads is low. On the other hand, the increase in the execution time of workloads causes an increase in the number of active servers. However, the proposed algorithms are based on the remaining time of already running workloads. A long execution time means a longer remaining time at any time. It is difficult to find the appropriate server from active servers with the least remaining time algorithm. Therefore, the only option is turning a new server on to meet the requirements, which gives rise to the need for an increment of the number of servers. With the increment of the number of active servers, the number of received workloads is also increased, but it causes significant energy consumption per workload since energy consumption and the number of active servers is directly proportional.

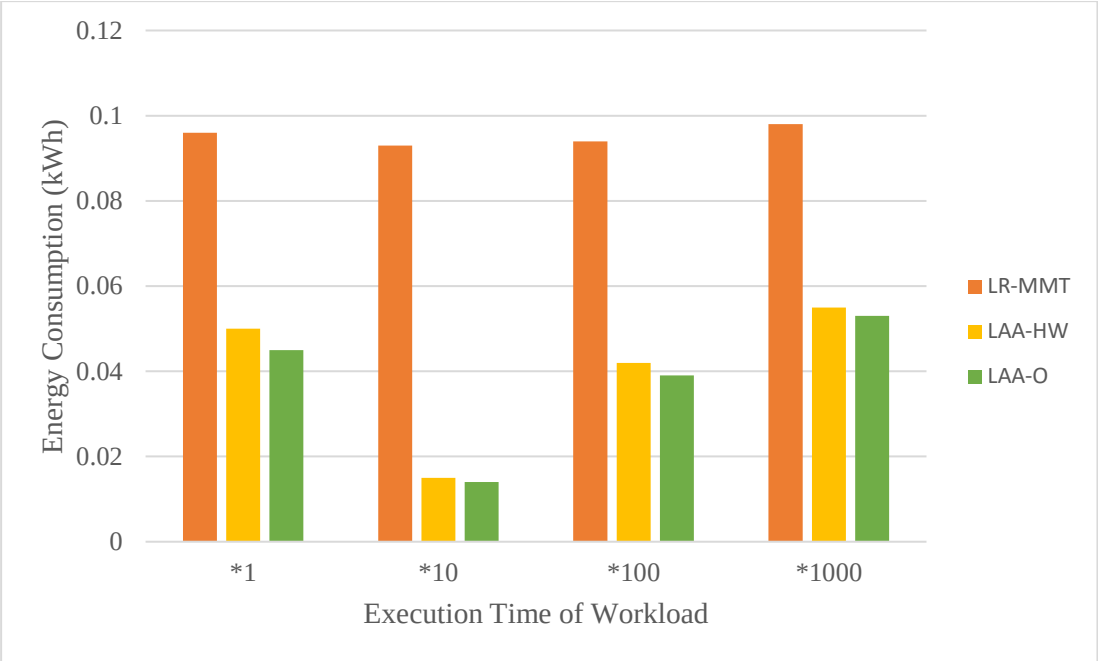

Figure 5.2 : Sensitivity analysis results for mean energy consumption per workload.

In addition, the experiment was conducted with first fit, best fit and worst fit algorithms to make comparisons in terms of the execution time for the same workloads. These algorithms are implemented as nonpower aware algorithms (NPAs). These algorithms consider CPU utilization requirements. First fit allocates the workload on the first hole that is large enough by scanning from the beginning. Best fit allocates the workload on the smallest hole that is large enough and produces the smallest leftover hole. Worst fit allocates the workload on the largest hole that is large enough and produces the largest leftover hole, which may be more useful than a leftover hole from a best fit approach. Figure 5.3 shows the mean execution time results in milliseconds belonging to the related algorithms to find appropriate place among search pool for single workload. It does not show the time the entire simulation takes to complete whole workloads. The first fit algorithm is the fastest algorithm among these three algorithms because it searches as little as possible. The best fit and worst fit algorithms show similar results, almost 25 times slower than the first fit algorithm. The NPA algorithms – first fit, best fit and worst fit and LR-MMT – started with 800 active hosts in the experiments. Then, according to the system's trend, the number of active hosts was decreased by shutting idle hosts down. LR-MMT searches for the most appropriate host among a host list that does not contain overutilized servers.

