Energy-efficient Resource Hyper-visioning in Private Clouds

by

Sahar Sohrabi

MSc. (Khajeh Nasir University of Technology)

BSc. (Tabriz University)

A thesis submitted to

Faculty of Science, Engineering and Technology

Swinburne University of Technology

for the degree of

Doctor of Philosophy

June 2017

to *Bahman,*

my amazingly supportive husband and best friend

whose unstinting care for me and our child

made the completion of this work possible

and *to* *Nita,*

my greatest motivation to try to be a better person

a little better day by day

Declaration

This thesis contains no material which has been accepted for the award of any other degree or diploma, except where due reference is made. To the best of my knowledge, this thesis contains no material previously published or written by another person except where due reference is made in the text of the thesis.

Sahar Sohrabi

June 2017

Acknowledgments

I would like to thank my Supervisors, Professor Yun Yang, Dr. Anthony Tang and Dr. Irene Moser.

I wish to thank my friends at Swinburne. They have been with me on this journey, in its marvelous moments and in the less beautiful ones. They have supported me both academically with research-related input and feedback and on a personal level. I thank Amin Rigi, who has accompanied me on this ride of ups and downs and made sure I never felt lonely. I thank my friend Mostafa Farshchi, who has supported me in the most difficult moments and has been a source of inspiration and optimism when I needed it the most. I thank Mrs Maheswaree Kissoon Curumsing for her insightful comments and amazing help on research and personal matters. I am thankful for the fantastic support I received from Ms Barbara Browne throughout my candidature who went above and beyond her duties to help me cross the line. I am grateful to all my friends: Maryna Vlasiuk, Shannon Pace, Yuri Mordovin, Sonia Golenkina and Iman Avazpour, and family members: Rasoul Sohrabi, Robab Seyyed-Razavi, Ghazaal Sohrabi, Sepehr Sohrabi. All of them have inspired me in many ways and have given me confidence.

Last but certainly not least, I want to express my gratitude to my husband, Bahman Nikkhahan, my best friend. Without him this achievement would have not been possible. I am delighted to share it with him.

Abstract

Cloud computing provides services on demand and has led to the establishment of large scale Cloud systems. Large scale Cloud systems entail energy concerns, which include drawbacks on the system and the environment. In a private Cloud, in particular, energy efficiency relates to the cost of providing Cloud services. First the impact of energy consumption on a private Cloud is explained. Then energy-efficient resource hyper-visioning policies and mechanisms are proposed and evaluated.

Energy consumption by Cloud systems contributes to a substantial level of energy consumption worldwide. High level of energy consumption, on one hand, adds to the cost of running the Cloud system (by adding to the energy bill), and the cost of running cooling devices because of the effect of energy consumption on thermal state. High energy consumption, on the other hand, contributes to carbon dioxide emission worldwide.

To address energy concerns in private Clouds, the resource hyper-visioning software, that arbitrates the virtualization of resources in terms of Virtual Machines (VMs), should make energy-efficient decisions. Hyper-visioning decisions include the mapping of VMs to hosts for execution and VM migration if needed. To deploy energy-efficient hyper-visioning, a set of hypotheses are made. These hypotheses relate to the development of adaptive energy-efficient mechanisms for VM mapping and VM migration using macro and micro level observation records to adaptively reduce energy consumption.

In this thesis, first an adaptive energy-efficient VM mapping mechanism is developed. The mechanism uses Bayesian Inference (BI) to associate the deployment of a VM mapping policies at a given state of the system and workload properties with VM mapping policy's total energy consumption, which is a macro level observation record. The proposed BI indicates which VM mapping policy

is more likely to result in less energy consumption given the current state of the system and workload pr operties. The mechanism adaptively switches between VM mapping policies according to the latest observation records for each VM mapping policy. The results proved that our proposed adaptive energy-efficient mechanism has total energy consumption level close to the best performing VM mapping policy, where the best VM mapping policy for the given state of the system and workload properties were not known beforehand.

VM mapping is about the initial assignment of VMs to hosts. Changes in resource requirements from VMs throughout their execution might change the state of hosts. As a result, hosts might encounter an imbalance problem, being either over-loaded (high utilization level) or under-loaded (low utilization level). Imbalance problem can be solved by migrating VMs from the imbalanced host to another. In this thesis, VM migration policies are developed that are based on micro level records (e.g. current hosts' utilization level, VM resource requests, VM memory size) to achieve energy efficiency. The results of the inclusion of VM migration proved to be effective in reducing total energy consumption compared to the situation where no migration was performed. Moreover, the proposed VM migration policies outperformed state-of-the-art VM migration policies.

VM migration can become adaptive in reducing energy consumption. We proposed VM migration mechanisms that base their inference on the latest micro level observation records, e.g. hosts' utilization level, VMs' memory size. The proposed mechanisms outperformed state-of-the-art migration heuristics in terms of total energy consumption while mean execution time was also significantly shortened.

In summary, energy concerns have driven the research into addressing the important issue of reducing energy consumption in private Clouds. Energy efficiency in private Clouds is achieved through hyper-visioning resources. The proposed

adaptive hyper-visioning mechanisms include Bayesian Inference to adaptively learn and update their decisions. The results indicated significant reduction of energy consumption while execution time was also improved.

Publications

The following publications in international conferences and journals are driven from this research.

- Sohrabi, S., and I. Moser. "A Survey on Energy-Aware Cloud." European Journal of Advances in Engineering and Technology 2.2 (2015): 80-91.
- Sohrabi, S. "The Application of Bayesian Heuristic for Scheduling in Real-Time Private Clouds." World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering 10.3 (2016): 481-488.
- Sohrabi, S., and I. Moser. "The Effects of Hotspot Detection and Virtual Machine Migration Policies on Energy Consumption and Service Levels in the Cloud." Procedia Computer Science 51 (2015): 2794-2798.
- Sohrabi, S., Tang, A., Moser, I., Aleti, A., "Adaptive virtual machine migration mechanism for energy-efficiency." Proceedings of the 5th International Workshop on Green and Sustainable Software. ACM, 2016.
- Sohrabi, S., Yang, Y., Moser, I., Aleti, A., "Energy-efficient Adaptive Virtual Machine Migration Mechanism for Private Clouds", Journal of Concurrency and Computation: Practice and Experience, DOI:10.1002/cpe.4170.

Contents

List of Algorithms

List of Figures

List of Tables

Chapter 1

INTRODUCTION

THIS thesis proposes dynamic, Bayesian based resource hyper-visioning mechanisms to address the issue of high energy consumption in private Clouds without imposing further delays on execution time. The mechanisms proposed investigate the reduction of total energy consumption through switching between available hyper-visioning policies and moving Virtual Machines (VMs) between hosts. A set of innovative mechanisms are designed and evaluated to adaptively hyper-vise resources to reduce total energy consumption. Simulations are done to evaluate the outcome of our proposed adaptive mechanisms in reducing energy consumption. The results indicate that our proposed mechanisms significantly improve total energy consumption in private Clouds.

This chapter provides a short background, problem analysis and key elements of this research. It is organized as follows. Section 1.1 covers an introduction to the energy-efficient resource hyper-visioning. Section 1.2 presents the energy problem analysis in the context of private Clouds. Section 1.3 outlines the key elements of this research: research scope, questions, hypotheses, methodology and contributions. Then, Section 1.4 gives an overview of the remainder of this thesis.

1.1 Energy-efficient Hyper-visioning in Private Clouds

U.S. National Institute of Standards and Technology (NIST) has defined Cloud computing as follows [69]: "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, application and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."

Four deployment models have been presumed for Cloud computing: Public, Private, Community and Hybrid Cloud.

Public Cloud

The cloud services are owned and provisioned by a cloud provider organization and users are public individuals and organizations. Services that Amazoon and Force.com provide are examples of a public cloud.

Private Cloud

The cloud is provisioned by an organization to provide services for multiple users such as its own business units. The cloud computing infrastructure can be owned and managed by the organization or a third party.

Community Cloud

A group of organization with common concerns, mission or requirements share cloud services. Cloud can be owned by a third party, one of the community members or a group of members.

Hybrid Cloud

If the cloud is not in a pure form of any mentioned deployments and a combination of two or three, it is called a hybrid cloud. For example, an organization provides cloud for its business units (private cloud). It also provides cloud services for other businesses in a same industry to decrease the maintenance and technology cost (community cloud).

Private Clouds have relatively limited resources compared with public Clouds. An energy efficient utilization of resources in a business's private Cloud has direct impacts on the financial trade-offs.

In private Clouds, total energy consumption can be reduced via energy-efficient resource hyper-visioning. First the importance of energy-efficient approaches should be explained. Then the factors contributing to total energy consumption are identified. Later, the effect of resource hyper-visioning on total energy consumption is explained. Resource hyper-visioning can be based on macro and micro level observation records in order to reduce total energy consumption in private Clouds.

The importance of reducing energy consumption by Cloud systems is noted as more and more Cloud systems are established. The growth of Cloud systems has resulted in the establishment of large scale data centers. Data centers are responsible for a considerable amount of energy consumption, 1.1 - 1.5% of global energy consumption in 2011 [56]. Given the continuous growth of data centers, more Cloud systems have emerged since the above mentioned report was published in 2011. In just two years, in 2013, in the United States alone, data centers were accounted for almost 91 billion kilowatt-hours of electricity usage [23] and it is estimated to reach 140 billion kilowatt-hours by 2020 [23]. High energy consumption leads to a high energy cost for the system and also a large carbon dioxide footprint. These have driven research into energy-efficient Cloud resource hyper-visioning in order to reduce energy consumption.

Total energy consumption can be calculated by adding the energy consumed by all resources in the system. Resources include: processor, memory, network and storage. CPU utilization was previously used to relate CPU utilization to energy consumption [45, 51, 84]. Chen et al. [16] reported the energy consumption by processors (host's CPU utilization) as the main contributor to energy consumption in Cloud systems. Deelman et al. [22] also confirmed this hypothesis by reporting the cost of running a scientific workload in 2008. They outlined that the computational cost (CPU utilization) outweighed the storage cost.

Energy consumption by processor has two parts: static and dynamic. Static energy consumption depends on the hardware (the processor). Dynamic energy consumption, however, is based on the frequency of the processor. Dynamic energy consumption by processor, E_d, is formulated by Kim, Beloglazov and Buyya [51] as equation 1.1. This formulation is then used in related studies [104, 9, 83, 19]. In equation 1.1 E_d represents the dynamic energy consumption, t is the time and f is the frequency of the CPU.

$$E_d = \int_0^{\frac{t}{(f/f_{max})}} C \times f_{max} \times f^2 \times t \tag{1.1}$$

Equation 1.1 represents the relation between the frequency of the processor, f, and the time, t. f and t are the main factors in calculating dynamic energy consumption, E_d, while f_{max} is the maximum frequency and C is a hardware dependent constant. The frequency of CPU is the CPU utilization indicator.

Frequency of the processor or CPU utilization is determined by the CPU cycles that are requested from a host. Because the resources in a host are virtualized, in terms of Virtual Machines (VMs), the resource requests from tasks on VMs determine the frequency of the processor or CPU utilization. The assignment of VMs to hosts is administrated by a software layer called resource hyper-visor.

Resource hyper-visioning software is a software layer responsible for assigning tasks to VMs and VMs to hosts. Resource hyper-visor controls the source sharing between VMs. Among resources, processor contributes to a major portion of energy consumption. Therefore, efficient utilization of resources, processor in particular, has become an active area of research since the advent of Cloud systems. In a private Cloud, resource hyper-visioning warrants further attention due to the relatively limited resources in private Clouds. Therefore, efficient utilization of the limited resources will have a high impact on reducing total energy consumption.

Energy-efficient VMs to hosts mapping has been researched in the field of Cloud computing [9, 30, 44, 51, 91]. Energy-aware mapping policies are evaluated against other policies on multiple simulation settings, workload and based on various measures. Nevertheless, the change of simulation settings and workload might lead to conflicting outcomes for each policy as each is expected to exhibit its strength in certain settings. So it is worth investigating the differences in the outcomes when simulation settings and workload properties are arbitrarily altered. The outcomes can be then measured in terms of energy consumption. Inclusion of other related measures can also provide a more detailed understanding of each mapping policy's outcome.

After recording the energy consumption of VM mapping policies in different states of the system, in terms of total energy consumption values (which are macro level observation records), the system can decide when to apply each policy to reduce energy consumption. That is, macro level observation records can be used to adaptively switch between the existing mapping policies based on what is learned from the outcome of each policy. It is to guarantee that each policy is used when it has previously shown strength in terms of reducing energy consumption.

Energy-efficient mapping policies aim to reduce energy consumption according to the information available about resources on the hosts and resource requests from VMs at the time of mapping. However, the VMs' resource requests can change throughout their lifetime, therefore changing available resources on hosts.

So, the mapping policies, that base their decision on the initial resource requests, might cause the system to become imbalanced.

An imbalanced problem refers to the system which has hosts that are either over-loaded or under-loaded. A host is over-loaded when it receives resource requests from its VMs that are more than what it can provide. An overloaded host's utilization is mostly in its highest level which according to the experiments on energy measurement [38, 39], even proportionally, evidences a high level of energy consumption. In addition, a high level of resource utilization in an over-loaded host is shown to increase the likelihood of hardware failures [25]. An under-loaded host, on the other hand, is kept in its active state where it serves a low level of resource requests from its VMs. An under-loaded host does not have issues serving its VMs. However, the energy it consumes is high, relatively to the resources it provides. Both over-loaded or under-loaded hosts convey an imbalanced problem in the system.

A solution to the imbalance problem is to migrate VMs from the imbalanced host to another. VM migration refers to the process of moving a VM from a host to another. When VM migration is done without stopping the execution in source and destination hosts (apart from a short time transferring the VM status), it is termed live migration [18]. VM migration and VM live migration are commonly used interchangeably.

VM migration consists of two steps: VM selection and VM re-mapping (mapping). There exist policies to select VMs for migration. They commonly use micro level record to select VMs for migration. Micro level observation records include the current details of the VMs and hosts (VMs' current resource requests, memory size and hosts' available resources) to decide about VM migration. Beloglazov and Buyya [8, 9] proposed three VM selection policies based on micro level observation records: Minimum Migration Time (MMT), Random Selection

(RS) and Maximum Correlation (MC). More energy-efficient VM selection policies can be proposed to compete with these VM selection policies to reduce total energy consumption.

Energy-efficiency can also be addressed by applying other VM migration approaches. Sallam and Li [83] proposed a five objective Bayesian game based heuristic for migrating VMs: load volume, energy consumption, thermal state, resource wastage and migration cost. They aim to find the most optimal VM to host assignment option to perform VM migration.

VM migration can be carried out adaptively, when micro level observation records set the deciding variables. Micro level observation records provide insight into the current utilization level of VMs and hosts. An adaptive VM migration is expected to outperform the existing policies. An adaptive resource hyper-visioning mechanism has the ability to learn. It updates its decision making process based on the observed outcomes of the resource hyper-visioning decisions.

The resource hyper-visioning decisions are expected to tackle the high energy consumption issue in the context of private Clouds.

1.2 Problem Analysis

High energy consumption has its drawbacks on the system and environment. The reduction of total energy consumption in private Clouds can be achieved through energy-efficient resource hyper-visioning mechanisms. First, problems driven from high energy consumption should be described. Then potential resource hyper-visioning solutions can be explained, according to either macro or micro level observation records for mapping and migration. Later, the advantages of adaptive mechanisms need to be explained.

Problems are driven by high energy consumption in Cloud. Problems include the increase in running costs due to the cost of energy required to run the system and the necessary cooling devices; and the impact on the environment by adding to the carbon dioxide emission level. Addressing these problems requires energy-efficient resource hyper-visioning mechanisms.

Energy-efficient resource hyper-visioning is a way to reduce energy consumption. One approach can be to switch between available resource mapping policies. Resource mapping policies assign VMs to hosts, therefore determining the utilization level on hosts. The utilization level then makes up the majority of the energy consumption. Macro level observation records can be defined as observing the outcome of a deployed mapping policy to decide upon the switch between them, based on the system and workload states. This approach proposes the selection of a mapping policy when it has previously presented strong reduction of energy consumption.

Reduction of energy consumption can also be achieved via resource hyper-visioning based on micro level observation records. It can be applied when the system encounters imbalance. An imbalanced system has hosts that are either over-loaded or under-loaded. An over-loaded host receives a high level of resource requests that it cannot provide. An under-loaded host, on the other hand, has low resource requests and is mostly idle. VMs can be migrated from an over-loaded host to ease the imbalance problem. Also, migrating VMs from an under-loaded host provides the system with the opportunity to switch the host off and save energy. The resource hyper-visioning approach is to select VMs for migration in a way that the imbalance problem is solved. Given that each migration instance needs to copy a VM from source host to destination host, it is desirable to solve the imbalance problem with fewer migration instances. To select a VM for migration, the decision can be made according to micro level observation records where the utilization level of VMs and hosts are the deciding factors in selecting VMs for migration.

The resource hyper-visioning solution for energy-efficiency, via either energy-efficient VM mapping or VM migration, can also be conducted based on micro level observation records of hosts' and VMs' utilization levels. Hosts' and VMs' utilization levels provide insight into the energy consumption in the system. By basing the decision making process on micro level observation records and incorporating both VM mapping and VM migrating, total energy consumption can be reduced significantly.

The reduction of energy consumption via resource hyper-visioning mechanisms can benefit significantly from becoming adaptive. Advantages of adaptive resource hyper-visioning mechanisms evolves around the volatility of the system state and resource requirements of VMs in private Clouds. VMs receive tasks from different units within a business or businesses. Diverse types of tasks, with relatively different behavior in their lifetime, are submitted to VMs. For example, an accounting unit sends tasks that have high processing requests followed by I/O requests for recording and reporting. Tasks from a control unit need a high level of I/O through its sensors, followed by recording and computations. It is then expected to respond to the sensed indicators quickly which requires a high computation and response generation. The Research & Development (R&D) units might deploy tasks of any nature, given their potential resources requirements. Although the resource requirements of some types of tasks might somehow be predictable there are other tasks which can have unexpected resource requests. Given the unpredictability of the resource requirements, energy-efficient mechanisms can benefit from becoming adaptive. Adaptivity enables them to respond to the changes in the state of the system and resource requests as they emerge, thus contributing to a reduction of total energy consumption.

Energy consumption can be reduced by deploying energy-efficient adaptive resource hyper-visioning mechanisms, either based on macro or micro level observation records. By reducing total energy consumption the drawbacks of high energy consumption on the system and the environment will be limited. The reduction of energy consumption can be achieved adaptively, based on macro and micro level observation records.

1.3 Key Elements of This Research

Resource hyper-visioning in private Clouds requires detailed decision making mechanisms to cater for the volatile resource state and VMs' resource requests. Resource hyper-visor is responsible for mapping VMs to hosts. This assignment is based on the current available information about the system state. A change in its status or VMs' resource requirements might make the outcome of the VM mapping policy unpredictable and/or unacceptable. The results of a deployment of a VM mapping policy can be an imbalanced system. VM migration can be an option for alleviating a potentially troublesome state when a system encounters an imbalance problem. Nonetheless, either mapping VMs to hosts or VM migration can benefit from the application of adaptive, feedback based mechanisms. It provides the hyper-visor with the chance to learn from the observed results - at macro and micro levels - and be able to respond to the changes as they emerge.

1.3.1 Research Scope

This thesis focuses on the problem of energy-efficient resource hyper-visioning in private Clouds. The scopes of this study are summarized in Table 1.1.