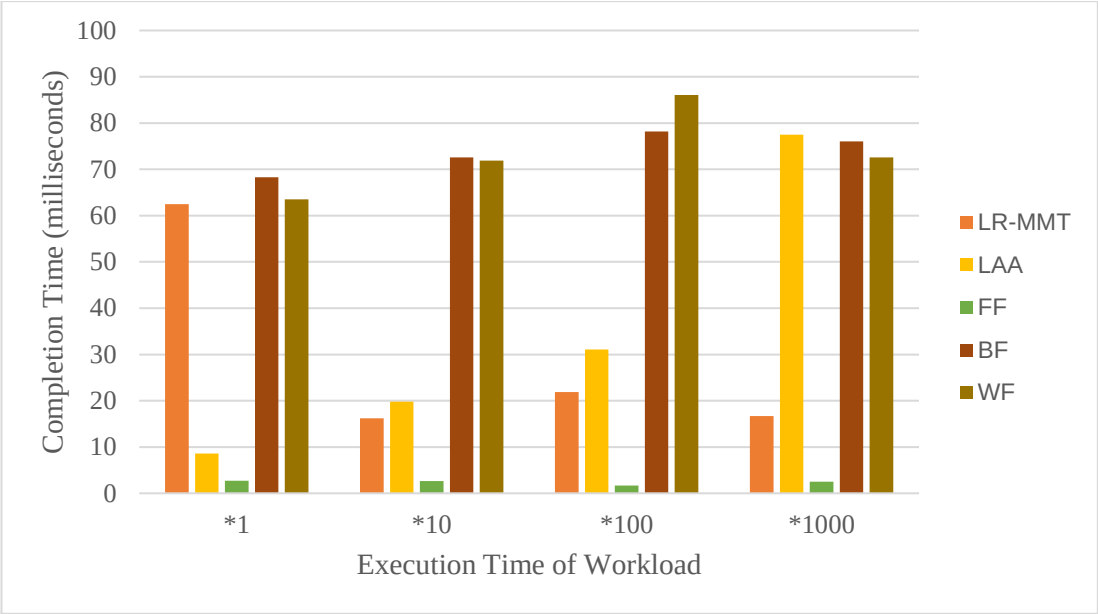

Figure 5.3 : Sensitivity analysis results for mean execution time of algorithms.

Therefore, the execution time of the LR-MMT algorithm is affected by both changes in workloads on servers and the number of active servers. If the number of active servers decreases, then the search pool decreases. The results of the sensitivity analysis show that when the execution time of workloads increases, the number of running workloads in parallel increases, which causes overutilization and decreases the size of the search area and execution time of the search algorithm. It starts with similar performance to the best fit and worst fit algorithms, and the execution time of the algorithm decreases to one-third as a result of the increased execution time of workloads with sensitivity analysis. The execution times of the NPA algorithms remain similar during the simulation period because the search area consists of all active servers. The LAA algorithm starts with 800 passive hosts and has discrete sets as the overloaded server list and the underutilized server list with a value of 0 at the beginning of the simulation. Based on the system's trend, shrinking, or enlarging, the search algorithm starts with the related server list or the passive or underutilized server list. The overloaded server list is only used for exclusion, which means that the search area becomes larger as the number of underutilized servers' increases. The LAA starts with 4 times the execution time of the first fit algorithm, but with the increasing number of underutilized servers, it starts to give similar results to the best fit and worst fit algorithms. As explained in the problem definition section, the monitoring interval is selected as 5 mins. The mean execution time of a day in Google tracelog is 1.13 min which is smaller than the monitoring window. It means existing workloads are

completed until next monitoring time comes. Therefore, Na and Nr has small values during the simulation time and the search pool consists of a small number of resources. If the lengths of the workloads are longer than the monitoring interval, then Na and Nr are increased and so does the completion time of the algorithm is increased. Approximately 10 minutes lengths workloads which LAA gives the optimal results of both energy efficiency and throughput with, also gives acceptable result in term of the completion time of the algorithm.