The aim is to reduce energy consumption in a heterogeneous private Cloud. To do so, resource hyper-visioning policies and mechanisms should be developed to administer virtualized resources (virtual machines). Resource hyper-visioning

Items	Scope
Target system	Heterogeneous private Cloud
Objective	Reduction of energy consumption
Energy saving techniques	Articulating the energy related indicators of certain system states and VMs' resource requirements, switching hosts on/off
Architecture	Adaptive resource provisioning using macro and micro level observation records

Table 1.1: Research Scope

includes both the VM mapping and VM migration, if needed; and switching hosts on/off to save energy. Such energy-efficient resource hyper-visioning is intended to be adaptive in responding to changes in the system and workload properties, using macro or micro level observation records.

1.3.2 Research Questions

Within the scopes of the research, the research questions can be summarized as follows:

1. *How do VM mapping policies perform, in terms of energy consumption, when workload properties are arbitrarily altered?* The energy consumption level of deploying multiple VM mapping policies might be substantially different when the simulation settings and/or workload properties (VM re-

source requests and/or task arrival rate) are changed. That is, an optimal VM mapping policy might not exist or be a practical approach within an acceptable time frame. This can be because each policy has its strengths and weaknesses in certain settings of system and workload properties.

2. *How can energy-efficient resource hyper-visor in private Clouds adaptively switch between VM mapping policies based on macro level observation records?* When different mapping policies demonstrate optimal results (in a macro level, in terms of energy consumption) when certain conditions (of the system and workload) are met, then there is a possibility to switch between policies and utilize their potential when they have previously had a strong outcome.

3. *How is resource hyper-visioning extended to VM migration, if energy-efficient mapping decisions caused imbalance, to reduce energy consumption in private Clouds?* A System experiences an imbalance when some hosts are either over-loaded with high resource requests or under-loaded with low resource requests. VMs should be selected from a host with an imbalance problem to be migrated. The process of selecting a VM for migration can be according to micro level observation records available at the time of migration. Thus the existing VM selection policies can be improved to reduce energy consumption.

4. *How can the micro level observation records be used to enhance the VM migration process to adaptively select a VM with the highest likelihood of reducing energy consumption in private Clouds?* When there is an imbalance problem in the system, VM migration can use micro level observation records to decide upon the energy efficient VM migration option. This process can be adaptive to the changes in the state of VMs and hosts.

5. *How can the micro level observation records about the state of hosts' resources and VMs' current requests can be used to adaptively map and migrate VMs, to reduce energy consumption?* Resource hyper-visioning decisions in private Clouds, either VM mapping or VM migration, can be based on an adaptive approach that learns from observing the outcome of its decisions given the state of hosts' resources and VMs' requests. That is, such adaptive resource hyper-visor relates the state of the system and its VMs to the expected outcome.

1.3.3 Research Hypotheses

The following hypotheses are expected from research questions:

1. *Mapping policies demonstrate their strengths and weaknesses (in terms of energy consumption as a macro level observation) in different states of system and workload properties.*

2. *Adaptively switching between mapping policies (based on macro level observation records) will demonstrate a low level of total energy consumption in comparison to individual policies when system and workload conditions are dynamically altered. The switching can become representative of the latest outcomes because of being adaptive.*

3. *When a VM migration needs to be carried out, VM migration policies can be improved to reduce energy consumption. The improved policies should outperform the existing policies in terms of total energy consumption.*

4. *How VM migration is enhanced to adaptively migrate a VM with the highest likelihood of reducing energy consumption in private Clouds?.*

5. *The process of VM mapping and VM migration can be incorporated in an adaptive and integrated mechanism. The adaptive mechanism will have the ability to relate (learn) the outcome of its decisions in terms of energy consumption given the state of system and its VMs for its VM mapping and VM migration mechanisms.*

1.3.4 Research Methodology

To answer the research questions, the following needs to be done.

1. *Simulations of the outcome of multiple mapping policies when system settings and workload properties are changed.* The results can provide insight into the outcome for each mapping policy. Mapping policies show their strengths and weaknesses given the state of the system and workload properties. Detecting the strength of each policy can help the system to use them when they are the most likely policy to reduce total energy consumption.

2. *When the conditions in which a mapping policy is outperforming the other policies are met, there can be an adaptive mapping mechanism that utilizes this macro level observation records to switch between mapping policies and reduce total energy consumption.* The overall result of such a mechanism needs to be close to the best performing VM mapping policy available to make the overhead of switches meaningful.

3. *VM migration policies should be developed when there is an imbalanced problem in the system and VMs should be migrated from hosts with an imbalance problem.* VM migration policies can be developed for energy-efficiency that utilize the current micro level observation records for VM migration decision. Such policies need to outperform the existing VM migration policies.

4. *VM migration should become adaptive and respond to the potential changes in the system and workload properties.* A VM migration mechanism can be developed that selects a VM with the highest likelihood of reducing energy consumption according to the micro level observation records.

5. *Mapping and migrating VMs in an adaptive way, using micro level observation records, can be achieved through the development of a mechanism that is inclusive of VM mapping and VM migration.* When VM mapping and VM migration are performed in a single mechanism, as a resource hyper-visioning mechanism, it becomes representative of both VM mapping and VM migration. Also, it will benefit from being adaptive as it will enable the resource hyper-vising mechanism to respond to the changes as they emerge, based on available micro level observation records.

1.3.5 Contributions

The contributions of this thesis can be divided into five categories: analysis and categorization of the related area of research, novel introduction of adaptive switching approach between VM mapping policies, optimization of VM migration process, and adaptive resource hyper-visioning inclusive of both VM mapping and VM migration. More specifically, the key contributions are:

1. *A review of the energy-efficient resource hyper-visioning techniques, inclusive of both VM mapping and VM migration, in Cloud computing.*

2. *Invention of an adaptive mechanism that switches between available mapping policies:*

    - Simulation of the outcome for multiple mapping policies when system settings and workload properties are changed.

- Detection of the conditions in which mapping policies demonstrate strength in reducing energy consumption.
- Development of an adaptive resource hyper-visioning mechanism that switches between mapping policies when previously detected conditions are met.
- Evaluation of the adaptive mechanism against individual mapping policies.

3. *Optimization of the VM migration process to solve the imbalance problem in private Clouds:*

   - development of VM migration policies that utilize available micro level observation records about VMs for VM migration decisions.
   - Comparison between the proposed policies and the state-of-the-art policies.

4. *Enhancement of the VM migration process to adaptively migrate the VM most likely to reduce energy consumption:*

   - *Formation of an inference to represent the relationships between the micro level observation records on VMs and hosts with observed energy consumption.*
   - *Simulation of the formed inference as an adaptive VM migration mechanism to adaptively migrate VMs to reduce energy consumption.*
   - *Evaluation of the proposed mechanism against state-of-the-art heuristics.*

5. *Invention of a novel adaptive resource hyper-visioning mechanism, inclusive of VM mapping and VM migration:*

- Establishment of an inference to relate the micro level observation records about system resources and VMs' resource requests to the energy consumption in the system.

- Application of the established inference in order to reduce total energy consumption as an adaptive energy-efficient resource hyper-visioning mechanism.

- Evaluation of the proposed hyper-visioning mechanism against state-of-the-art VM mapping and VM migration heuristics.

1.4 Overview of This Thesis

This thesis covers the development and evaluation of an energy-efficient adaptive resource hyper-visioning mechanism. It includes a set of new concepts and innovative mechanisms derived from papers published in international conferences and journals. An overview of the thesis chapters is as follows:

- Chapter 2 provides a review of the related work in the field of energy-efficient resource hyper-visioning inclusive of VM mapping and VM migration.

- Chapter 3 studies the motivation and the design of adaptive mapping mechanism by simulating the outcome of six basic mapping policies when the system settings and workload properties are altered. Then an adaptive mapping mechanism is proposed to switch between mapping policies according to the system and workload conditions based on macro level observation records. The proposed adaptive mechanism is then evaluated against individual mapping policies.

- Chapter 4 details the migration process and the proposal of two energy-efficient VM migration policies. The proposed policies are evaluated against an state-of-the-art policy.

- Chapter 5 presents an adaptive, Bayesian based VM migration mechanism and an adaptive energy-efficient mechanism, inclusive of both VM mapping and VM migration, to reduce energy consumption in private Clouds, based on micro level observation records. The proposed adaptive mechanisms are evaluated against a state-of-the-art heuristics.

- Chapter 6 summarizes the thesis main findings, analyzes the future research directions, and includes final remarks.

Chapter 2

LITERATURE REVIEW

INCREASING level of energy consumption by Cloud systems has instigated concerns related to the high cost of running the system and the associated high carbon dioxide footprint. These concerns have motivated research in developing resource utilizing approaches that seek energy efficiency in the Cloud system. Resource utilization is determined by resource hyper-visors. To identify challenges in energy-efficient resource hyper-visioning and to provide context for further advancements, it is essential to summarize and classify related research.

This chapter reviews the existing research related to energy-efficient resource hyper-visioning and provides a taxonomy and survey of the energy-efficient resource hyper-visioning in Cloud computing. A more general review of energy-efficient studies in the context of Cloud computing is provided in our survey paper [88].

This chapter is organized as follows. Section 2.1 gives a general introduction to the research objectives in the context of Cloud computing, and energy concerns, contributors and measures. It establishes the background for achieving energy efficiency through resource hyper-visioning. Resource hyper-visioning covering VM mapping and VM migration is reviewed in Section 2.3 and Section 2.4 respectively. Then a short discussion on the overall related literature and identified challenges are detailed in Section 2.5. Section 2.6 summarizes this chapter.

2.1 Optimization, Energy and Related Problems

Researchers choose criteria to be optimized in the context of Cloud computing. These objectives are reviewed and the relationships between the objectives are explained. As part of the objectives, the importance of energy consumption in Cloud systems should be explained. To achieve energy efficiency, contributors to the total energy consumption in the Cloud should be identified. By detailing these contributors, energy-efficient resource hyper-visioning decisions can be made.

2.1.1 Objectives in Cloud Related Studies

A range of objectives has driven research in Cloud computing. Studies have investigated the Cloud from the point of view of these objectives.

- *Energy/Power consumption:* Energy consumption is measured in terms of total energy consumed where power represents the energy used in an interval.

- *Resource utilization/wastage:* Resource utilization represents utilization level of all resources in the system.

- *Execution/turnaround time:* Execution time is the length of time taken for executing tasks. Turnaround time, however, is the time it takes from the submission of task to completion, which is the execution time and the waiting time.

- *Temperature:* Temperature of hosts in a Cloud system needs to be kept within a range to prevent burn-outs of hosts.

- *Cost:* In addition to the cost of hardware, its maintenance, software updates, staff and similar contributors, the cost of running a Cloud system needs to

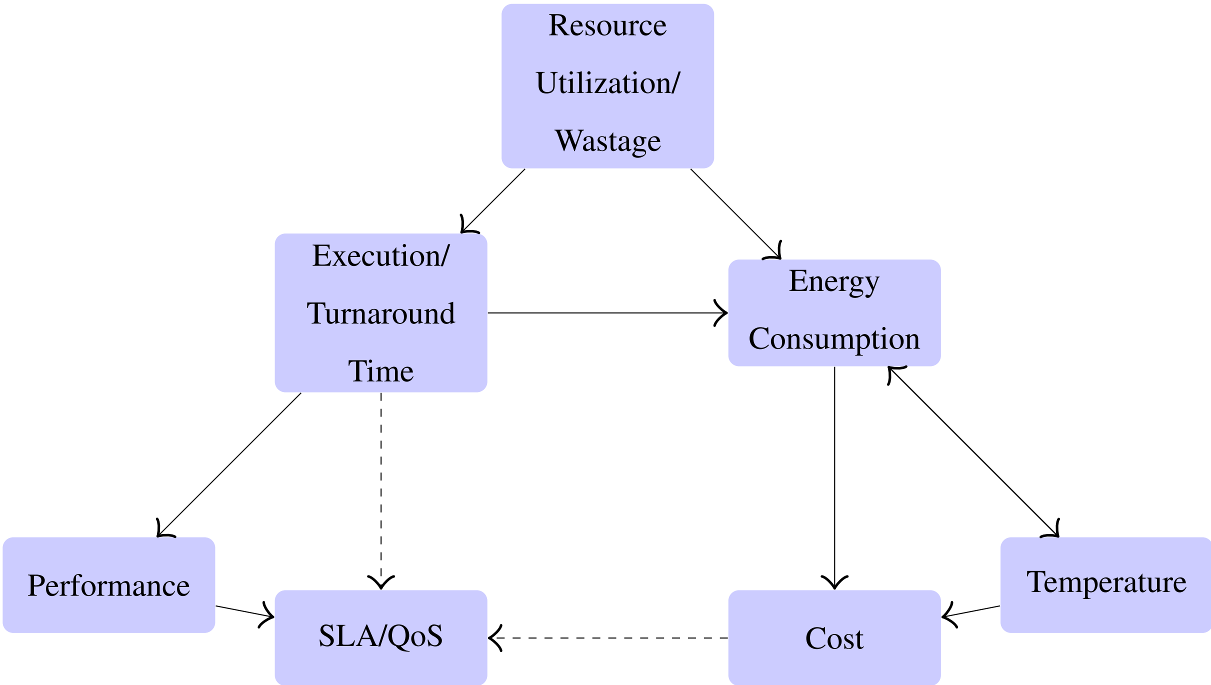


Figure 2.1: The relation between energy/power consumption and other potential objectives

include its energy bill.

- *Service Level Agreement(SLA) or Quality of Service (QoS):* SLA and QoS are defined according to a range of constraints including utilization level, execution/turnaround time and cost.

- *Performance:* The definition of performance varies in the literature. It is sometimes reported as the number of finished tasks or the utilization of a resource per unit of time.

Figure 2.1 depicts the relationships between these objectives. Energy/power consumption is directly related to resource utilization. Resource utilization, also, determines the execution/turnaround time. A given resource utilization level lasting a particular length of time indicates the energy consumed for execution. The higher the resource utilization or longer the execution time, the more the energy consumption.

In turn, high resource utilization and, therefore high energy consumption level increases the temperature. A rise in the temperature entails the need for more cooling devices that then adds to the total energy consumption and consequently the cost. To address the drawbacks of cooling devices' energy consumption, Tang, Gupta and Varsamopoulos [96] minimized energy consumption of cooling devices by developing heuristics that aimed at lowering the temperature. They reported 20 to 30 % energy saving. A similar approach is taken by Moore et al. [72].

Based on Figure 2.1, resource utilization determines execution/turnaround time. Shorter execution/turnaround time improves performance, when performance is defined as the number of finished tasks. Depending on the definition of SLA/QoS, either execution/turnaround time or cost affects SLA/QoS measures. Nonetheless, other objectives such as performance are also related to QoS in the literature, which itself relates to the execution time. A recent survey covers the broad range of QoS approaches in the literature [2].

As a result, the resource utilization level affects the energy consumption and execution time. In turn, resource utilization influences other objectives. Therefore, resource utilization is the major element affecting research objectives. Administering resource utilization enables the system to move towards optimal energy consumption and execution time. In this research, the objective is defined as a reduction in total energy consumption via resource hyper-visioning mechanisms where execution time is guaranteed not to be sacrificed.

Energy efficiency gains its importance from another aspect in Cloud computing which covers the problems caused by high energy consumption in Cloud data centers.

2.1.2 High Energy Consumption Problems

High level of energy consumption by Cloud systems gives rise to major problems within the Cloud system and for the environment. Problems within the Cloud system relate to the cost of running the Cloud system. In addition, problems for the environment are from the effect of Cloud's high energy consumption level on the environment in terms of carbon dioxide emission, when non-renewable energy sources are supplied [1].

The energy consumption by Cloud data centers accounted for 0.8% energy consumption worldwide in 2000 and 1.5% in 2005 [59]. In a report published in 2006, U.S. data centers alone accounted for 61 billion kilowatt-hours of energy consumption. This amount of energy was approximately 1.5% of the total power consumption in the U.S in that year. In 2007, Koomey [57] estimated the energy consumption by data centers was as high as 26 Giga Watts. In 2011, Koomey [56] reported that Cloud data centers were responsible for 1.1% - 1.5% of worldwide energy consumption.

Such high levels of energy consumption and the rapid growth in total energy consumption level mean higher energy bills for Cloud service providers. According to a university report [98], a medium-size data center pays an annual energy bill of about £1 million.

Moreover, the energy consumption directly impacts on the thermal state of hosts within the system. According to Sallam and Li [83], thermal state, T, for a host is proportional to Power, P, and the ambient temperature, T_{amb} as equa-

[1] Despite growth in generating energy from renewable sources, from 18% in 2007 to 21% in 2012, only 22% of energy generated worldwide was from renewable sources in 2013 [40]. One of the major energy consuming economies, China, still relies on oil as their primary source of energy [41].

tion 2.1, where R represents thermal resistance.

$$T = P.R + T_{amb} \tag{2.1}$$

Based on equation 2.1, a high energy consumption increases the thermal state of hosts. A high energy consumption level requires more cooling devices to keep hosts' temperature within an acceptable range. Running cooling devices adds to the total energy consumption, and further increases the energy bill.

In a private Cloud in particular, the cost of running the Cloud is one of the business expenses. A high level of energy consumption might prompt a business to question the viability of the provided services. Therefore, a business objective may be to reduce the energy consumption of its Cloud system to decrease total costs. Private Clouds are also responsible for the environmental effects resulting from their energy consumption level.

The high level of energy consumption has an impact on the environment. In a report published in 2006 [11], Cloud data centers were accounted for 116.2 million metric tons of carbon dioxide emission. Google data centers alone were responsible for 1.46 million metric tons of carbon dioxide emission in 2010 [71].

To tackle the problems caused by the high energy consumption level, energy-efficient approaches should be put into practice. Firstly, high impact energy contributors should be first identified. Then, energy estimation models and energy measurement approaches need to be investigated. With this information, energy-efficient approaches can be developed to reduce energy consumption in the Cloud, specifically in private Clouds with their relatively limited resources.

2.1.3 Contributors, Measurements and Models

Given the impact and the problems of high energy consumption in a private Cloud, it is essential to articulate the energy contributing factors within a private Cloud.

Then the energy measures and estimation models should be explained and the differences identified.

2.1.3.1 Energy Contributors

Total energy consumption in a Cloud system is the sum of energy consumed by hosts - based on their processing, memory, network and storage utilization level.

Processing utilization level represents the frequency with which the processor is executing instructions per second. The processing utilization level also includes the processor's activities regarding memory and network input/output.

The memory utilization level gets highlighted in specific scientific areas such as astronomy [20] and bio-informatics [74] where scientists need to store and analyze a large volume of data. Scientific memory needs to generate a large amount of data during the analyzing process [21]. Memory management (data/storage management) strategies are developed to optimize utilization of the available memory [112, 113]. Hence, such a level of memory utilization necessitates executing instructions by the processor.

Network traffic, also, adds to the total energy consumption in a Cloud system. Network transits involve copying the data through the network and that adds to memory and processing utilization level.

Nevertheless, research by Chen et al. [16] indicated that hosts' processing utilization is the main contributor to total energy consumption. The majority of energy is used by processors where memory, network and storage energy account for the majority of the remainder. Deelman et al. [22] reported that for running a scientific workload, computational cost (CPU utilization) outweighed the storage cost. As a result, the processor utilization level is commonly used to relate CPU frequency to energy consumption [45, 51, 84].

2.1.3.2 Energy Measurements and Models

One of the factors in energy related studies is the way energy consumption of processors is modeled or measured. Two main ways are used in research for calculating energy consumption:

- *Power meters*
- *Energy estimation models*

Power meters are attached to the hardware. They read energy consumption in intervals. The use of power meters has drawbacks including the following. First, the values read depend on the type of power meter used. Second, where the power meter is attached to the hardware affects the values read. A power meter connected to the motherboard adds a marginal energy consumption because of the reading process. Third, depending on how often the power values are read (big/small reading intervals), the power meters provide a discrete set of continuous energy consumption values.