6. CONCLUSIONS AND FUTURE WORKS

Cloud Computing provides services to the customers through service providers while ensuring Service Level Agreements. Services are mainly divided into three categories; Software as a Service, Platform as a Service and Infrastructure as a Service. Cloud Computing brings several advantages to the users such as isolating them from resources maintenance and taking the responsibility of owning resources from them. Moreover, thanks to pay-as-you-go model, the cost is reduced. Users can access resources from anywhere via the internet connection. According to the users demand, resources can be easily added and removed. Cloud computing offers easy access to the resource highly scalability. Beside these all advantages of the cloud computing, there are some open challenges which have been investigated in recent such as increases in CO2 emission rate, IP traffic and energy consumption, negative effects on global warming.

Energy efficiency in cloud environments has received significant attention in the past few years because of the increasing usage of system resources with developing technology and decreasing prices. Energy efficiency can be achieved in different levels. In this thesis scope, energy efficiency solutions are divided into three main categories; Hardware, Consolidation and Workload Aware Resource Scheduling. Researches in the literature are investigated based on these three categories.

Due to the improvement on hardware technologies, more efficient hardware devices can be used to reduce energy consumption. Via the governors, frequency and voltage can be configured to save power or supply performance. Beside that switching off servers, sleep state is provided as an other option to save power during the idle state of the server. In a nutshell, energy efficiency can be ensured by hardware companies with manufacturing more energy efficient hardware devices. In addition, according to the deadline requirement of jobs, dynamic voltage and frequency, resource throttling and sleep states are another options to be applied to save power.

Consolidation, second category of the taxonomy of energy efficiency techniques is divided into two sub categories; allocation and migration. The main goal of consolidation is reducing the number of active servers to reduce the energy consumption. Consolidation can be achieved in both allocation and migration. During the allocation, server is selected from active servers to host workloads if available, to

prevent increasing the number of active servers in the system and to increase the utilization of resources. Migration is among servers in which workloads are already running. Server consolidation through migrating workloads into a small set of servers and switching idle servers off is a commonly used technique to save energy. Migration is a process of replacement VMs from the source hosts to target hosts to reduce the number of active servers. VM migration also consumes energy and causes execution delays since time is needed for VM migration. Frequently migration causes network vibration. The impact of migration on energy consumption can be evaluated with two aspects; energy consumption during migration transition period and energy consumption during VM migration further. Remaining execution time of VMs is an important criteria to take into account while deciding the migration. Running the VM on the same server consume less energy that to migrate it to another server when the remaining execution time of the VM is less than the time needed for the migration and execution after the migration. Consolidation is an important technique to reduce the energy consumption but it alone is not enough. Migration may cause extra energy consumption in some cases. Migration consumes energy and causes execution delays since time is needed for VM migration. Therefore, it should be avoided to decide unnecessary migration. Moreover, the aim of the consolidation with migration is reducing energy by switching the idle servers after the migration process. The decision of switching off servers should not be taken by considering the only system's current state but not the possible future demand. When only the system's current state is considered under the enlarging condition, it may cause unnecessary switching off process. Therefore, extra energy is consumed to switch them on again to be ready for newly arrived workloads.