Energy estimation models are required to provide a common basis for comparing the energy consumption in different studies. Such energy estimation models can then be adapted by power-aware simulation packages such as GreenCloud [54, 55], MDCSim [66], GSSIM [6, 61] and CloudSim [13, 14]. In 2013, a survey paper published by Kaur studied the different characteristics of MDCSim, CloudSim and GreenCloud [46]. CloudSim is one of the commonly used energy-aware simulation packages in Cloud computing. A review of CloudSim and its various versions can be found in research by Goyal, Singh, and Agrawal [33].

An energy-aware simulation package requires a formulae/model for calculating the energy consumption associated with a specific resource utilization level. Some of the energy consumption models are driven from the values measured on a hardware and provide an approximation of energy consumption [38, 39]. Hsu et al. [39] provided a set of rules that determines the energy consumption for VM_i, at the time t using α, the idle energy consumption and $\beta = \alpha$ as in equation 2.2 in Watts (W).

$$E_t(VM_i) = \begin{cases} \alpha & if\ idle \\ \beta + \alpha & if\ 0\% < CPUutil. \leq 20\% \\ 3\beta + \alpha & if\ 20\% < CPUutil. \leq 50\% \\ 5\beta + \alpha & if\ 50\% < CPUutil. \leq 70\% \\ 8\beta + \alpha & if\ 70\% < CPUutil. \leq 80\% \\ 11\beta + \alpha & if\ 80\% < CPUutil. \leq 90\% \\ 12\beta + \alpha & if\ 90\% < CPUutil. \leq 100\% \end{cases} \tag{2.2}$$

Energy consumption by processors can be divided into constant and dynamic consumption [51]. Constant energy consumption, E_c, is hardware dependent. Dynamic energy consumption, E_d, on the other hand, is based on the frequency of the processor and time as in equation 2.3.

$$E_d = \int_0^{\frac{t}{(f/fmax)}} C \times f_{max} \times f^2 \times t \tag{2.3}$$

where E_d is the dynamic energy consumption, t is the time, f is the frequency of the processor, and C is a constant. This is one of the widely used energy models and is adopted by CloudSim [13, 14] and is used in related studies [104, 83, 19]. CloudSim [13, 14] also includes the energy consumption values for utilization ranges of certain host types and is used for evaluation of the proposed heuristics [8, 9].

Given the effect of resource utilization levels, with processors in particular, resource hyper-visors should be reviewed to identify potential energy reduction opportunities.

2.1.4 Resource Hyper-visioning Software

In a Cloud system, resources on hosts are virtualized and Virtual Machines (VMs) are defined to accommodate tasks in the workload. The concept of a Virtual Machine (VM) was introduced by Intel during the 1960s, to provide interactive access to the mainframes to improve the utilization of the system resources [32]. Virtualization is defined [17] as "*a technology that combines or divides computing resources to present one or many operating environments*". Virtualizing resources enables sharing them between multiple VMs and therefore tasks.

VMs share the resources on the host that they are deployed on. Each task is assigned to a VM. Then the VM is mapped to a host along with potentially other VMs, all sharing resources on the host. VM mapping decisions are made by a software layer, Virtual Machine Monitoring software or resource hyper-visor, that administers workload isolation, workload consolidation and workload migration [100, 105].

Workload isolation is the process of assigning tasks to VMs. Workload consolidation refers to the process of mapping VMs to hosts. Workload migration, live migration or application mobility [7] refers to the action of migrating a VM from one host to another. In this research, workload consolidation and workload migration are addressed. The reason for leaving out the workload isolation is the fact that this research bases its mechanisms on resource requests and utilization level of VMs, regardless of the number of tasks deployed on them. Therefore, we address VM mapping and VM migration as VM monitoring/resource hyper-visioning.

Common VM monitoring softwares (resource hyper-visors) are: VMware [42], Denali [109], Xen [7], Kernel-based Virtual Machine (KVM) [53] and Virtual PC [37].

VMware

VMware is a software company that provides virtualization software and Cloud services. It has VMware Workstation software that runs on Microsoft Windows, Linux, and Mac OS X. It also has VMware GSX Server and VMware ESX Server that are embedded hyper-visors and run directly on server hardware without requiring an additional underlying operating system [42].

Denali

The University of Washington's Denali project came up with a technique, which is called paravirtualization [109]. paravirtualization tries to increase the scalability and performance of Virtual Machines. It also modifies the traditional virtualization architecture for new customized guest operating systems in order to obtain extra performance. Denali is designed to support thousands of virtual machines running network services. Its VMs are designed with the aim of hosting a single task, single-user unprotected guest OS and thus does not have support for virtual memory which is a common feature in most OSes.

Xen

Researchers at the University of Cambridge proposed an architecture for virtualization, called Xen [7]. Xen aims to maximize performance and resource isolation using a paravirtualized architecture.

Kernel-based Virtual Machine (KVM)

The Kernel-based Virtual Machine (KVM) is a Linux virtualization subsystem. KVM, a hardware-assisted virtualization, improves performance and allows the system to support unmodified guest operating systems [53]. The cooperative Linux (COLinux) is an example of it, which is a variation of user-mode Linux. User-mode Linux resides on top of the OS and works in user-space, while cooperative Linux might run some specific kind of VMs in kernel-mode.

Virtual PC

Microsoft Virtual PC is similar to VMWare. It provides the facility of having multiple operating systems and multiple VMs. It supports two features: undoing disk and binary translation [37]. The option of undoing disk lets user undo some previous operations on the hard disks of a VM. Binary translation, provides x86 machines on Macintosh-based machines.

The VM monitoring software, namely resource hyper-visor, arbitrates the access to the physical resources so that different operating systems in VMs can share the host infrastructure. Resource hyper-visor decides about the tasks to VMs assignment. It then maps VMs to hosts. It is also responsible for migrating VMs if needed.

Achieving energy efficiency via resource hyper-visioning is a challenge because of the heterogeneity of resource types, the unpredictability of the future resource requests and the nonlinearity of energy consumption based on resource utilization.

The resource requests from the task assigned to a VM are sent to the host that has the VM running on it. It is the effect of the resource hyper-visor's decisions on the hosts' resource utilization. Its VM to host mapping decisions and VM migration decisions can become energy aware so that total energy consumption for running a given workload is reduced. Energy-efficient VM mapping and VM migration heuristics are developed to achieve this.

2.2 Energy-aware Resource Hyper-visioning Approaches

Energy-aware resource hyper-visioning is achieved via different approaches in the context of Cloud computing, including the formulation of resource hyper-visioning as a *Backpack problem* or *Trade-off*.

Resource hyper-visioning in Cloud is formulated as a Backpack problem by some studies where available resources indicate the capacity/capacities (considering dimension to serve processing, memory and network capacities). Examples of a Backpack formulation approach include: Aydin et al. [5] proposed an energy-aware task execution where tasks' deadlines are met; Raghavendra et al. [78] mapped VMs to hosts based on hosts' remaining capacity and the constraints where they switch hosts on and off as required; Srikantaiah, Kansal and Zhao [93] mapped as many tasks to a host as possible and measured the energy consumption and reported a result close to optimal energy consumption.

Resource hyper-visioning in Cloud is also formulated as a trade-off between objectives. Salehi and Buyya [82] experimented with cost and time trade-offs when dealing with the problem of executing tasks on available public Clouds, given the budget and time constraints. Ghosh et al. [31] quantified power (energy per time) and performance trade-offs.

To address the energy consumption problem, whichever way it is formulated, different strategies are applied, including a variety of optimization approaches, e.g. Genetic Algorithm (GA) by Lee et al. [106], Yu and Buyya [111]; Ant Colony Optimization (ACO) by Zhu, Li and He [115], Feller, Rilling and Morin [24], Gao et al. [29]; and Honey Bee heuristic by Krishna [60].

A review of the closely related papers addressing energy-efficient VM mapping and VM migration strategies is provided in the next two following sections.

2.3 Energy-aware VM Mapping

Energy-efficient VM mapping heuristics aim at optimizing energy consumption. Energy optimization can be achieved by applying energy-efficient heuristics on hardware or at the resource hyper-visioning level. It can be achieved by adjusting the frequency of the processor, Dynamic Voltage and Frequency Scaling (DVFS), or switching hosts on/off when needed.

At the hardware level and to adjust the frequency of the processors, Ren et al. [80] proposed three DVFS-based VM mapping heuristics: Least performance Loss Scheduling (LLS), No performance Loss Scheduling (NLS) and Best Frequency Mach scheduling (BFM) policies. These policies were invented to tackle the drawback of Xen [7] of adjusting the frequencies fast, which leads to performance degradation. They evaluated the proposed policies on single-core and multi-core processors and reported a reduction in energy consumption. Cardosa et al. [15] utilized Xen [7] hyper-visioning software's property, of determining min, max and CPU proportion allocated to a VM, to devise a power-efficient VM mapping heuristic.

In this study, however, the focus is on the resource hyper-visioning level when either DVFS or host switches are applied.

Kim, Cho and Seo [52] estimated the energy consumption of a VM based on the in-processor events. They proposed a VM mapping policy, Energy-Credit Scheduler (ECS), which regulates the energy consumed by a VM of a certain budget. This approach is suitable for a public Cloud where users are given access to VMs for a certain time and at a given price.

Plaxton et al. [77] designed a resource to request mapping policy that reduces the resources required for executing the requests. Irani et al. [43] mathematically proved that their proposed probabilistic mapping approach outperforms the deterministic approach by a substantial margin. Kim, Buyya and Kim [51] used a DVFS technique in a way that aims to meet tasks' deadline. They evaluated their energy saving heuristics against space sharing and time sharing policies (without DVFS) and reported significant energy reduction. Executing tasks with deadlines in private Clouds is then extended by Kim, Beloglazov and Buyya [49, 50] to improve energy efficiency and tasks acceptance rates; and meet tasks' deadlines. Lin et al. [67] proposed an algorithm for mapping requests to hosts so that hosts are switched off when there is low level of resource requests.

Verma et al. [101] developed an algorithm to minimize power consumption. In their recent papers [102, 103], bi-criteria (deadline and budget) algorithms are proposed where they applied Particle Swarm Optimization (PSO) for VM mapping. They compared their algorithms with BHEFT proposed by Zheng and Sakellariou [114], which itself is an alteration of HEFT [97]. PSO is also used by Xiong and Wu [110], and Sridhar and Babu [92] for mapping. Another VM mapping policy is proposed by Kaur and Challa [47] where energy consumption and execution time are the objectives to be optimized.

The above mentioned studies are static approaches where a set of rules is put in place to optimize the corresponding objectives. However, they do not take into account the potential and frequent changes in the resource requirements and available resources at run-time.

Back-filling refers to the process of selecting the next task to assign to the resources. Mu'alem and Feitelson [73] used back-filling on IBM servers. Variations of back-filling are applied by Tsafrir, Etsion and Feitelson [99] and Suresh and Vijayakarthick [94] for VM mapping in the context of Cloud computing. Back-filling relies on information or estimations of resource requirements. Such information might not be available and estimations might manifest an error rate that jeopardizes the viability of the mechanism.

Reguri, Kogatam and Moh [79] designed improved VM mapping heuristics that clustered VMs based on their resource requirements. They evaluated their heuristics on CloudSim [13, 14] and compared them against heuristics without clustering. The results demonstrated significant energy reduction when Best Fit VM (BFV) and Best Fit Host (BFH) policies are enhanced by the proposed clustering approach. They, however, presume prior knowledge about resource requirements which might not be available or accurate.

Gaggero and Caviglione [27] proposed a VM mapping heuristic to minimize energy consumption. They aimed at maintaining a certain level of SLA/QoS. They evaluated their heuristic against First Fit Decreasing (FFD) and an improved FFD heuristic developed by Panigrahy et al. [75]. They reported reduction of energy consumption. However, they set a static rule for performing VM mapping based on the available information. It does not adapt to potential changes in the state of the system and VMs' resource requirements after VMs are mapped.

Gandhi et al. [28] analyzed the effect of task arrival rate on resource utilization levels and consequently response time and power consumption. They set utilization levels and a power cap. They, then, directed the future work towards switching between utilization levels to achieve shorter response time and power caps. However, the presumed switching approach bases its decisions on the utilization level only where it could benefit from a feed back loop.

One of the common approaches to define the VM mapping problem is to formulate it as a bin-packing problem. Srikantaiah et al. [93] studied the minimization of energy consumption while meeting performance requirements when resource requests are to be mapped to bags of resources. The authors illustrated performance degradation due to high utilization level. They reported an optimal resource utilization level in terms of performance. However, reported results are validated on specific workloads and task types.

Aydin et al. [5] proposed a mapping heuristic to map VMs to hosts while meeting their deadlines and saving energy. They assume that the best-case and worst-case resource requirements of each VM is known to the VM mapper. Raghavendra et al. [78] explored energy management and utilized a bin-packing algorithm for VM mapping that works with available information regarding VMs' resource requirements and the remaining capacities on hosts. Ghosh et al. [31] described a trade-off between power and performance when mapping VMs to hosts. They assumed the availability of information regarding the average resource utilization by each VM.

Lee et al. [106] mapped tasks to hosts based on their resource requirements, using GA. Although in the context of Grid computing, Yu and Buyya [111] proposed a mapping policy based on deadline and budget constraints using GA.

Ant Colony Optimization (ACO) is applied by Zhu, Li and He [115] to map tasks (their VMs) to hosts when the resource requirements are known to the mapper. ACO is also used by Feller, Rilling and Morin [24] to map VMs to hosts when formulating the problem as a multi-dimensional bin-packing problem. They consider the presence of knowledge about all resource requirements. Gao et al. [29] utilized ACO for VM mapping as well. The proposed heuristic works based on the idea that the resource requirements of VMs are known and a VM is mapped to a host only if the host can provide the resources.

In these studies, the request to resource matching problem is investigated. They considered prior knowledge about tasks' resource requests. These studies did not consider the potential changes in resource requests after the VM to host mapping heuristics are applied. They also do not include feedback from the previous mapping decisions to guide the heuristic in the next phase. Therefore, they do not represent an adaptive VM mapping strategy with the ability to learn from its previous decision and outcome scenarios.

Beloglazov and Buyya [8, 9] proposed threshold based mapping policies: Thr, IQR, MAD, LRR and LR. In a recent study by Sharma and Saini [85] a median based threshold policy is evaluated against Beloglazov and Buyya's policies and is proven to outperform THR, IQR and MAD. However, prescribing a threshold for the utilization level on hosts might impose non necessary pressure on hosts and keep more hosts than needed in their active state. Such threshold levels can also show deficiency when dealing with a volatile resource requests.

This research, however, keeps an overall view of the system state and VMs' resource requests throughout the execution. The reason is to cater for ever changing state of system resources and resource requests from VMs. It is emphasized in private Clouds where tasks come from a broad range of business units with relatively different resource request profiles in their life-time.

Caglar and Altilar [12] proposed a VM mapping heuristic that predicts the number of required hosts to avoid hosts being over-loaded. They avoid VM migration because of the VM migration cost, in terms of energy and time required for copying VMs. They evaluated their heuristic on CloudSim [13, 14]. They compared the results of the proposed heuristic with VM mapping and VM migration policy proposed by Beloglazov and Buyya [8, 9]. The results of the comparison indicated that the proposed VM mapping heuristic [12] keeps less hosts in their active state and imposes no migration. The results of energy consumption by the heuristic that includes migration [8, 9] was significantly lower. The authors argued that their heuristic imposed less host switches and therefore less switching energy overhead. Nevertheless, energy consumption (switching energy overhead included) is not reported or compared with heuristics from VM migration.

Despite available VM mapping heuristics aiming at minimizing energy consumption in the Cloud, they have limitations achieving this aim. First they commonly rely on historical information on resource requests or an estimation of them. Second, they do not include a feedback loop or a learning capability. A method with such an ability can potentially help the VM mapping heuristics to reflect on the latest outcome.

Most machine learning methodologies learn from data and aim to predict an outcome, e.g. Regression and Logistic Regression, Decision Tree, Random Forest. Among these methodologies, some have a relatively high cost of learning where the model should be repeatedly run for every change in the data, so that the model is representative of the newly added data. The cost of such re-runs are different for different methods. Bayesian Inference (BI) is based on the likelihood of an event according to other contributing factors. Therefore, it is relatively fast in reflecting on newly added data because it only requires the re-calculation of some probabilities. It suggests BI is a suitable candidate in response to problems with high volatility level.

With the goal of guaranteed execution of tasks in case of hardware failure, Wang et al. [107] fairly distributed the load in the system using the concept of trust and Bayesian cognitive. The degree of trust in each host is described as the node's (host's) performance and the feedback about its performance in cooperation with other hosts. The proposed framework is an extended version of a framework presented by Harrison [36] in 1975. The data from a study by Calheiros et al. [14] are used for evaluation and shows their approach can reduce tasks' failure ratio.

In one of our publications [86], Bayesian Inference is used to learn from the outcome of six VM mapping policies where the proposed adaptive VM mapping mechanism switches between available VM mapping policies based on the current state of the system and previous results of deploying a given VM mapping policy. The proposed adaptive VM mapping mechanism is evaluated against individual VM mapping policies. The results indicate the adaptive VM mapping mechanism outperformed all policies in terms of energy consumption as well as execution time.

Although the optimization of energy consumption is the paramount reason for inventing energy-efficient VMs to hosts mapping policies, the unpredictability of resource requests and changes in available resources in the system might make the initial mapping of VMs to hosts less efficient throughout the execution. One of the major issues can be an imbalance problem. A solution to solving imbalance problem is to migrate VMs.

2.4 Energy-aware Migration

Since the advent of Cloud technologies researchers have devised VM mapping algorithms to achieve energy efficiency. However, VM mapping policies can lead to imbalance in the system as the resource requirements from VMs and available resources on hosts change throughout execution. The imbalance problem occurs when there are hosts that are either over-loaded or under-loaded. VM migration is a potential solution for easing an imbalance problem. Few research efforts have been directed at the problem of migrating VMs when they are running on imbalanced hosts, either over-loaded or under-loaded. Nevertheless, an energy-efficient VM migration process can reduce energy consumption and has the potential to improve execution time by solving the imbalance problem.

VM migration is used by Khanna et al. [48] with the incentive to prevent performance degradation. They aim at fitting as many VMs on a host as possible without sacrificing the performance.

A Honey Bee Behavior-Load Balancing (HBB-LB) strategy is devised by Krishna [60] where VMs are mapped and migrated between hosts when needed. The initial VM mapping strategies covered in the study include experiments with FIFO and WRR (Weighted Round Robin). Then, the honey bee foraging strategy divides the processors into three groups: overloaded, underloaded and balanced. This strategy migrates VMs from overloaded hosts to a host in the underloaded group. The previously overloaded host will be added to the balanced group, if, after the migration the host is no longer overloaded. Throughout the migration process, overloaded hosts and VMs running on them are not prioritized in order to minimize energy consumption. The host from the underloaded group is selected randomly without considering energy consumption.

Bobroff, Kochut and Beaty [10] perform VM migration to reduce the number of active hosts in a way that SLA threshold is not violated. They evaluated their heuristic against static VM mapping (absence of VM migration) and reported up to 50% reduction in resource utilization (consequently energy consumption). However, the proposed heuristic sets static rules for VM migration and relies on forecasting resource requests which might not be available or accurate.

In a recent paper by Mazumdar and Pranzo [68] VMs are migrated to switch hosts off when possible in order to reduce energy consumption. They compared their approach with Best Fit heuristics and reported 6 to 15 percent improvement.

Kusic et al. [63] minimized energy consumption in virtualized environment using Limited Lookahead Control (LLC) [3]. The proposed optimization heuristic aimed at maximizing the Cloud provider's benefit by reducing energy consumption and avoiding SLA violation. They estimated future resource requests using the Kalman filter in order to migrate virtualized resources if needed and switch hosts off to achieve lower energy consumption. Nevertheless, the complexity of the estimations and optimization processes imposes a potential increase in the execution time.

Gaggero and Caviglione [26] proposed a VM migration heuristic to minimize energy consumption while maintaining SLA/QoS. They evaluated their heuristic on a small data center - with 50 hosts - and a maximum of 5 VMs per host. The evaluation was conducted based on different scenarios and reported that frequent migrations leads to minimization of total energy consumption as more hosts are switched off. They also reported higher SLA violation because of migrations. The trade-off between energy consumption and SLA violation was outlined.

To choose an optimal VM to host migration candidate, Sallam and Li [83] proposed a multi-objective Bayesian game approach. The objectives of load volume, energy consumption, thermal state, resource wastage and migration cost are being minimized. They compared their multi-objective heuristic against heuristics with one of the five objectives at a time. However, their approach aims at minimizing the load volume, and this might cause the system to keep more hosts in their active state than are needed. Energy consumption and thermal state are related. The impact of selecting both energy consumption and thermal state as two of the five objectives makes the optimization algorithm skewed.

Among approaches that tackle the imbalance problem by migrating VMs, Mi et al. [70] applied Genetic Algorithm (GA) to adjust the data center to the dynamically changing demands with the goal of minimizing energy consumption. Resource requirement prediction is used that its affects the outcome of such approaches.

Tang et al. [95] used a peer-to-peer infrastructure to poll many hosts at the same time for available resources by estimating the tasks' resource requirements according to the previous resource utilization of the same tasks. However, previous utilization levels might not represent the actual future requests. Moreover, their approach is focused on enterprise data centers where, the managing software continuously checks all loaded hosts, switching off redundant VMs and hosts, or shifting VMs if necessary.

Kusic et al. [62] proposed SLA insuring algorithms for small server clusters - termed as Gold and Silver - while minimizing energy consumption. The algorithm projected the expected demand/request to prevent frequent switches and dynamically decides upon the number of VMs and hosts. The accuracy of the expected demand/request plays a role in the effectiveness of the algorithm.

Beloglazov and Buyya developed three energy-aware VM migration policies and reported Minimum Migration Time (MMT) as the most energy optimal migration policy [9]. MMT selects a VM that is the fastest to be transferred through the network. The transfer/migration time is calculated based on the VM's memory size of the VM and the available bandwidth between hosts. Where a host with low bandwidth is imbalanced, VMs from that host will be the last to be chosen by MMT.

We developed VM selection policies [87] called MedianMT and MaxUtil. MedianMT and MaxUtil are evaluated against MMT. Both MedianMT and MaxUtil outperformed MMT in terms of energy consumption as well as mean execution time. MaxUtil manifested the best outcome among the three in terms of both energy consumption and mean execution time.

However, in the Cloud, with its volatile resource status, an adaptive mechanism can benefit from selecting the best VM from the imbalanced host to migrate, given the current state of resources and VMs in order to reduce energy consumption. It can be achieved by developing a Bayesian Inference that relates the current state of the system with the outcome of a VM migration in terms of energy consumption. Such adaptive mechanism will learn and respond to changes in the resource state.

So we invented an adaptive VM migration mechanism [89], and an adaptive resource hyper-visioning mechanism, inclusive of both VM mapping and VM migration, and evaluated them against state-of-the-art heuristics. The result showed substantial reduction in energy consumption as well as a shorter mean execution time.

2.5 Discussion

Existing VM mapping and VM migration heuristics either rely on information about VMs' resource requirements or an approximation of future resource requirements. They commonly recommend a definite set of rules to apply in order to minimize energy consumption. However, future resource requests are not always known to the hyper-visor. Moreover, setting rules without the flexibility of learning might result in failure to respond to changes in either the system state or VMs' resource requirements. Adaptive VM mapping and VM migration mechanisms are needed to seek energy reductions without relying on prior information about VMs' resource requirements. An adaptive mechanism has the learning ability to respond to changes as they occur.

One approach for developing adaptive VM mapping and VM migration mechanisms is the application of probabilistic methods. Probabilistic methods have their foundation in their decision on the expected probability of an event according to the recorded values, i.e. a specific input is related to an output. The adaptability emerges from the ability to update the probabilities based on the observed values.

One of the widely used probabilistic approaches is Bayesian Inference (BI). BI has shown strength when dealing with problems of high volatility levels. Problems cover a range of research areas including but not limited to embedded software optimization [4], failure management [34, 35] and mobile Cloud computing [108].

BI has its potential in statistically estimating the likelihood of reduction in energy consumption given the state of the system. BI updates its probabilities according to the results. Such methodology can enable the resource hyper-visor to learn and adapt to the changes as they emerge. It provides the Cloud hyper-visor with the versatility and adaptivity it requires.

A VM mapping policy equipped with Bayesian learning technique is expected to provide a better system outcome for Cloud computing systems and private Cloud in particular, because of its ability to efficiently utilize its limited resources. To investigate the impact and the significance of an adaptive VM mapping mechanism, we developed a dynamic energy-efficient VM mapping mechanism using BI [86]. The mechanism switches between available VM mapping policies, whenever the proposed BI indicates a likely reduction of energy consumption by a policy. The outcome indicated significant reduction of energy consumption. It also resulted in shortened execution time.

We, then, proposed two VM selection policies for migration [87] called MedianMigration Time (MedianMT) and Maximum Utilization (MaxUtil). Both policies outperformed, what was at the time, the best VM selection policies reported by Beloglazov and Buyya [9], MMT when combined with multiple VM mapping policies included in CloudSim simulation package [13, 14]. Later, we designed an adaptive VM selection for migration mechanism [89], based on BI, that further reduced total energy consumption and shortened mean execution time.

Despite the proposal of adaptive VM mapping mechanism and adaptive VM mapping mechanism using BI, these mechanisms used records of different levels in the corresponding BI. That means, two sets of information log and Bayesian probability calculations, one for VM mapping and another for VM migration. In addition, VM mapping decisions are unaware of their outcome in terms of the imposed VM migrations. And VM migration decisions are only based on selecting the best VM for migration. The required dual log and lack of relation between VM mapping and VM migration decisions motivated this research to design an energy-efficient dynamic VM hyper-visioning mechanism. By applying the proposed mechanism, the total energy consumption is significantly reduced, while the mean execution time is substantially shortened.

2.6 Summary

In this chapter, Cloud resource hyper-visioning and its objectives are explained. The problems caused by high energy consumption level are analyzed. The importance of energy consumption and its relation with resource hyper-visioning is then described. Studies related to resource hyper-visioning, either VM mapping or VM migration, are reviewed. Furthermore, the literature review provided the background and necessity of adaptive resource hyper-visioning mechanisms. A survey of the energy-efficient resource hyper-visioning in Cloud computing is provided.

Chapter 3

ADAPTIVE ENERGY-EFFICIENT SWITCHING BETWEEN VM MAPPING POLICIES IN PRIVATE CLOUDS

RESOURCE hyper-visioning includes mapping VMs to hosts. The first step in answering research question 1) *How do VM mapping policies perform, in terms of energy consumption, when workload properties are arbitrarily altered?* - is to study the changes in the results of different VM mapping policies in terms of energy consumption, given arbitrary alterations in workload properties. Because reducing energy consumption should not entail an increase in execution time, total execution time should also be reported.

Based on the results of VM mapping policies, and to answer the research question 2) *How can energy-efficient resource hyper-visor in private Clouds adaptively switch between VM mapping policies based on macro level observation records?*, a novel adaptive mechanism is proposed to switch between VM mapping policies at run-time. The aim is to reduce energy consumption by learning from the observed outcome of each policy using Bayesian Inference (BI). The outcome is measured in terms of total energy consumption, a macro level observation record. Based on the observation records an adaptive mechanism is invented that switches between available mapping policies according to system state and workload properties.

The proposed mechanism is evaluated against individual VM mapping policies and resulted in a total energy consumption level close to the best performing VM mapping policy. It also had similar execution time to the best performing VM mapping policy. The results are published as a paper [86].

This chapter is organized as follows. Section 3.1 reports the simulations that motivated switching between VM mapping policies. Section 3.2 details the BI introduced, comprising: Bayesian Network, its components and algorithms for calculating Bayesian probabilities. Our Adaptive Switching Mechanism (ASM) is then evaluated in Section 3.3. It is followed by a discussion in Section 3.4 and a summary of this chapter in Section 3.5.

3.1 Motivational Simulations

The propagation of Cloud data centers has radically changed the IT industry by providing services for businesses and individuals per demand. Private Clouds in particular aim to serve the internal units within a business. Private Clouds, in comparison to public Clouds, have less resources available. The way these limited resources are utilized affects total energy consumed for serving business units. It also affects the execution time. Resource utilization is determined by resource hyper-visor. Resource hyper-visioning includes mapping VMs to hosts.

Among available VM mapping policies, six basic VM mapping policies which do not rely on information about future resource requests for mapping, are selected. The results of deploying these VM mapping policies are to be simulated, in terms of energy consumption, when task arrival rate and resource requests are arbitrarily altered. The incentive for these alterations is to identify the strength of each VM mapping policy given the state of system and workload properties.

3.1.1 Six Basic VM Mapping Policies

Among existing VM mapping policies, six basic policies are selected. The reason for this selection was that these policies do not rely on information about future resource requests. Policies are as follows.

- *Maximum Utilization (MU) [64]:* It maps VMs based on the utilization level of available hosts. It tries to maximize the utilization level (processor) on a host before switching another host on. In other words, it focuses on consolidating as many tasks onto fewer hosts, in order to reduce the number of active hosts.

- *Intensity Based (IB):* It follows the elementary idea of mapping the VM of an intensive task (processing-intensive, memory-intensive, network-intensive) to a host with the most available capacity on that resource. A non-intensive task is randomly mapped.

- *Greedy Deadline (GD):* It aims to improve execution time by mapping VMs on hosts with the least (processing) utilization level. It does not turn on a new host unless the active hosts cannot accommodate any new VMs.

- *Intensity-aware Greedy Deadline (IGD):* IGD is similar to GD with the difference being how it deals with intensive tasks. If the host with minimum utilization on resource A (intensity is defined on the following resources: CPU and memory) is running an A-intensive task, the host with the second minimum utilization on resource A will be selected, unless all hosts have the same number of A-intensive tasks.

- *Equal Load (EL) [65]:* It is a VM mapping strategy that tries to balance the load in the system. EL keeps the number of VMs on each host as equal as possible.

- *Intensity-aware Equal Load (IEL):* IEL behaves like EL but takes into account the intensity of the tasks as described in IGD. If the host selected by EL is running an A-intensive task, another host will be chosen, unless all hosts are running the same number of A-intensive tasks.

3.1.2 Simulation Attributes

To observe the results of deploying VM mapping policies when workload properties are altered, resource requests in 100 workload sets are generated using different statistical random number generators. The mean arrival rate is also altered within a range.

3.1.2.1 Workload Property, Resource Requirements

To cover a diverse set of workloads, resource requirements are generated randomly following a diverse set of statistical distributions (with the same mean value). Each workload consists of 1000 tasks. There is the same number of tasks from CPU-intensive, memory-intensive, network-intensive and non-intensive tasks which translates to 250 each with no specific order of arrival.

To generate tasks that are intensive on a resource, the mean value for the random number generator is set higher which increases the chance of generating a larger number. To obtain a diverse set of random number generators the following discrete distributions are used for generating workload sets: Uniform, Bionomial and Zipf. These distributions are to represent the statistically different workloads. Each distribution is used to generate 100 workload sets which makes the total of 300 workload sets, each with 1000 tasks.

3.1.2.2 Workload Property, Arrival Rate

The tasks' arrival intervals are generated according to the Poisson distribution. Random numbers generated by Poisson distribution are unrelated to their preceding or succeeding number. It makes Poisson distribution a suitable option for generating task arrival rates. The Poisson distribution's mean value varies from one to seven units of time to cover the circumstances where the tasks arrive relatively fast (1) to slow (7). The alterations are to represent the changes in the system's behavior when it is under more pressure or relaxed. For each case the simulation is repeated 100 times.

3.1.2.3 Simulation Settings

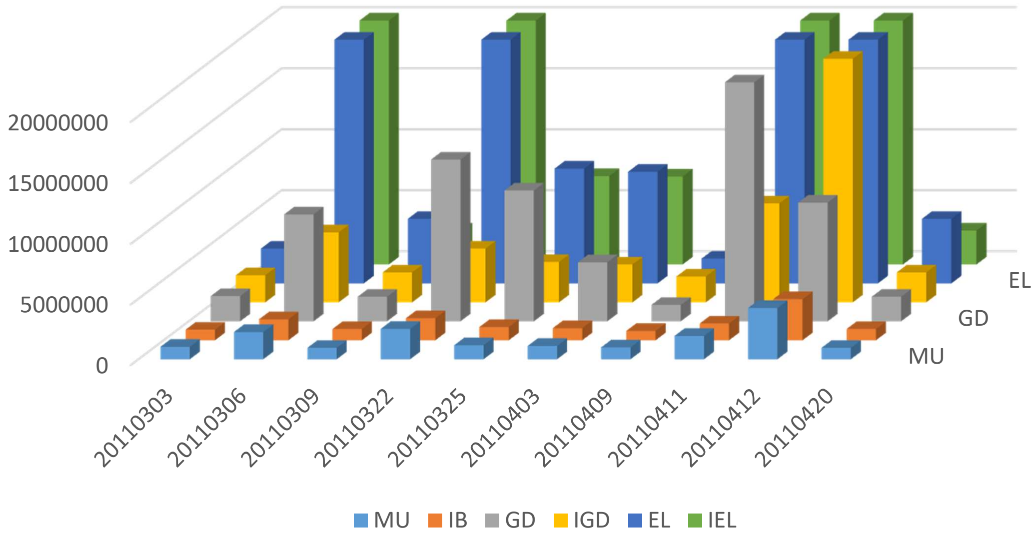
A system with 20 homogeneous hosts is simulated. The assumption of homogeneous hosts was made to eliminate the impact of differences in host capacities on the outcome of VM mapping policies. The initial available bandwidth to each host is set to 1000 Hertz.

3.1.3 Analysis of VM Mapping Policies' Results

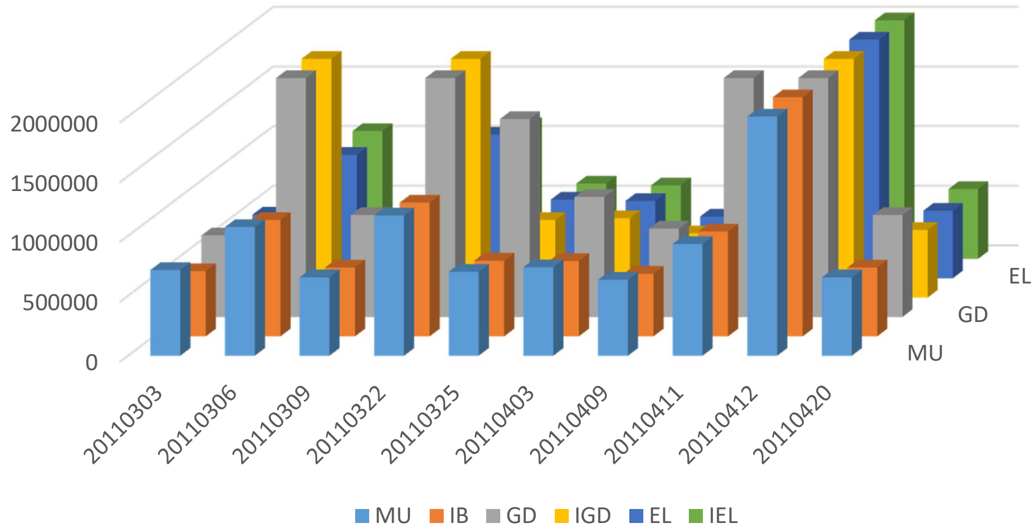
VM mapping policies in Section 3.1.1 are deployed when mapping VMs with initial resource requirements of workloads provided in CloudSim [14, 13] which are from PlanetLab [76].

The results of deploying these VM mapping policies are illustrated in Figures 3.1 and 3.2 in terms of total energy consumption (Watt) and total execution time respectively, when task mean arrival rates range from 1 to 7 time units.

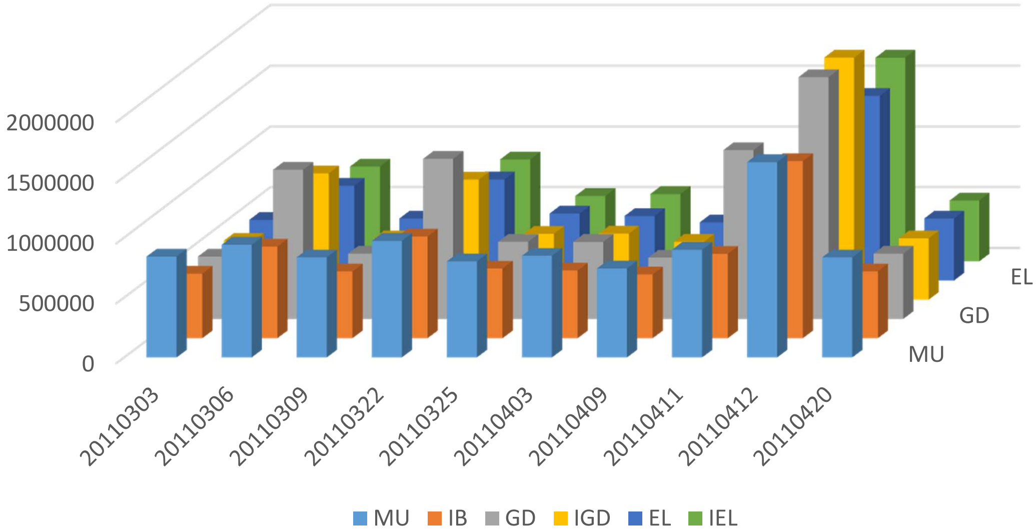
As in Figure 3.1a EL and IEL mapping policies demonstrated a high level of energy consumption compared to other policies. IB, GD, on the other hand, resulted in a relatively - to other VM mapping policies - low energy consumption level. As the mean arrival interval increases as shown in Figures 3.1b to 3.1g the outcome of VM mapping policies changes considerably. In Figure 3.1g VM mapping policies of low energy consumption level, including IB, GD and MU, have higher energy consumption than the worst performing VM mapping policies, EL and IEL, when the mean arrival rate is smaller/shorter.