The last category of the taxonomoy of the energy effiency techniques is Workload Aware Resource Scheduling. Researches under this category aims to find the most suitable server for allocation of workload based on the characteristics of workloads such as deadline, required resource types. Researches in this scope may use consolidation techniques or DVFS to reduce energy consumption. Energy saving is the primary goal compared to the throughput for the workloads without strict deadline. For this case, workloads can be consolidated in fewer servers via the consolidation techniques. In another example, when the workloads are directed acyclic graph, a job which have predecessor jobs cannot be executed until predecessor jobs are finished.

DVFS is an appropriate solution for different execution time of predecessors. Through adjusting voltage and frequency, the job with shorter execution time can be finished at the same time with the job with longer execution time. Thus, idle period is shortend and power consumption can be reduced. Therefore, it is important to analyze the system's requirements and the main expectation of the system (namely; energy, performance, deadline, throughput). According to the resources and workloads, it is aimed to find the most appropriate solution to meet the actual requirement.

Datacenters account for 1% of the world's electricty consumption. Servers are the components of datacenter which have main portion of electricity usage in datacenter. To handle energy inefficiency in server level, one of the most common resource provisioning approaches is based on running servers at an optimal utilization rate. Resource and energy efficiency should be ensured at the allocation of workloads on servers while considering the performance loss. There are researches shows that running servers at fully loaded causes performance degradation. It is valid for all types of resources of a server in different rates of usages. For example, if the disk is running at more than 50% of the capacity, it causes the performance degradation. This utilization rates of resources are considered as optimal utilization rates/thresholds. Avoiding from exceeding the optimal utilization rates of resources while allocating workloads on resources is one of the energy efficiency approaches. However, this approach alone is not enough to provide efficiency. Because, running at the optimal utilization rate may require turning a new server on to meet the requirement of incoming workloads, and it may consume more energy than is consumed due to the performance degradation from allocating the incoming workload on already active servers that are at optimal utilization rates. Remaining execution time of the workloads already running on the servers is the another important parameter during the allocation decision. When the incoming workload is allocated on the active server which is at the optimal utilization rate, the servers utilization rate exceeds the optimal utilization rate after this allocation for a while. This overutilization duration is depend on the least remaining time of workloads already running on the server. Therefore, if all active servers will be overutilized after the newly arrived workload, the server with the least remaining time will be the best candidate for the allocation since it causes over utilization for the least duration. Switching off idle servers is another approach to save energy. So that, remaining time of already running workloads on a server which will

be switched off is again an important criteria to decide allocation. The selection of the server with the highest remaining time for the existing workloads makes it possible to overlap the execution time of the newly arrived workload with it. Therefore, the newly arrived workload is expected to be completed after the completion of the existing workloads in the worst case. Actually, all these mentioned approaches can be evaluated under the consolidation. It is aimed to use the optimal number of resources with optimal utilization rates to meet users' requests while considering performance degradation. Switching off idle servers save energy when these servers are not required for a satisfied duration. When the server is idle state, the common approach is switching of this server to save energy. However, beside that the current state of the server, future demand also is considered. If the system trend is enlarging, it means the number of existing active servers are not enough to meet incoming requests and additional servers are required. Because of additional active servers requirement, switching off the idle server is an unnecessary process to cause energy overhead and time consuming. The proposed model consists of forecasting module, allocation module and monitoring module.

To predict the future demand of the system, 5 minutes monitoring window is used. Monitoring window is decided based on the server's power spec. Two different forecasting approaches are applied to the selected tracelog. These are Holt Winters and ARIMA. Holt Winters technique is the suitable forecasting methodology for time series data exhibits seasonality and trending pattern. ARIMA describes time series with its historical values and probabilistic error term. It gives three parameters p,q,I to expressed past values will be used for auto regressive and moving average and if the data shows non-stationary, stationary parameter respectively. ITU IT Department Web Site User Entrance data is used to analyze the performance of the proposed allocation algorithm. ITU IT Department time series consists of the number of user entrance within 1 hour interval during 3 days period with 1 day cycle. To simulate the workload's resource requirement, random type of VM is created for each user entrance. To make prediction more accurate, more data with the longer period than 3 days is required. Therefore, the google cluster tracelog is decided to be used. The number of submitted workload within 1 hour interval during 29 days period with 1 day cycle is used. Holt Winters gives better result than ARIMA for both ITU IT Department data and Google Tracelog. Then, CPU requirement and execution time of