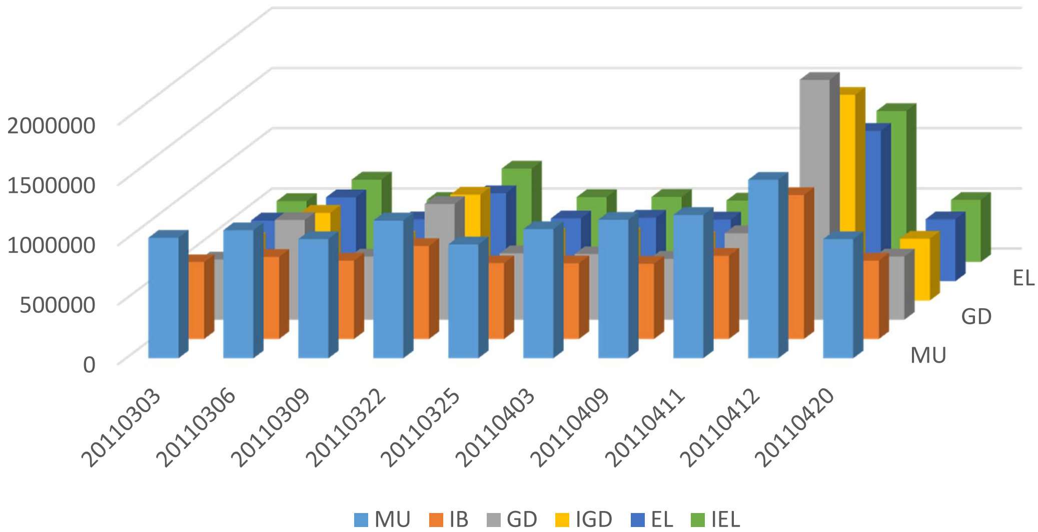
(a) Mean arrival interval = 1



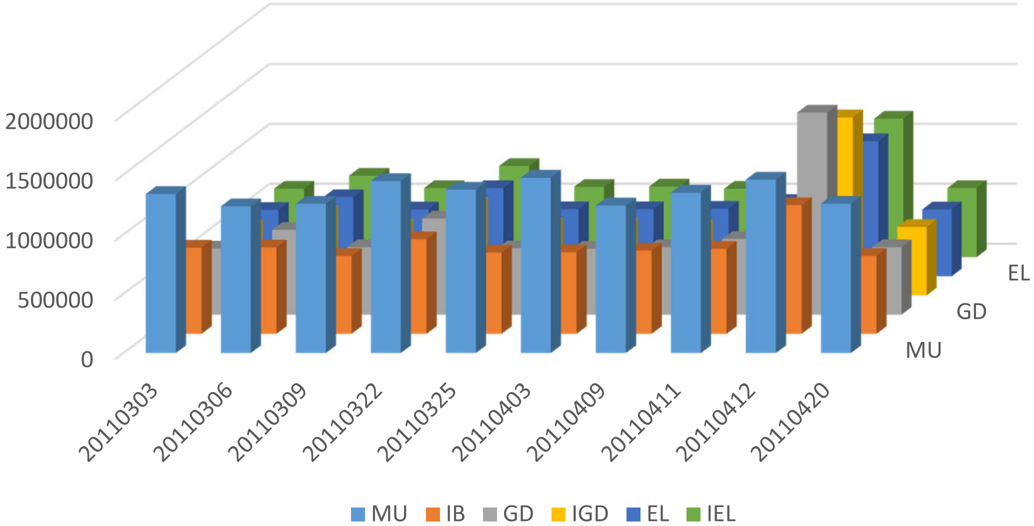
(b) Mean arrival interval = 2



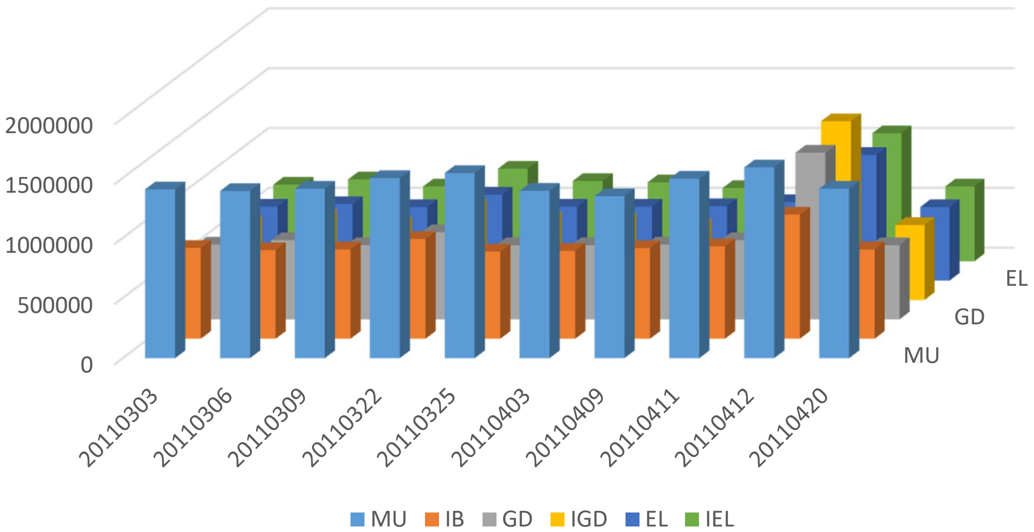
(c) Average arrival interval = 3



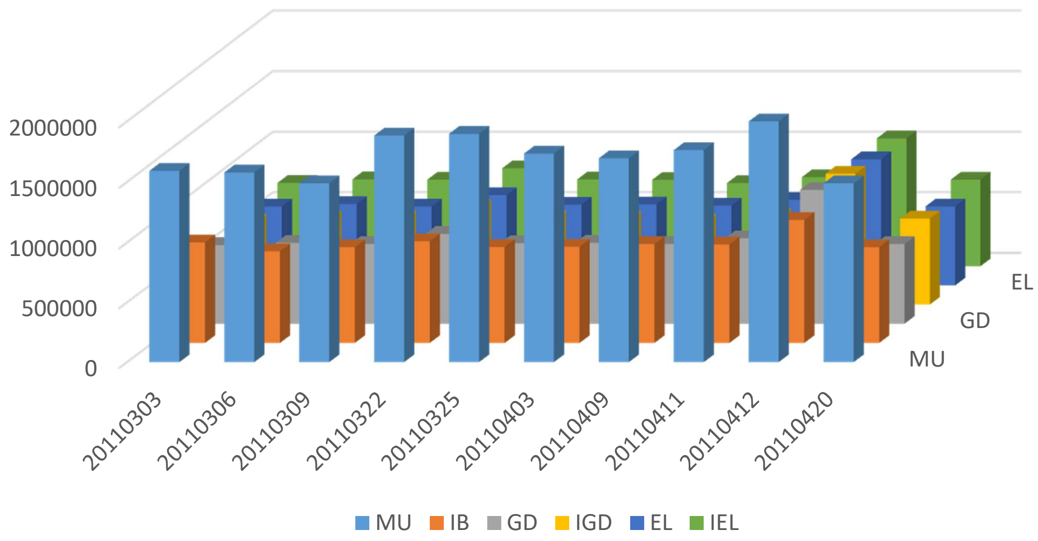
(d) Average arrival interval = 4



(e) Average arrival interval = 5



(f) Mean arrival interval = 6

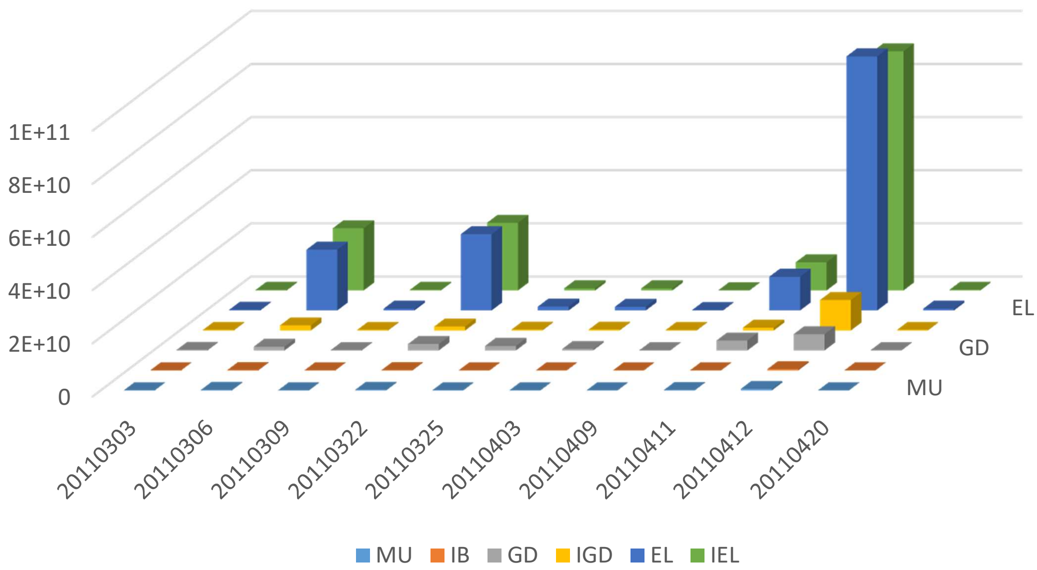


(g) Mean arrival interval = 7

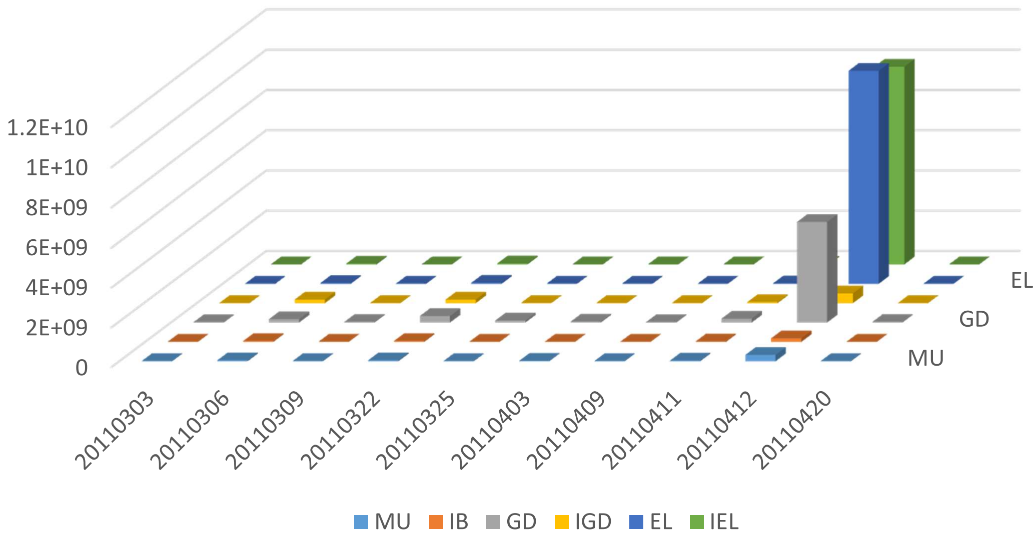
Figure 3.1: Total energy consumption of the selected VM mapping policies running PlanetLab workloads for ten days, when mean arrival intervals range from 1 to 7 seconds

In terms of total execution time of VM mapping policies, when the mean arrival rate ranges from 1 to 7, the same pattern of total energy consumption is observed. Based on Figures 3.2a to 3.2g, EL and IEL that had the longest execution time for most workloads, had the shortest execution time when the mean arrival rate was larger/longer. Moreover, VM mapping policies with relatively short execution time (when the mean arrival interval is short) have a relatively longer execution time when the mean arrival interval is larger/longer.

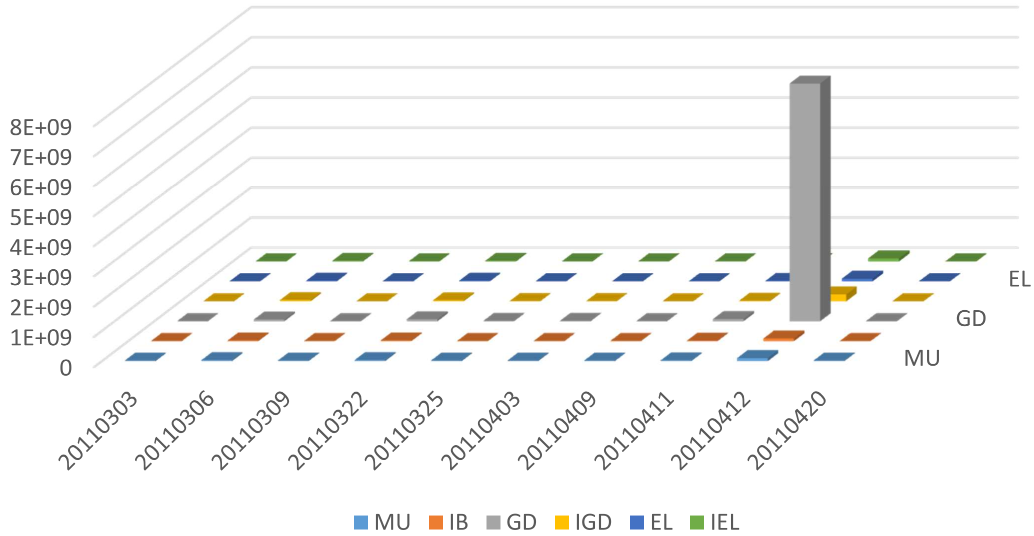
When the mean arrival interval is relatively short, the system is under pressure to map VMs that arrive in relatively short intervals (for instance 1 time unit, every time unit). In such circumstances, IB had the lowest energy consumption and shortest execution time compared to other VM mapping policies. IB is then followed by IGD, GD and MU in both energy consumption and execution time.



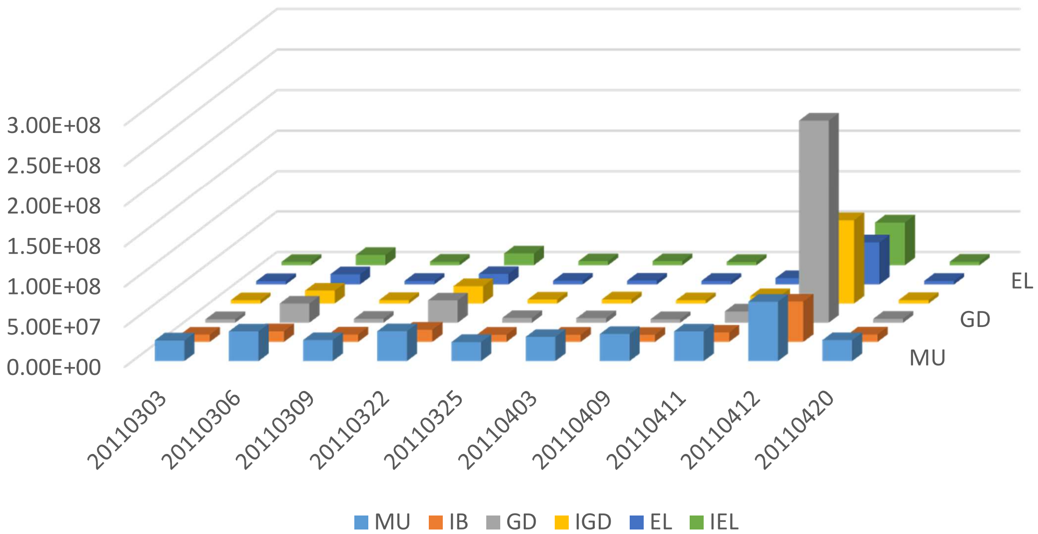
(a) Mean arrival interval = 1



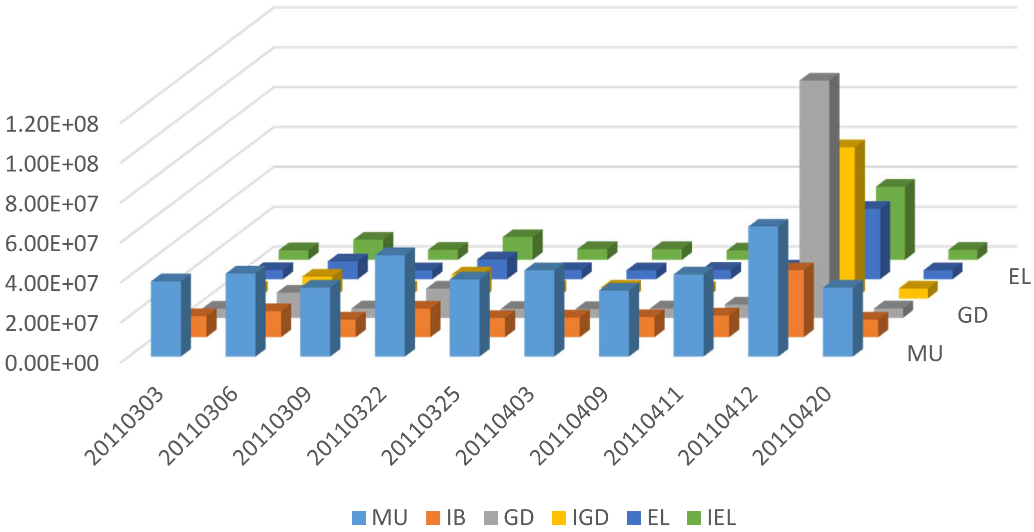
(b) Mean arrival interval = 2



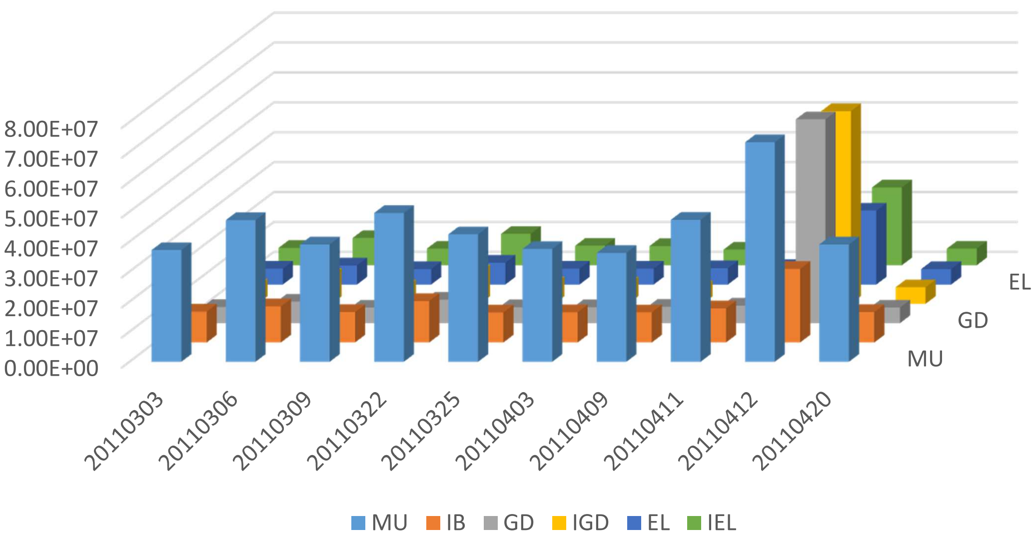
(c) Average arrival interval = 3



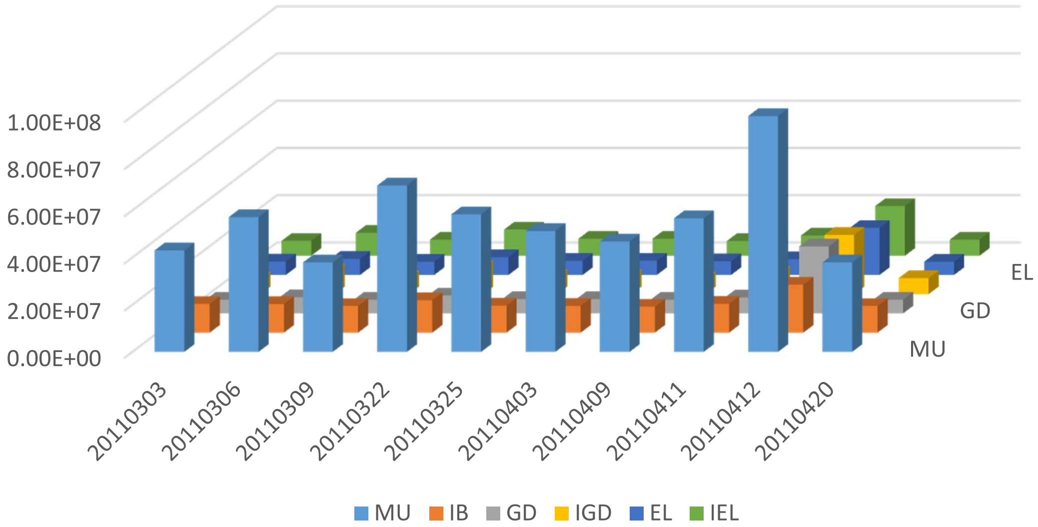
(d) Average arrival interval = 4



(e) Average arrival interval = 5



(f) Mean arrival interval = 6



(g) Mean arrival interval = 7

Figure 3.2: Total execution time of the selected VM mapping policies running PlanetLab workloads for ten days, when mean arrival intervals range from 1 to 7 seconds

Although execution times for IEL and EL were close to other policies, their energy consumption was significantly high. Overall, in a pressurized system, IB represented itself as a desirable option compared to other VM mapping policies in terms of total energy consumption and total execution time. Albeit, EL and IEL were the worst in both energy consumption and execution time.

As the task arrival rate increases, the system receives in a relatively slow speed (for instance every 7 time units), EL and IEL showed substantial improvements in terms of total energy consumption and total execution time. The best performing VM mapping policies in a pressurized system, however, represented a high level of total energy consumption and long total execution time, compared to other VM mapping policies.

3.1.4 Problem Analysis

The results suggested that there might not be a single optimal VM mapping policy that results in an optimal outcome in all circumstances. The alteration on task arrival rate made the system under pressure or relaxed. This changed the outcome of each VM mapping policy. The state of the system, i.e. average utilization level, is another indication of the system being under pressure or relaxed. So the pressure/relaxed state of the system affects the outcome of each VM mapping policy.

To reduce total energy consumption in the system, a VM mapping mechanism can be deployed that switches between available VM mapping policies by choosing a policy that is more likely to reduce total energy consumption given the task arrival rate and the state of the system.

3.2 Adaptive Mechanism for Switching Between VM Mapping Policies

The sub-optimality of VM mapping policies brought up the need for an adaptive VM mapping mechanism that switches between available VM mapping policies according to the information available. Such information includes the mean arrival rate of VMs and the system's current state of resources.

The switching mechanism, between VM mapping policies, can be adaptive. So that the switching decisions correspond to the workload property (arrival rate), current resource state and observed energy consumption records of each VM mapping policy.

Such an adaptive mechanism needs to have a feedback loop to enable learning from the results of its decisions. Therefore, it is given the ability to learn and respond to potential changes in the workload and the system. One of the approaches that can be applied for adaptively switching between VM mapping policies is Bayesian Inference (BI).

3.2.1 Notations

Given $H = \{h_1, h_2, \cdots, h_n\}$ being the list of hosts in the system, each host, h_i, has the following: a list of VMs running on h_i as VM_i and HU_i shows the host's CPU utilization. $VM_i = \{vm_1, vm_2, \cdots, vm_m\}$ is the set of VMs on h_i. Every VM mapping decision, mapping vm_p to h_q, adds vm_p to VM_q.

3.2.2 Constraints

Let $C = \{c_{maxVMs}\}$ denote the constraint set. Based on the virtualization software used, the number of VMs that can be assigned to a host is limited as this constraint is termed as c_{maxVMs}.

3.2.3 Proposed Bayesian Inference

Bayesian inference is based on Bayesian Theorem. Bayesian Theorem and its inference can be graphically presented in a Bayesian Network. First the Bayesian theorem is described. Then the graphical model of our proposed BI for switching between VM policies, its Bayesian Network (BN), is detailed. Then, its components are explained. Later, Bayesian probabilities are calculated according to the training instances, macro level observation records of total energy consumption by VM mapping policies.

3.2.3.1 Bayesian Theorem

Bayesian theorem is a restatement of the conditional probability formula [81]. Bayesian theorem provides a way of updating the probabilities of unobserved outcomes, given that other conditions are met. Nevertheless, the joint probability of conditions must be zero [58].

In Bayesian theorem, there exists the prior probability of the outcome that will be updated (posterior probabilities will be calculated) given the occurrence of the conditions.

The probability of A conditional to B is called posterior probability and is calculated based on Bayesian theorem as equation 3.1.

$$p(A|B) = \frac{p(B|A).p(A)}{p(B)} \tag{3.1}$$

where the probability of A given B, $p(A|B)$, is based on prior probabilities of A, $p(A)$, and B, $p(B)$, and the likelihood of B given A, $p(B|A)$.

To graphically illustrate Bayesian theorem, Bayesian Networks (BNs) are used.

3.2.3.2 Proposed Bayesian Network

A Bayesian Network (BN) represents the conditions and the outcome, and illustrates the relationship between the two. For adaptively switching between available VM mapping policies, BN in Figure 3.3 is proposed.

3.2.3.3 Bayesian Network Components

In Figure 3.3 the proposed Bayesian Network has four nodes, three of which are inputs, AR, CU and policy, and one output, EC. First the Bayesian Network hypothesis should be tested where there should not be dependencies between the contributing factors (AR, CU and Policy). It can be explained that the arrival

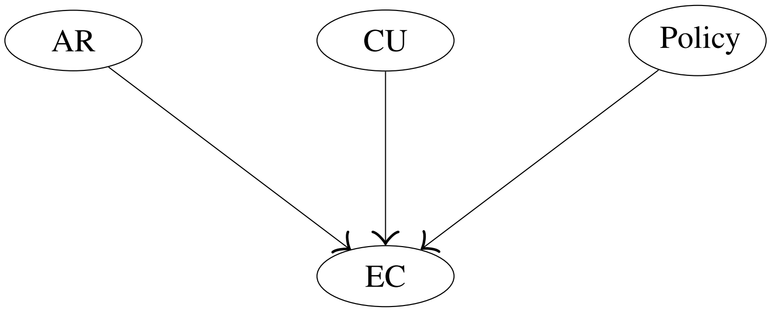


Figure 3.3: Proposed Bayesian Network for switching between mapping policies

rate is one of the workload properties and is not, in any way dependent on either current utilization or the VM mapping policy deployed. The current utilization level is also independent from the arrival rate and the deployed VM mapping policy. And, policy is independent of the other factors. That is, the BN's hypothesis holds for the context of the proposed Bayesian network.

All four nodes in the proposed BN are to be explained and their value ranges specified.

- Input Nodes:
  - ✓ *Arrival Rate (AR):* AR is the mean arrival rate so far into the execution. Its values are set according to the earlier set of simulations where AR is categorized into five ranges. AR_1 is when mean AR is 2 or less units of time. AR_2 is observing mean AR of 3. AR_3 and AR_4 show the AR of 4 and 5, respectively. AR_5 is when AR is 6 or more units of time.
  - ✓ *Current Utilization (CU):* Current Utilization level, CU is the average HU level of all hosts in H. In other words, $CU = \frac{\sum_{i=1}^{n} HU_i}{n}$. CU level is divided into three ranges as: CU_1, CU_2 and CU_3. CU_1 means that the average current utilization level is less than 33% of the full capacity of hosts. CU_2 is when the average current utilization level is between

34% and 66%. CU_3 is when the average current utilization level is 67% and above.

✓ *Policy:* Policy shows the VM mapping policy deployed. According to our earlier simulations, $Pol_1, Pol_2, Pol_3, Pol_4, Pol_5, Pol_6$ represent MU, IB, GD, IGD, EL and IEL, respectively.

- Output Node:

✓ *Energy Consumption (EC):* The values observed in our earlier simulations are divided into three sections and labeled as EC_1, EC_2 and EC_3 to denote low, medium and high energy consumption levels, respectively. The division is based on values only and does not guarantee the same number of instances in each section.

Directed arcs in the proposed BN, Figure 3.3, represent the likelihood of observing a certain level of energy consumption given the arrival rate mean value, AR, and current average utilization level, CU, when a specific VM mapping policy is deployed.

Bayesian probabilities, prior probabilities and likelihoods, should be calculated to be used in equation 3.1 to calculate posterior probabilities. Prior probabilities for each arrival rate, utilization level and VM mapping policies are illustrated in Figure 3.4.

The likelihood of observing certain levels of energy consumption given each possible combination of arrival rate, utilization level and VM mapping policy is shown in Figure 3.5.

Each component (node) in the proposed Bayesian network should be explained. Their value ranges are to be specified.

AR

Arrival Rate (AR)				
AR_1 :	AR_2 :	AR_3 :	AR_4 :	AR_5 :
$AR \leq 2$	$AR = 3$	$AR = 4$	$AR = 5$	$AR \geq 6$
p(AR_1)	p(AR_2)	p(AR_3)	p(AR_4)	p(AR_5)

(a) Prior probabilities for observing mean arrival rates: 2 or less; 3; 4; 5; and 6 or more.

CU

Current Utilization (CU)		
CU_1:	CU_2:	CU_3:
$\leq 33\%$	$34\% - 66\%$	$\geq 67\%$
$p(CU_1)$	$p(CU_2)$	$p(CU_3)$

(b) Prior probabilities for average CPU utilization of certain ranges

Policy

VM mapping policies					
Pol_1:MU	Pol_2:IB	Pol_3:GD	Pol_4:IGD	Pol_5:EL	Pol_6:IEL
p(Pol_1)	p(Pol_2)	p(Pol_3)	p(Pol_4)	p(Pol_5)	p(Pol_6)

(c) Prior probabilities for each VM mapping policy

Figure 3.4: Bayesian prior probabilities

EC

			Energy Consumption (EC)		
AR	CU	policy	EC_1: low	EC_2: medium	EC_3: high
AR_1	CU_1	Pol_1	$\mathrm{p}(EC_1 \mid AR_1, CU_1, Pol_1)$	$\mathrm{p}(EC_2 \mid AR_1, CU_1, Pol_1)$	$\mathrm{p}(EC_3 \mid AR_1, CU_1, Pol_1)$
AR_1	CU_1	Pol_2	$\mathrm{p}(EC_1 \mid AR_1, CU_1, Pol_2)$	$\mathrm{p}(EC_2 \mid AR_1, CU_1, Pol_2)$	$\mathrm{p}(EC_3 \mid AR_1, CU_1, Pol_2)$
AR_1	CU_1	Pol_3	$\mathrm{p}(EC_1 \mid AR_1, CU_1, Pol_3)$	$\mathrm{p}(EC_2 \mid AR_1, CU_1, Pol_3)$	$\mathrm{p}(EC_3 \mid AR_1, CU_1, Pol_3)$
AR_1	CU_1	Pol_4	$\mathrm{p}(EC_1 \mid AR_1, CU_1, Pol_4)$	$\mathrm{p}(EC_2 \mid AR_1, CU_1, Pol_4)$	$\mathrm{p}(EC_3 \mid AR_1, CU_1, Pol_4)$
AR_1	CU_1	Pol_5	$\mathrm{p}(EC_1 \mid AR_1, CU_1, Pol_5)$	$\mathrm{p}(EC_2 \mid AR_1, CU_1, Pol_5)$	$\mathrm{p}(EC_3 \mid AR_1, CU_1, Pol_5)$
AR_1	CU_1	Pol_6	$\mathrm{p}(EC_1 \mid AR_1, CU_1, Pol_6)$	$\mathrm{p}(EC_2 \mid AR_1, CU_1, Pol_6)$	$\mathrm{p}(EC_3 \mid AR_1, CU_1, Pol_6)$
AR_1	CU_2	Pol_1	$\mathrm{p}(EC_1 \mid AR_1, CU_2, Pol_1)$	$\mathrm{p}(EC_2 \mid AR_1, CU_2, Pol_1)$	$\mathrm{p}(EC_3 \mid AR_1, CU_2, Pol_1)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
AR_5	CU_2	Pol_6	$\mathrm{p}(EC_1 \mid AR_5, CU_2, Pol_6)$	$\mathrm{p}(EC_2 \mid AR_5, CU_2, Pol_6)$	$\mathrm{p}(EC_3 \mid AR_5, CU_2, Pol_6)$
AR_5	CU_3	Pol_1	$\mathrm{p}(EC_1 \mid AR_5, CU_3, Pol_1)$	$\mathrm{p}(EC_2 \mid AR_5, CU_3, Pol_1)$	$\mathrm{p}(EC_3 \mid AR_5, CU_3, Pol_1)$
AR_5	CU_3	Pol_2	$\mathrm{p}(EC_1 \mid AR_5, CU_3, Pol_2)$	$\mathrm{p}(EC_2 \mid AR_5, CU_3, Pol_2)$	$\mathrm{p}(EC_3 \mid AR_5, CU_3, Pol_2)$
AR_5	CU_3	Pol_3	$\mathrm{p}(EC_1 \mid AR_5, CU_3, Pol_3)$	$\mathrm{p}(EC_2 \mid AR_5, CU_3, Pol_3)$	$\mathrm{p}(EC_3 \mid AR_5, CU_3, Pol_3)$
AR_5	CU_3	Pol_4	$\mathrm{p}(EC_1 \mid AR_5, CU_3, Pol_4)$	$\mathrm{p}(EC_2 \mid AR_5, CU_3, Pol_4)$	$\mathrm{p}(EC_3 \mid AR_5, CU_3, Pol_4)$
AR_5	CU_3	Pol_5	$\mathrm{p}(EC_1 \mid AR_5, CU_3, Pol_5)$	$\mathrm{p}(EC_2 \mid AR_5, CU_3, Pol_5)$	$\mathrm{p}(EC_3 \mid AR_5, CU_3, Pol_5)$
AR_5	CU_3	Pol_6	$\mathrm{p}(EC_1 \mid AR_5, CU_3, Pol_6)$	$\mathrm{p}(EC_2 \mid AR_5, CU_3, Pol_6)$	$\mathrm{p}(EC_3 \mid AR_5, CU_3, Pol_6)$

Figure 3.5: Likelihood of observing a certain level of energy consumption - low, medium or high - given the observed arrival rate, average CPU utilization and deployed VM mapping policy from the training instances

3.2.3.4 Calculating Probabilities

In Figure 3.4a, $p(AR_i)$ is prior probability for AR_i. Accordingly, $p(CU_j)$ in Figure 3.4b and $p(Pol_k)$ in Figure 3.4c are prior probabilities for observing average utilization levels of CU_j and deployment of pol_k. Prior probabilities are calculated based on the training set, comprising training instances. Prior probability are the number of instances from the presumed element. For instance, prior probability for AR_1 is the number of training instances with their arrival rate of 2 or less, divided by the total number of instances. If in a training set of 500 instances there are 126 instances where the mean arrival rate was 2 or less, the prior probability for AR_1, $p(AR_1)$ will be $\frac{126}{500} = 0.252$.

According to the Bayesian theorem in Section 3.2.3.1, probability of a certain energy consumption level, EC_p, conditional to AR_i, CU_j and Pol_k will be calculated according to equation 3.2.

$$p(EC_p|AR_i,CU_j,Pol_k) = \frac{(p(AR_i,CU_j,Pol_k|EC_p)).(p(EC_p))}{p(AR_i,CU_j,Pol_k)} \tag{3.2}$$

$p(AR_i,CU_j,Pol_k)$ and $p(EC_p)$ are prior probabilities. $p(EC_p)$ is the number of training instances with an energy consumption level within EC_p range divided by the total number of training instances. To calculate prior probability for each AR_i,CU_j,Pol_k combination in Figure 3.4, $p(AR_i,CU_j,Pol_k)$, algorithm 3.1 can be used.

Prior probabilities for each energy consumption level, EC_1, EC_2 and EC_3, can be calculated as the number of training instances with energy consumption level of that range divided by total number of training instances.

Equation 3.2 also needs the likelihood of observing AR_i,CU_j,Pol_k combination when the result was an energy consumption level within EC_p range, $p((AR_i, CU_j,Pol_k)|EC_p)$. Algorithm 3.2 calculates $p((AR_i,CU_j,Pol_k)|EC_p)$.

After calculating prior probabilities for each possible combination of AR, CU

Algorithm 3.1 Prior probability for (AR_i, CU_j, Pol_k) combination, $p(AR_i, CU_j, Pol_k)$

for each instance **do**
    if Arrival rate is within AR_i range **then**
        if Current utilization is within CU_j range **then**
            if Deployed policy is Pol_k **then** increase $counter$
$p(AR_i, CU_j, Pol_k) = \frac{counter}{total\ number\ of\ instances}$

Algorithm 3.2 Calculating likelihoods, $p(AR_i, CU_j, Pol_k | EC_p)$

for each instance **do**
    if Arrival rate is within AR_i range **then**
        if Current utilization is within CU_j range **then**
            if Deployed policy is Pol_k **then** {
                increase $counter_{ijk}$
                if Energy consumption level is within EC_p range **then**
                    increase $counter_{ijkp}$
            }
$p(AR_i, CU_j, Pol_k | EC_p) = \frac{counter_{ijkp}}{counter_{ijk}}$

and *Pol*; and the likelihood of each combination of these combinations leading to certain energy consumption levels, posterior probabilities can be calculated based on equation 3.2.

Bayesian inference is meant to seek energy efficiency. Therefore, the first desirable EC option is $EC =$ low. In other words, the probability of achieving $EC =$ low, the third column in the bottom in Figure 3.3, is the first point of decision. Among possible combinations in the corresponding column, a combination of AR_i, CU_j and Pol_p with the highest probability is the chosen option for selecting a VM mapping policy.

3.3 Evaluation

Our proposed adaptive switching mechanism, between VM mapping policies, is evaluated against individual VM mapping policies. In this section, experimental settings, simulation constraints and the results are explained.

3.3.1 Experimental Settings

To evaluate our adaptive switching mechanism, resource requests of workloads from PlanetLab [76] project are used. Selected days are the days included in the CloudSim simulation package [14, 13] [1]. We base our evaluation on these days to facilitate the reproduction of the results. However, the workloads do not belong to a private Cloud, and tasks do not have a deadline [2]. Uniformly random numbers are associated to tasks as deadlines. The intensity of tasks is determined by processor to memory request ratios. if the processor to memory request is

[1] CloudSim could not be used for evaluating our switching mechanism as it does not support the change in VM mapping policy at run time.

[2] GD and IGD mapping policies use tasks' deadlines to decide the mapping of the corresponding VM.

above 10, it is labeled as processing-intensive and if the memory to processor request is 10 or above it is memory-intense. Otherwise tasks are considered as non-intensive.

There are 20 homogeneous hosts. This eliminates the effect of different host capacities on the outcome. The initial available bandwidth to each host is set to 1000 Hertz. The experiments are conducted on an Intel Core i7 CPU machine, running Windows 10.

Our adaptive policy switching mechanism starts with equal probabilities for all combinations of arrival rate (AR), current utilization (CU) and policies (Pol) until the first 100 training instances are collected. The simulation is run with 100, 300 and all training instances for Bayesian probabilities are calculated. Including all instances made the calculation of mechanism a lengthy process and was not more effective than smaller training sets. The results for 100 and 300 instances were consistent. Thus, the results of simulation for updating Bayesian probabilities with 300 latest instances are reported as a representative.

Energy consumption is calculated based on the latest energy model proposed by Hsu [39] that models the energy consumed by a VM based on its utilization level as equation 2.2.

3.3.2 Constraints

Our simulation puts constraints on the maximum number of VMs that can be mapped to a host. If the set of constraints is C, $c_{MaxVMs} \in C$ is the constraint on the maximum number of VMs on a host. In our simulations $c_{MaxVMs} = 10$.

3.3.3 Results

Our Adaptive Switching Mechanism (ASM) is compared with individual mapping policies in Figure 3.6 when normalized values for energy consumption and execution time are added and are box plotted.