jobs are also taken into consideration. Monitoring window is changed to 5 minutes from 1 hour. However, the obtained time series have noises to make the prediction hard. Therefore, it is decided to make prediction for three separate time series instead of single time series, namely; number of submitted workloads, mean execution time and mean CPU requirement of jobs within same window.

In this thesis, the proposed model focuses on the open issues of the research in the literature. VM migration and the unnecessarily switching of servers on/off cause additional energy consumption. In addition, completing a workload with the migration overhead consumes more energy since it takes more time. Therefore, to avoid migration by optimizing the placement of new requests well and to avoid unnecessarily switching servers on/off, the proposed approach uses prediction methodology. Holt Winters is preferred as a forecasting technique because of its suitability to time series. Furthermore, most approaches in the literature propose using resources at the optimal utilization rate since over- and underutilization cause energy inefficiency. However, this means increasing the number of required resources, leading to more energy consumption. Based on this motivation, we propose an adaptive approach for VM placement without VM migration. The most important contribution is preventing VM migrations while considering the remaining time of running workloads. Optimum utilization is a significant factor in providing energy efficiency. However, even if its load will exceed the optimum utilization rate, allocation of a workload to an already active host instead of allocation of the workload to a new server should be preferred under some circumstances. We propose an adaptive decision-making approach to energy efficient allocation without migration. We consider not only history but also future demands and the remaining time of running workloads. To determine the systems behavior for workloads with longer execution times, sensitivity analysis is performed. The real execution times of workloads are extended by multiplying by 10, 100 and 1000 to perform the sensitivity analysis. A short execution time means a lower processing requirement and a smaller value for Nr. However, it causes bottlenecks and fewer received workloads than with LR-MMT. When the execution times of workloads are increased, Nr is increased. Thus, the number of received workloads is increased compared to LR-MMT. In this manner, the energy consumption decreases with the proposed algorithm and VM migration overheads are avoided. In addition, the

system performance in terms of energy efficiency and throughput is better than LR-MMT for longer execution times.

In this thesis, we focus on CPU intensive workloads. We plan to extend the proposed approach to work with workloads based on other system resources apart from CPUs, such as networking components, I/O devices and storage.

REFERENCES

[1] **Zhang, Q., Cheng, L., & Boutaba, R.** (2010). Cloud computing: state-of-the-art and research challenges. *Journal of internet services and applications*, *1*(1), 7-18.

[2] **Kaur, T., & Chana, I.** (2018). GreenSched: An intelligent energy aware scheduling for deadline-and-budget constrained cloud tasks. *Simulation Modelling Practice and Theory*, *82*, 55-83.

[3] **Masanet, E., Shehabi, A., Lei, N., Smith, S., & Koomey, J.** (2020). Recalibrating global data center energy-use estimates. *Science*, *367*(6481), 984

[4] **Siddik, M. A. B., Shehabi, A., & Marston, L.** (2021). The environmental footprint of data centers in the United States. *Environmental Research Letters*, *16*(6), 064017.

[5] **Shehabi, A., Smith, S., Sartor, D., Brown, R., Herrlin, M., Koomey, J., ... & Lintner, W.** (2016). United states data center energy usage report, Berkeley Lab, URL< https://www.osti.gov/servlets/purl/1372902>

[6] **Koot, M., & Wijnhoven, F.** (2021). Usage impact on data center electricity needs: A system dynamic forecasting model. *Applied Energy*, *291*, 116798.

[7] **Beloglazov A, Buyya R** (2012) Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers. Concurrency and Computation: Practice and Experience. Published online in Wiley InterScience, DOI: 10.1002/cpe.1867; 24:1397-1420.