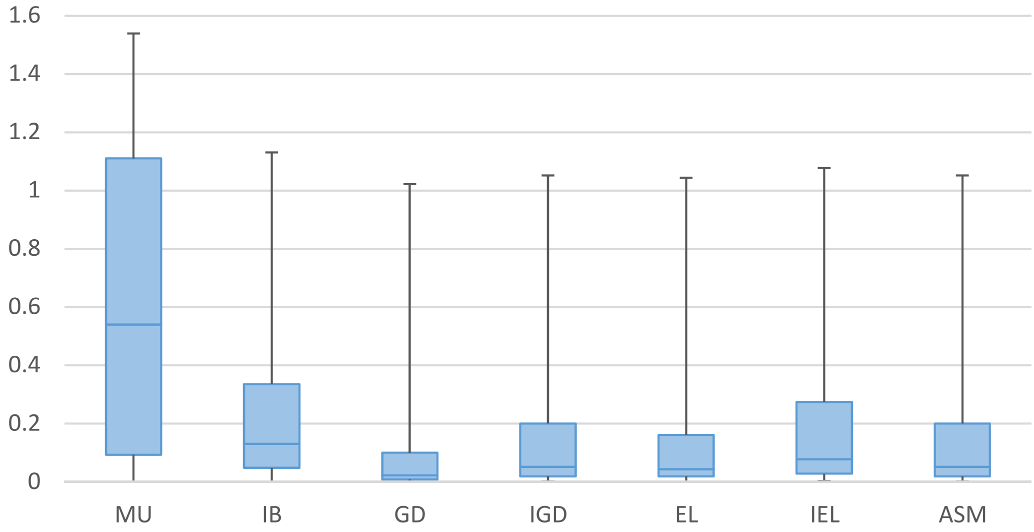


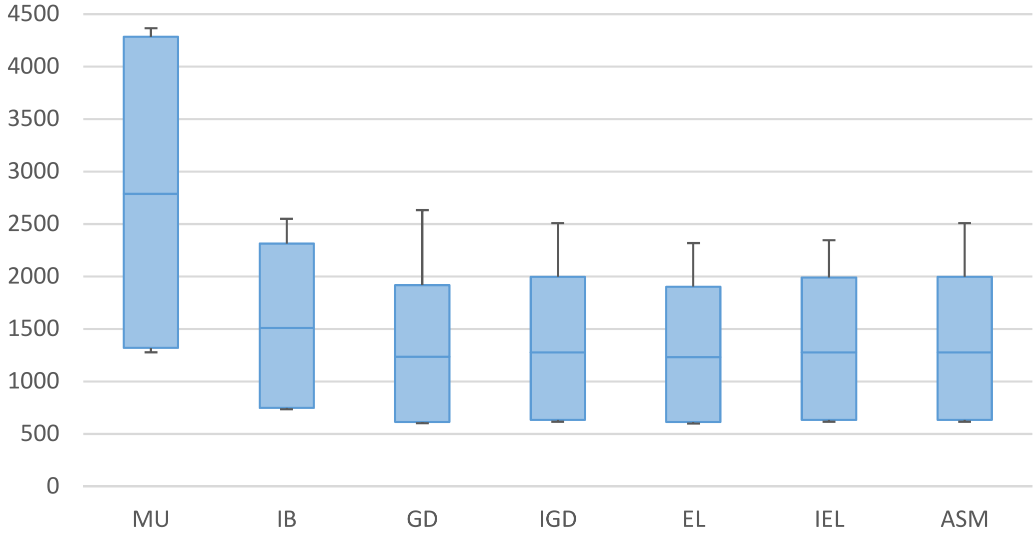
Figure 3.6: Average normalized energy consumption and execution time values

Figure 3.7 illustrates the results in terms of total energy consumption and execution time separately and respectively.

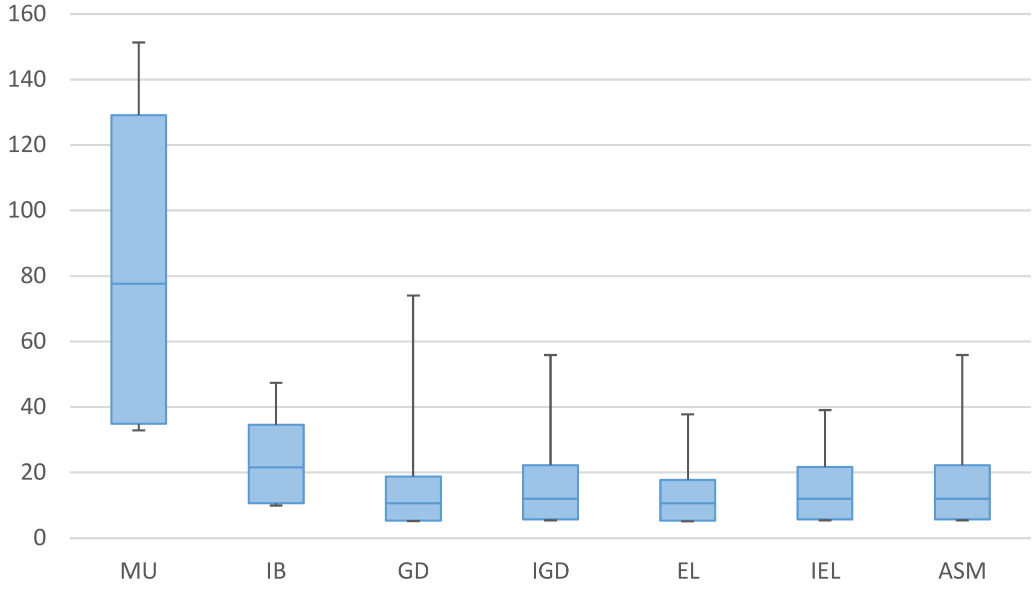
According to Figure 3.7a and Figure 3.7b, ASM had results close to the best performing VM mapping policies. Because the best VM mapping policies for the given workload and arrival rate are unknown, ASM achieved a result, in terms of both total energy consumption and total execution time, that is close to the best performing VM mapping policies.

In order to statistically compare the results, first descriptive statistics of observed total energy consumption and total execution times are presented in Tables 3.1 and 3.2.

According to Tables 3.1 and 3.2 ASM had results close to other VM mapping policies. To compare the policies with ASM in greater depth, Wilcoxon Sign Ranked Test is used that compares the paired values of ASM and a competing VM mapping policy. It also shows the significance differences between the results of VM mapping policies compared to ASM. Wilcoxon Ranked Test of the results



(a) Energy consumption (kW)



(b) Total execution time (M time units)

Figure 3.7: Energy consumption and execution time by ASM compared to individual VM mapping policies

	Mean	St. Deviation	Minimum	Maximum
MU	1429312.80	102338.628	1277019	1577763
IB	793068.60	91531.454	734724	1041399
GD	715263.30	243964.272	600960	1398795
IGD	721486.80	187233.004	615459	1233519
EL	678611.10	148607.858	597390	1088202
IEL	701596.20	137015.943	615459	1069503
ASM	721486.80	187233.004	615459	1233519

Table 3.1: Statistical description of total energy consumption by VM mapping policies and ASM

	Mean	St. Deviation	Minimum	Maximum
MU	45036915.50	12378588.52	32873471	73648825
IB	12826670.30	4696699.442	9914866	25726635
GD	11801678.50	18185274.35	5186810	63425053
IGD	10825896.80	11773617.94	5397144	43846947
EL	7947805.40	6818366.759	5122930	27174030
IEL	9023479.50	6570381.675	5397144	27035076
ASM	10825896.80	11773617.94	5397144	43846947

Table 3.2: Statistical description of total execution time by VM mapping policies and ASM

of paired comparisons are reported in Table 3.3 for total energy consumption and total execution time.

	Asympt. Sig. (2-tailed), total energy consumption	Asympt. Sig. (2-tailed), total execution time
MU - ASM	.005 (< .05)	.005 (< .05)
IB - ASM	.074 (> .05)	.074 (> .05)
GD - ASM	.074 (> .05)	.074 (> .05)
IGD - ASM	1.000 (> .05)	1.000 (> .05)
EL - ASM	.005 (< .05)	.005 (< .05)
IEL - ASM	.109 (> .05)	.109 (> .05)

Table 3.3: Results of the Wilcoxon Signed Ranks test for total energy consumption (kW) and total execution time

An asymptotic significance value of less than .05 rejects the null hypothesis, that the compared pairs are statistically similar. Therefore, the results of ASM are statistically (based on paired value comparison of all workloads and mean arrival intervals) not different from IB, GD, IGD and IEL in terms of total energy consumption and total execution time. However, ASM is statistically different from MU and EL because of the asymptotic significance value of less than .05.

The results of the statistical test show that ASM is significantly better statistically than MU, but has no statistically different results than IB, GD, IGD and IEL, and is worse than EL in terms of energy consumption and execution time. It is important to note that ASM has closer results to the best performing policy, EL. EL on average has only 5% lower energy consumption than ASM. And these results are achieved in the absence of prior knowledge about the performance of each VM mapping policy and its switching overhead. The switching overhead makes it difficult for ASM to outperform all individual policies. However, the switching capability facilitates the application of such methodology when there is not prior knowledge about the effectiveness of the existing policies.

To further understand the underlying reasons for the observed results, Figure 3.8 illustrates the number of host switches in our proposed VM mapping mechanism and all individual VM mapping policies.

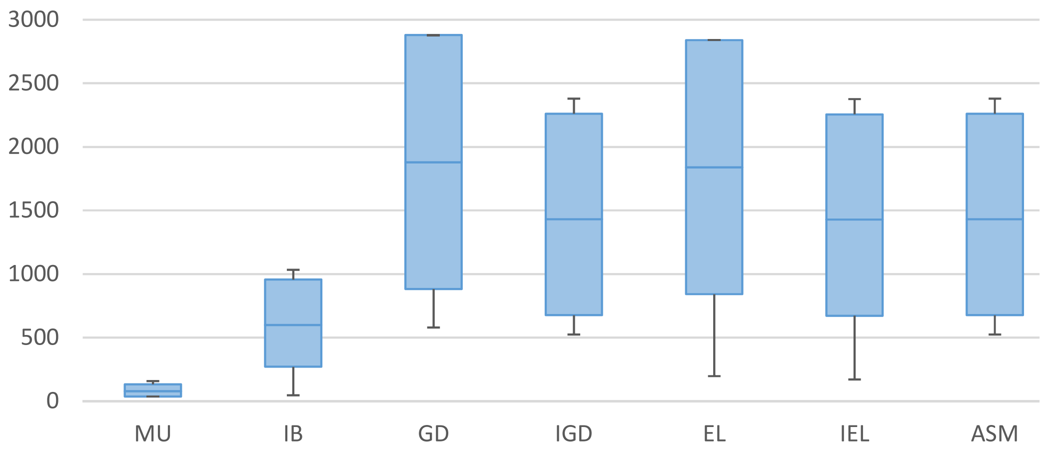


Figure 3.8: Number of host shutdowns by ASM in comparison to all individual VM mapping policies

Given that VMs are not moved/migrated between hosts after their mapping is done, a host can only be switched off if it has finished the execution of all its VMs. Based on the number of host shutdowns reported in Figure 3.8 it can be concluded that MU performs the fewest host shutdowns, seemingly keeping a large number of hosts in their active state. EL and GD on the other hand perform a relatively high number of host shutdowns. Despite host start-ups (after a shutdown) adding to the total energy consumption, EL is proved to make energy efficient decisions so that compensates for the energy drawback of the host switches compared to other policies. ASM exhibits a similar number of host shutdowns as GD but has a lower energy consumption level than GD. In summary, the number of host shutdowns (when no VM is moved/migrated between hosts) correlates neither directly with the deployed VM mapping policy's energy consumption nor execution time.

3.4 Discussion

In the absence of knowledge about the best VM mapping policy for the given state of the system and workload properties, our Adaptive Switching Mechanism (ASM) had results close to the best performing VM mapping policies on ten workloads when mean arrival interval ranged from 1 to 7 time units.

ASM had statistically similar results to IB, GD, IGD and IEL in terms of both total energy consumption and total execution time. ASM was statistically different from MU and EL. On average, MU had close to 198% higher total energy consumption and took 412% longer to execute the workloads than ASM. EL, on average, had a slightly lower energy consumption (5%) and shorter (26%) execution time than ASM.

Based on Figure 3.6 that reported the accumulated normalized values of total energy consumption and total execution time, the best performing VM mapping policy, on average, was GD. And based on the statistical analysis performed, the results of ASM is not statistically different from GD. Moreover, ASM represented closeness to the best performing VM mapping policies in the absence of prior knowledge about the mean arrival interval.

However, in some cases, ASM has slightly higher total energy consumption and longer total execution time than some VM mapping policies. The reason can be tracked back to the time observations are recorded. Bayesian inference used in AMS records a macro level observation, total energy consumption, as its deciding factor. It means our Bayesian probabilities are updated after finishing execution of a workload and recording total energy consumption given the conditions, in terms of arrival rate, average utilization and the deployed VM mapping policy. It delays the act of updating Bayesian probabilities, therefore seemingly extending the effect of the observed outcome being reflected on in the short term decision making phase.

A faster and more efficient way of updating our Bayesian probabilities can further improve the result in terms of both total energy consumption and total execution time. One way is to have run-time measures such as changes in the average utilization level throughout the execution period (instead of at the end of executing a workload). This can boost the ability of the system to respond to changes in a faster pace. This approach is followed in the following chapters.

3.5 Summary

In this chapter, a novel mechanism is presented that adaptively switches between available VM mapping policies to reduce total energy consumption. It also shortened the execution time. In the absence of knowledge about the best VM mapping policy, given the state of the system and workload properties, our proposed adaptive switching mechanism illustrated results close to the optimal VM mapping policy.

Our proposed Bayesian inference records total energy consumption, macro level observation records, and updates its probabilities accordingly. A faster update of Bayesian probabilities based on observed changes in the system state at run-time can further improve the adaptability as well as total energy consumption and execution time.

This chapter detailed the invention of an adaptive mechanism that switches between available mapping policies to reduce energy consumption in private Clouds.

Chapter 4

ENERGY-EFFICIENT VM MIGRATION POLICIES IN PRIVATE CLOUDS

MAPPING policies determine the way VMs (and their tasks) are mapped to hosts. Based on the observed results of multiple VM mapping policies in the previous chapter, it can be concluded that each VM mapping policy has its strengths and weaknesses in a given state of the system and workload properties. In the last chapter we proposed a dynamic switching mechanism that switches between VM mapping policies when they are likely to reduce energy consumption. In the absence of prior knowledge about the outcome of available VM mapping policy, in terms of energy consumption, our proposed mechanism demonstrated results close to the policy with lowest energy consumption and shortest execution time.

Despite the VM mapping mechanism having close results to the policy with the lowest energy consumption and shortest execution time, the results were not promising in terms of a guaranteed energy efficiency for every given system setting and workload properties. To further improve energy efficiency, VMs can be migrated between hosts when there is a potential for energy reduction. By migrating VMs, the utilization of the running hosts can be increased (given it does not exceed their resource capacity). Performing VM migration can facilitate the process of switching some hosts off to save energy.

VM migration policies should be developed to answer the research question 3) *How is resource hyper-visioning extended to VM migration, if energy-efficient mapping decisions caused imbalance, to reduce energy consumption in private Clouds?*. We propose VM migration policies where they utilize micro level observations e.g. VMs' memory size and utilization level.

In this chapter, VM migration policies are proposed that select VMs to migrate. Our proposed VM migration policies outperformed state-of-the-art policies. The proposed policies use micro level observation records, in contrast to our VM mapping switching mechanism that used macro level observation records of total energy consumption.

Proposed VM migration policies are evaluated against a state-of-the-art VM migration policy and are proved to outperform it significantly in terms of energy consumption and execution time. The results are reported in our paper [87].

This chapter is organized as follows. Section 4.1 reviews the VM migration concept. Proposed VM migration policies are detailed in Section 4.2 and evaluated in Section 4.3 including the results in terms of energy consumption and mean execution time in comparison to state-of-the-art policies. It is then followed by a discussion on the results in Section 4.4. A summary of the chapter is presented in Section 4.5.

4.1 VM Migration

VM migration is described as the process of moving or migrating a VM from one host to another. When VM migration is performed without interupting the execution in hosts (either source or destination hosts), except the short time for transferring the VM status, it is called live migration [18]. In this study VM migration refers to VM live migration.

VM migration is a solution to managing imbalance problems in the system. An imbalance problem refers to the state of system where some hosts are either over-loaded or under-loaded.

An over-loaded host is shown to be more likely to face hardware failure [25]. VM migration eases the imbalance problem by migrating VMs from over-loaded hosts. VM migration, also, migrates VMs from an under-loaded host in order to switch this host into idle mode. In a system with under-loaded hosts, VM migration provides the system with the option of migrating the VMs from the under-loaded host and switching them off to save energy.

4.1.1 Notations

Given $H = \{h_1, h_2, \cdots, h_n\}$ being the list of hosts in the system, VM_i represents the VMs running on h_i and VM_j is the VMs running on h_j. When migrating $vm_p \in VM_i$ from host h_i to host h_j, VM migration removes vm_P from VM_i set and adds it to VM_j, if vm_p is a migratable VM. A complete set of notations is provided in Appendix A.

If MVM represents the set of migratable VMs from the hosts with imbalance problem, VM migration can be defined as $Mig = \{mig | mig : MVM \to H\}$, where Mig is a set of combinations of mvm_q and h_j where mvm_q is removed from the set of its host's VMs and added to VM_j.

Because MVM and H are finite sets, Mig is also a finite set. A migration map can be $Mig = \{(mvm_1, h_4), (mvm_1, h_7), , (mvm_2, h_4), \cdots, (mvm_r, h_k)\}$. A host, e.g. h_k, might appear more than once, if it is a suitable destination host for multiple mvms.

4.1.2 VM Migration Process

To ease the imbalance problem on a host, a VM from the imbalanced host can be migrated to another host that does not have an imbalance problem. To migrate a VM from a host to another, first a VM migration policy should be applied to select one VM (migratable VM) from the list of migratable VMs running on the hosts with imbalance problem. VM migration policy affects energy consumption by the number of VM migrations that need to be performed to solve the imbalance problem.

VM migration policy can utilize the information available to the system when aiming to reduce total energy consumption. Available information includes micro level observation records of VMs' current utilization level, memory size and available bandwidth between source and potential destination host.

Basing VM migration decisions on these micro level observation records facilitates the move toward energy reduction while migrating VMs to ease the imbalance problem in the system. The number of VM migrations performed to ease imbalance affects total energy consumption and execution time because of each VM migration's energy and time drawbacks.

4.1.3 VM Migration Drawbacks

A VM migration entails time and energy drawbacks. These drawbacks relate to the time it takes to copy VM memory and VM's computation status from its current (source) host to the destination host.

VM migration time is calculated by Beloglazov and Buyya [9] as VM's memory size divided by available bandwidth between source and destination hosts. The energy drawback of a VM migration is due to the required action of copying the VM memory from the source host to destination host through network links. Moreover, frequent migration of VMs on an under-loaded hosts might entail frequent host switches (on/off) that itself adds to total energy consumption because of start-up energy peaks.

VM migration eases the imbalance problem in the system, however, each VM migration adds to the execution time and energy consumption. As a result, efficient VM migration policies are required to solve imbalance problems where the migration drawbacks of energy peaks and extended execution (due to copying VMs through network) are compensated by a reduction in total energy reduction and the shortening of the total execution time.