[8] **Fan X, Weber WD, Barroso LA (**2007) Power Provisioning for a Warehouse-Sized Computer. ACM 24 SIGARCH computer architecture news, pp. 13-23.

[9] **Hsu CH, Slagter KD, Chen SC, Chung YC** (2014) Optimizing energy consumption with task consolidation in clouds. *Information Sciences*, *258*, 452-462

[10] **Galloway JM, Smith KL, Vrbsky SS** (2011, October) Power aware load balancing for cloud computing. In Proceedings of the World Congress on Engineering and Computer Science Vol. 1, pp. 19-21

[11] **Lee YC, Zomaya A.Y.** (2012) Energy efficient utilization of resources in cloud computing systems. *The Journal of Supercomputing*, *60(2)*, 268-280

[12] **Dabbagh M, Hamdaoui B, Guizani M, Rayes A** (2015). Energy-efficient resource allocation and provisioning framework for cloud data centers. IEEE Transactions on Network and Service Management, 12(3), 377-391

[13] **Dhiman G, Marchetti G, Rosing T** (2010) Vgreen: A system for energy-efficient management of virtual machines. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 16(1), 6

[14] **Calheiros RN, Ranjan R, Beloglazov A, De Rose CAF, Buyya R** (2011) CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and experience*; 41.1: 23-50

[15] **Lee, Y. C., Karunaratne, D. T., Wang, C., & Zomaya, A. Y.** (2012). Energy-efficiency models for resource provisioning and application migration in clouds. In *Cloud Computing: Methodology, Systems, and Applications* (pp. 369-388). CRC Press.

[16] **Fan, X., Weber, W. D., & Barroso, L. A.** (2007). Power provisioning for a warehouse-sized computer. *ACM SIGARCH computer architecture news*, *35*(2), 13-23.

[17] **Pallipadi, V., Li, S., & Belay, A.** (2007, June). cpuidle: Do nothing, efficiently. In *Proceedings of the Linux Symposium* (Vol. 2, pp. 119-125). Citeseer.

[18] **Pallipadi, V., & Starikovskiy, A.** (2006, July). The ondemand governor. In *Proceedings of the linux symposium* (Vol. 2, No. 00216, pp. 215-230).

[19] **Orgerie, A. C., Assuncao, M. D. D., & Lefevre, L.** (2014). A survey on techniques for improving the energy efficiency of large-scale distributed systems. *ACM Computing Surveys (CSUR)*, *46*(4), 1-31.

[20] **Lee, Y. C., & Zomaya, A. Y.** (2010). Energy conscious scheduling for distributed computing systems under different operating conditions. *IEEE Transactions on Parallel and Distributed Systems*, *22*(8), 1374-1381.

[21] **Ferreto TC, Netto MA, Calheiros RN, De Rose, CA.** (2011) Server consolidation with migration control for virtualized data centers. *Future Generation Computer Systems*, *27*(8), 1027-1034

[22] **Khan AA, Zakarya M, Khan R, Rahman IU, Khan,M** (2020) An energy, performance efficient resource consolidation scheme for heterogeneous cloud datacenters. *Journal of Network and Computer Applications*, *150*, 102497

[23] **Dhiman, G., Mihic, K., & Rosing, T.** (2010, June). A system for online power prediction in virtualized environments using gaussian mixture models. In *Proceedings of the 47th design automation conference* (pp. 807-812).

[24] **Hsu, C. H., Slagter, K. D., Chen, S. C., & Chung, Y. C.** (2014). Optimizing energy consumption with task consolidation in clouds. *Information Sciences*, *258*, 452-462.

[25] **Li X, Qian Z, Chi R, Zhang B, Lu S** (2012) Balancing resource utilization for continuous virtual machine requests in clouds. In Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), *2012 Sixth International Conference on IEEE*, pp. 266-273

[26] **Chen J, Du T, Xiao G** (2021). A multi-objective optimization for resource allocation of emergent demands in cloud computing. *Journal of Cloud Computing*, *10*(1), 1-17

[27] **Cao J, Wu Y, Li M** (2012, May) Energy efficient allocation of virtual machines in cloud computing environments based on demand forecast. *In International Conference on Grid and Pervasive Computing Springer*, Berlin, Heidelberg, pp. 137-151

[28] **Srikantaiah S, Kansal A, Zhao F** (2008) Energy aware consolidation for cloud computing. *Proceeding of the 2008 conference on Power aware computing and systems*

[29] **Beloglazov A, Buyya R** (2010) Energy Efficient Resource Management in Virtualized Cloud Data Centers. *10th IEEE/ACM International Conference on Cluster Cloud and Grid Computing*

[30] **Zhou Z, Abawajy J, Chowdhury M, Hu Z, Li K, Cheng H, Li F** (1 2018) Minimizing SLA violation and power consumption in Cloud data centers using adaptive energy-aware algorithms. *Future Generation Computer Systems*, 86, 836-850

[31] **Ruan X, Chen H, Tian Y, Yin S** (2019) Virtual machine allocation and migration based on performance-to-power ratio in energy-efficient clouds. *Future Generation Computer Systems*, 100, 380-394

[32] **McGough, A. S., Forshaw, M., Gerrard, C., Robinson, P., & Wheater, S.** (2013). Analysis of power-saving techniques over a large multi-use cluster with variable workload. *Concurrency and Computation: Practice and Experience*, *25*(18), 2501-2522.

[33] **Liu, N., Dong, Z., & Rojas-Cessa, R.** (2012, August). Task and server assignment for reduction of energy consumption in datacenters. In *2012 IEEE 11th International Symposium on Network Computing and Applications* (pp. 171-174). IEEE.

[34] **Beloglazov A, Buyya R** (2010) Energy Efficient Allocation of Virtual Machines in Cloud Data Centers. *10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*

[35] **Verma A, Ahuya P, Neogi A** (2008) pMapper: power and migration cost aware application placement in virtualized systems. *Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware*, Springer-Verlag New York, Inc., p. 243-26

[36] **Duan H, Chen C, Min G, Wu Y** (2017) Energy-aware scheduling of virtual machines in heterogeneous cloud computing systems. *Future Generation Computer Systems*. 2017; 74: 142-150

[37] **Tian W, He M, Guo W, et al** (2018) On minimizing total energy consumption in the scheduling of virtual machine reservations. *Journal of Network and Computer Applications.* 2018; 113: 64-74

[38] **Zhang, Y., Cheng, X., Chen, L., & Shen, H.** (2018). Energy-efficient tasks scheduling heuristics with multi-constraints in virtualized clouds. *Journal of Grid Computing*, *16*(3), 459-475.

[39] **Yigitbasi, N., Datta, K., Jain, N., & Willke, T.** (2011, December). Energy efficient scheduling of mapreduce workloads on heterogeneous clusters. In *Green Computing Middleware on Proceedings of the 2nd International Workshop* (p. 1).

[40] **Toosi, A. N., Sinnott, R. O., & Buyya, R.** (2018). Resource provisioning for data-intensive applications with deadline constraints on hybrid clouds using Aneka. *Future Generation Computer Systems*, *79*, 765-775.

[41] **Rodriguez, M. A., & Buyya, R.** (2014). Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds. *IEEE transactions on cloud computing*, *2*(2), 222-235.

[42] **Chowdhury, M. R., Mahmud, M. R., & Rahman, R. M.** (2015, June). Study and performance analysis of various VM placement strategies. In *2015 IEEE/ACIS 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)* (pp. 1-6). IEEE.

[43] **Chatfield, C., Yar, M.,** Holt Winters forecasting: some practical issues, *Journal of the Royal Statistical Society.* Series D (The Statistician), 1988.

[44] **Goodwin, P.**, The holt-winters approach to exponential smoothing: 50 years old and going strong, Foresight pp. 30–33, 2010.

[45] **Hyndman,R.J. ve Athanasopoulos, G.**, "Forecasting: Principles and Practice", Published By Otexts.com, 2014

[46] **Pankratz, A.**, Forecasting With Univariate Box-Jenkins Models Concepts and Cases, *John Wiley & Sons*, Inc., 1983.

[47] URL <http://www.minitab.com/en-us/products/minitab/quick-start/>

[48] **Swanson, D.A., Tayman,J. ve Bryan, T.M.**, "MAPE-R: a rescaled measure of accuracy for cross-sectional subnational population forecasts*", Journal of Population Research Springer*, 2011.

[49] Google cluster-usage traces: format and schema. https://github.com/1google/cluster-data. Published May 6, 2013. Updated October 17, 2014. Accessed June, 2015

[50] **Voorsluys W, Broberg J, Venugopal S, Buyya R**, Cost of virtual machine live migration in clouds: A performance evaluation. *In IEEE International Conference on Cloud Computing.* Springer, Berlin, Heidelberg.2009; pp. 254-265

CURRICULUM VITAE

Name Surname **:** İlksen ÇAĞLAR

EDUCATION **:**

- **B.Sc.** **:** 2008, Sakarya University, Engineering Faculty, Computer Engineering Department
- **M.Sc.** **:** 2010, Ege University, Engineering Faculty, Computer Engineering Department

PROFESSIONAL EXPERIENCE AND REWARDS:

- 2019-2023 General Electric – Staff Software Architect
- 2018-2019 General Electric – Staff Software Engineer
- 2017-2018 General Electirc – Senior Software Engineer
- 2014-2017 TÜBİTAK BİLGEM – Uzman Araştırmacı
- 2011-2014 TÜBİTAK BİLGEM – Araştırmacı
- 2010-2011 Sakarya Üniversitesi – Araştırma Görevlisi

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **ÇAĞLAR, İlksen;** ALTILAR, Deniz Turgay. Look-ahead energy efficient VM allocation approach for data centers. *Journal of Cloud Computing*, 2022, 11.1: 1-16.
- **ÇAĞLAR, İlksen;** ALTILAR, D. Turgay. Cloud work load prediction through different models based on time-series. In: *2017 International Conference on Computer Science and Engineering (UBMK)*. IEEE, 2017. p. 856-860.
- **ÇAĞLAR, İlksen;** ALTILAR, D. Turgay. Google İş Yükleri için Zaman Serisi Tahminleme Yöntemlerinin Karşılaştırılması, 2017, 5. Ulusal Yüksek Başarımlı Hesaplama Konferansı
- **ÇAĞLAR, İlksen;** ALTILAR, Deniz Turgay. An energy efficient VM allocation approach for data centers. In: *2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS)*. IEEE, 2016. p. 240-244.
- **ÇAĞLAR, İlksen**; ALTILAR, Deniz Turgay. Bulut Bilişimde Adaptif Kaynak Yönetimi Yaklaşımı, 2015, 4. Ulusal Yüksek Başarımlı Hesaplama Konferansı

OTHER PUBLICATIONS, PRESENTATIONS AND PATENTS:

- **ÖZCAN, I.;** BORA, Ş. A hybrid load balancing model for multi-agent systems. In: 2011 International Symposium on Innovations in Intelligent Systems and Applications. IEEE, 2011. p. 182-187.

- **ÖZCAN, İlksen;** ALTILAR, Turgay. MPI'da Uygulama Seviyesinde Aksaklığa Dayanıklılık.
- **ÖZCAN, İlksen;** BORA, Şebnem. Çok Etmenli Sistemlerde Yük Dengeleme ve Yük Paylaşımı.