4.2 VM Migration Policies

In this section two VM migration policies are introduced. These policies aim at migrating VMs to as fewer hosts as possible to potentially switch some hosts off and reduce total energy consumption. These policies utilize micro level observations from the VMs and intend to minimize the number of VM migrations required for solving the imbalance problem in the system.

Every VM migration entails potential energy and time drawbacks. To reduce energy consumption, number of VM migrations can be minimized while easing the imbalance problem. It is expected to improve total energy consumption and execution time.

Two VM migration policies are proposed in order to reduce total energy consumption and shorten the execution time. Section 4.2.1 explains the details of one of the proposed VM migration policies, namely MedianMT. MedianMT bases its decisions on the memory size of VMs and available network bandwidth to hosts to select a VM for migration.

In Section 4.2.2 we propose another VM migration policy, namely MaxUtil that decides upon the selection of a VM for migration according to its current utilization (processing) level.

4.2.1 Median of Migration Times (MedianMT)

Once an imbalance problem is detected, a set of migratable VMs, MVM, from hosts with imbalance problems should be gathered. To estimate the time it will take to migrate a given VM, migration time, MT, can be calculated for VMs in MVM using equation 4.1, where M_i is the memory size of vm_i and NB_{host} is the network bandwidth available to the host.

$$MT = \frac{M_i}{NB_{host}} \quad [9] \tag{4.1}$$

After calculating the migration times for VMs in MVM, the list of VMs' migration times is sorted. The middle element in the list has the median value for migration time. MedianMT VM migration policy selects the VM with median migration time to be moved/migrated.

4.2.2 Maximum Utilization (MaxUtil)

In order to ease the imbalance by performing fewer migrations, we propose a VM migration policy that selects a VM with the highest utilization level to be migrated (selected from MVM). MaxUtil works based on the assumption that migrating a VM with a high current utilization level will ease the imbalance problem by fewer migrations, therefore minimizing the need for more migrations.

Minimization of the number of migrations is expected to lead to the reduction in energy consumption and execution time by avoiding frequent VM migrations. Avoiding frequent VM migrations minimizes the drawbacks associated with VM migration in terms of energy consumption and execution time.

4.3 Evaluation

Our proposed VM migration policies, MedianMT and MaxUtil, are evaluated against the best VM migration policy developed by Beloglazov and Buyya [9], namely Minimum Migration Time (MMT) when combined with Threshold based VM mapping policy, Thr [9].

4.3.1 Experimental Settings

We simulated a Cloud system with 800 heterogeneous hosts. Half of the hosts are HP ProLiant ML110 G4 hosts, while the other half consists of HP ProLiant ML110 G5 hosts. Each host is assumed to have 1 GB network bandwidth.

The characteristics of the VM types correspond to Amazon EC2 instance types with the only exception that all the VMs are single-core, which is explained by the fact that the workload data used for the simulations comes from single-core VMs.

The proposed VM migration policies are compared with the best policy proposed by Beloglazov and Buyya [9], Minimum Migration Time, MMT. Each VM migration policy, MedianMT, MaxUtil and MMT, is combined with VM mapping policy Thr. Thr is a threshold based VM mapping policy. Its threshold values are set to 0.6, 0.7, 0.8, 0.9 or 1 as in the original paper [9].

All VM migration policies are deployed when the same VM mapping policy, Thr [9], is deployed. It is to represent the effects of VM migration policies on energy consumption level, regardless of the VM mapping policy.

Workload sets are from PlanetLab [76] project. We use the workloads of the selected ten days included in CloudSim [14, 13] to enable the reproduction of the results.

4.3.2 Constraints

During the simulation there is a constraint on the hosts that should receive the migrating VM.

If C represents the set of simulation constraints, $C = \{c_{ExcludedHosts}\}$. $c_{ExcludedHosts}$ denotes the set of hosts that should not receive the migrating VMs because they are in an imbalanced state themselves. That is, a host that has imbalance problem is not a suitable candidate to receive a migrating VM.

4.3.3 Results

The results of our proposed VM migration policies are compared with 1, no VM migration is performed to highlight the impact of VM migration on total energy consumption, 2, the best VM migration policies proposed by Beloglazov and Buyya [9], Minimum Migration Time (MMT) in terms of their total energy consumption, mean execution time as well as the number of host shutdowns and the number of VM migrations.

Figure 4.1 illustrates total energy consumption by VM mapping policy Thr [9] when no migration is performed and when competing VM migration policies (MMT, MedianMT and MaxUtil) are applied after the VM mapping by Thr [9]. The result of VM mapping policy, in the absence of VM migration, is illustrated to accentuate the impact of including VM migration in resource hyper-visioning. The results are reported when the VM mapping policy threshold value ranges from 0.6 to 1 (portion of total host capacity).

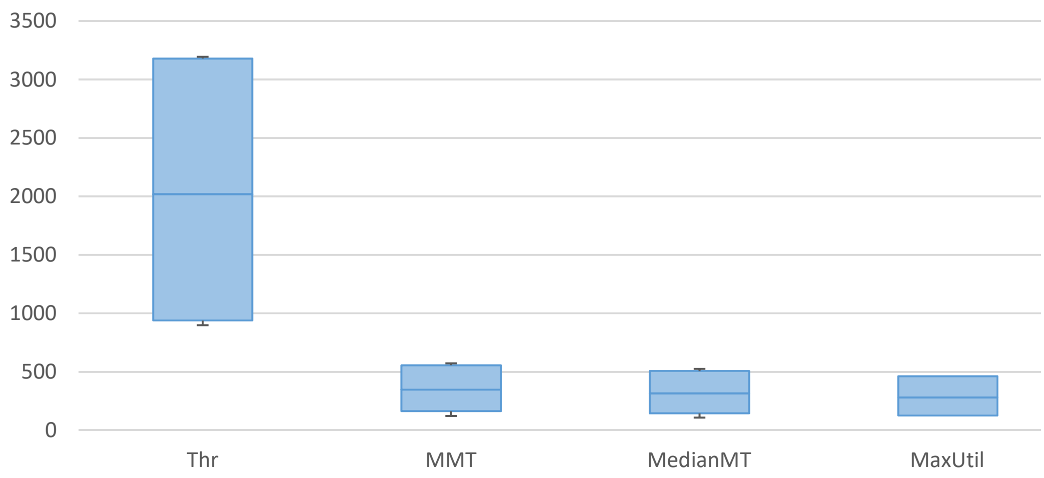


Figure 4.1: Total energy consumption (kW) for VM mapping policy Thr (in the absence of VM migration) and when VM migration policies MMT, MedianMT and MaxUtil are deployed

Figure 4.1 clearly demonstrates the significant improvement made by performing VM migration after VM mapping in terms of total energy consumption.

Mean execution time for Thr [9] VM mapping policy and when VM mapping policies MMT [9], MedianMT and MaxUtil are deployed is reported in Figure 4.2.

According to Figure 4.2 performing VM migration led to significant shortening of mean execution time in the system.

However, the scale of the box plots in Figures 4.1 and 4.2 does not enable the comparison between the VM migration policies.

The result of total energy consumption for VM migration policies MMT [9], MedianMT and MaxUtil are reported in Figure 4.3 where Figure 4.3a represents the results based on the threshold values and Figure 4.3b is the box plots of all total energy consumption values by each VM migration policy.

According to Figure 4.3 MedianMT outperforms MMT [9] both on every threshold value and as depicted in box plots. MaxUtil outperforms MMT while

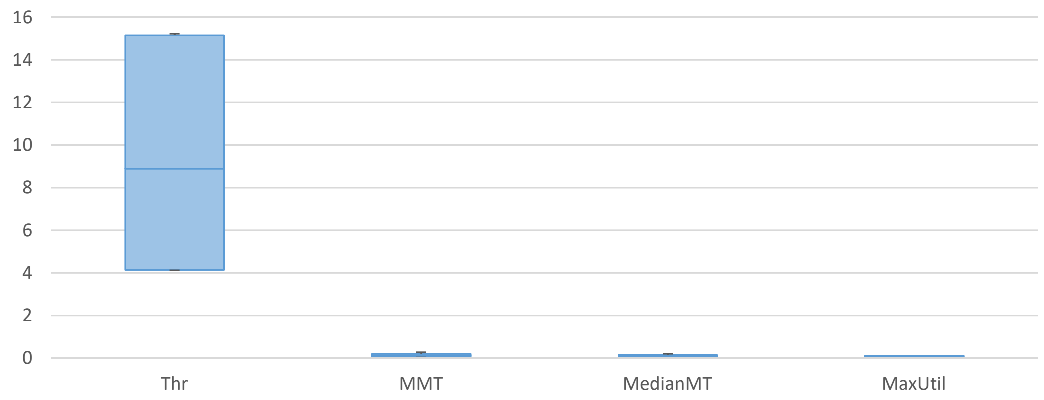
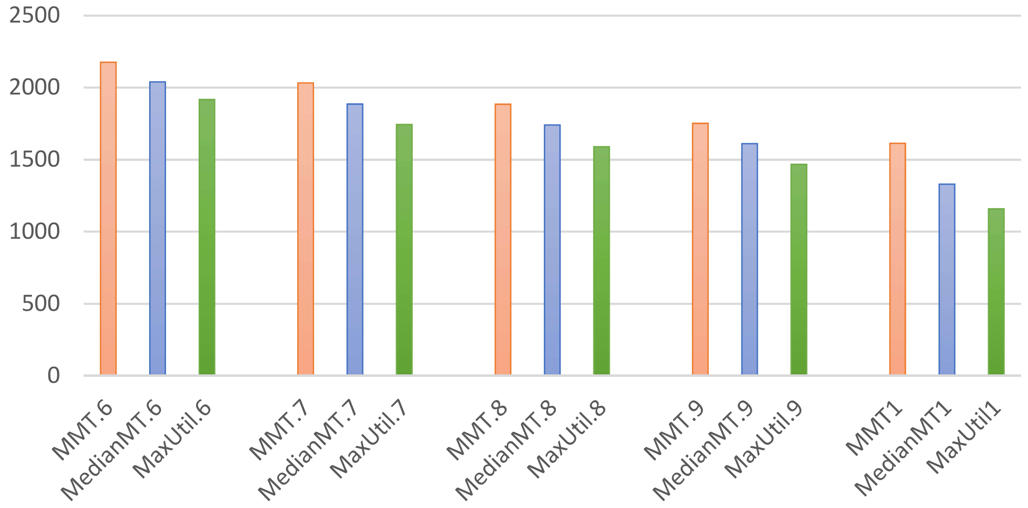


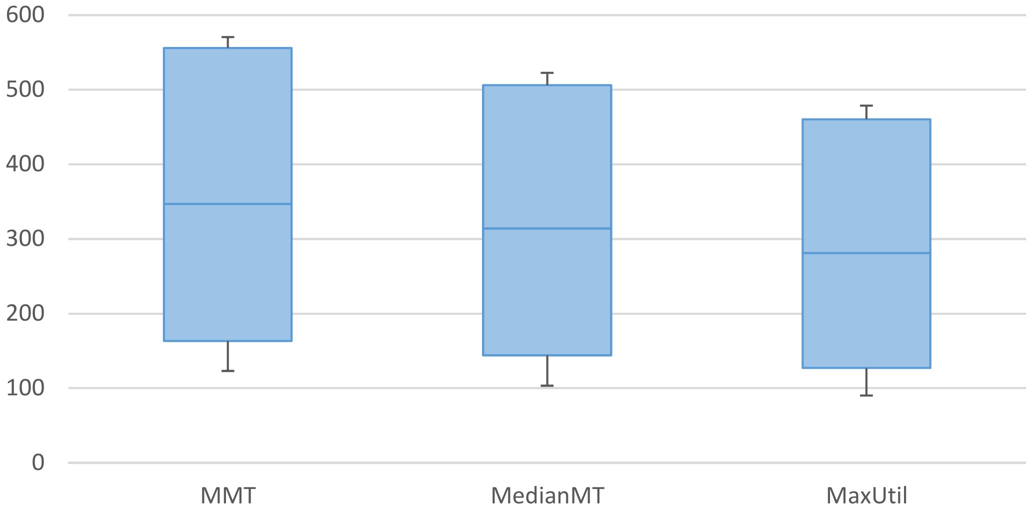
Figure 4.2: Mean execution time (seconds) for VM mapping policy Thr (in the absence of VM migration) and when VM migration policies MMT, MedianMT and MaxUtil are deployed

also outperforming MedianMT.

Mean execution time for VM migration policies MMT [9], MedianMT and MaxUtil are illustrated in Figure 4.4 for each VM mapping threshold value ranging from 0.6 to 1 and their box plots.

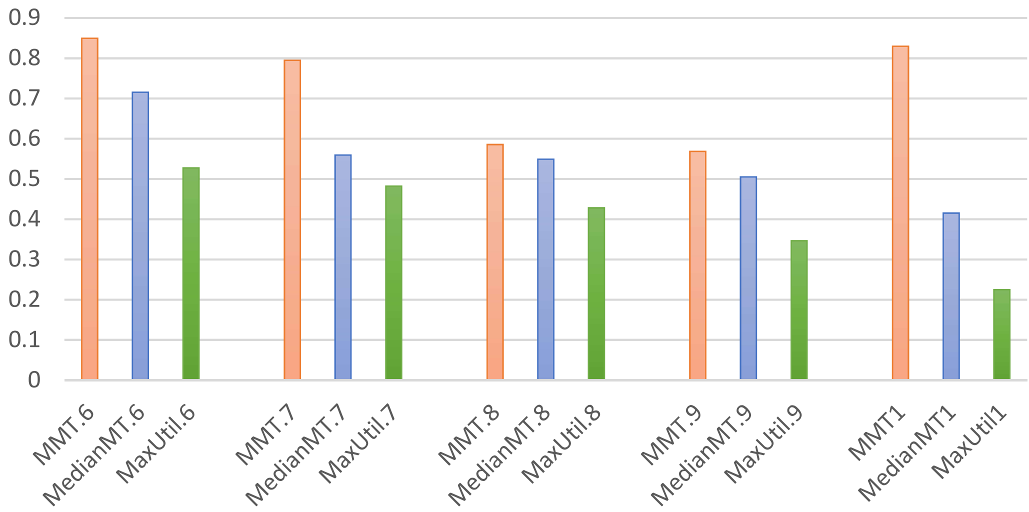


(a) Energy consumption (kW) by MMT, MedianMT and MaxUtil when VM mapping threshold value ranges from 0.6 to 1 (portion of hosts capacity).

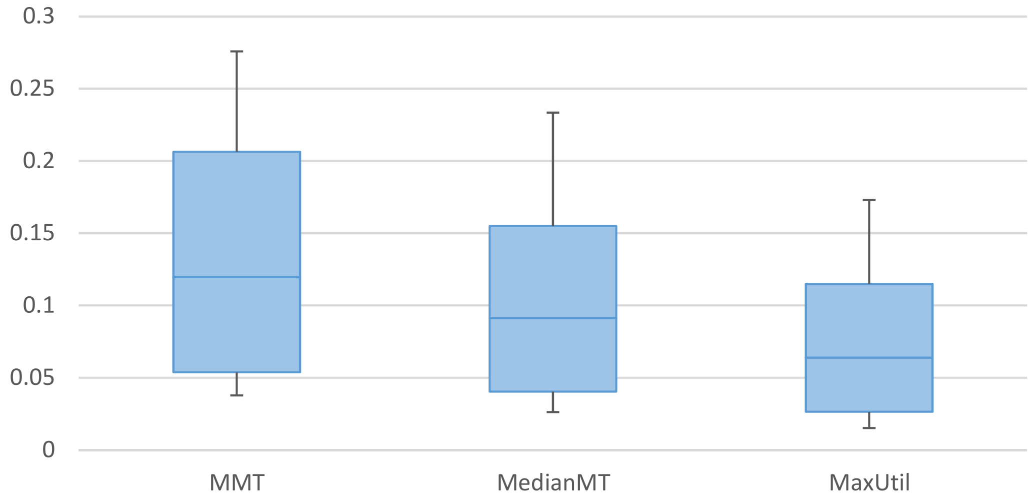


(b) Box plots of total energy consumption (kW) values by MMT, MedianMT and MaxUtil.

Figure 4.3: Energy consumption by Thr, MedianMT, MaxUtil and MMT



(a) Mean execution time (seconds) by MMT, MedianMT and MaxUtil when VM mapping threshold value ranges from 0.6 to 1 (portion of hosts capacity).



(b) Box plots of mean execution time (seconds) values by MMT, MedianMT and MaxUtil.

Figure 4.4: Mean execution time by Thr, MedianMT, MaxUtil and MMT

Tables 4.1 and 4.2 present the statistical descriptions of total energy consumption and mean execution time of VM migration policies, respectively.

According to Tables 4.1 and 4.2, MaxUtil has the least total energy consump-

	Mean	St. Deviation	Minimum	Maximum
MMT	189.1982	37.99089	123.15	301.60
MedianMT	172.1864	38.39878	103.36	284.47
MaxUtil	157.5686	38.59788	90.28	269.25

Table 4.1: Statistical description of total energy consumption of VM migration policies: MMT, MedianMT and MaxUtil

	Mean	St. Deviation	Minimum	Maximum
MMT	.07256140	.027134476	.037800	.156340
MedianMT	.05489600	.020151246	.026240	.142200
MaxUtil	.04020220	.017879943	.015220	.109140

Table 4.2: Statistical description of mean execution time of VM migration policies: MMT, MedianMT and MaxUtil

tion, on average, and the shortest mean execution time, on average, compared to MedianMT and MMT [9]. On average, MedianMT represented less total energy consumption and shorter mean execution time than MMT [9]. However, MedianMT's total energy consumption and mean execution time values were higher than MaxUtil's.

To demonstrate the magnitude and significance of the differences between VM migration policies should the null hypothesis be tested. Null hypothesis tests whether the results of the VM migration policies are statistically different. To verify this, the non-parametric Wilcoxon Signed Ranks Test is done. This test essentially looks at pairs of data values from paired groups and counts the numbers each group of values ranks higher/lower than the value of the other group.

Table 4.3 represents Wilcoxon Signed Ranks test for paired comparison of total energy consumption (kW) and mean execution time (seconds) of competing

VM migration policies.

	Asympt. Sig. (2-tailed), total energy consumption	Asympt. Sig. (2-tailed), total execution time
MedianMT - MMT	.000 (< .05)	.000 (< .05)
MaxUtil - MMT	.000 (< .05)	.000 (< .05)
MaxUtil - MedianMT	.000 (< .05)	.000 (< .05)

Table 4.3: Results of the Wilcoxon Signed Ranks test for energy consumption (kW) and mean execution time (seconds)

Table 4.3 represents the statistical significance of the difference between the paired VM migration policies in its "Asympt. Sig." columns for total energy consumption and mean execution time. A significance value of less than 0.05 means the difference between the paired values is significant. That is, statistically, MedianMT has significantly less energy consumption than MMT [9] and MaxUtil has statistically significantly less energy consumption than MMT [9] and MedianMT.

According to Table 4.3 paired comparisons of mean execution time by competing VM migration policies have are statistically different when the asymptotic significance (Asympt. Sig.) is less than 0.05 which represents their statistical difference. MedianMT proved to have statistically significant shorter mean execution time compared to MMT [9] while MaxUtil demonstrated statistically significant shorter mean execution time than MMT [9] and MedianMT.

In order to further investigate the reasons for energy reduction and shortening of mean execution time, the number of host shutdowns and VM migrations relevant results are provided in Figures 4.5 and 4.6.

According to Figures 4.5 and 4.6 MaxUtil has the fewest VM migrations and host shutdowns. That is, MaxUtil resolves the imbalance problem by performing few VM migrations and causing few host switches. It explains the MaxUtil's

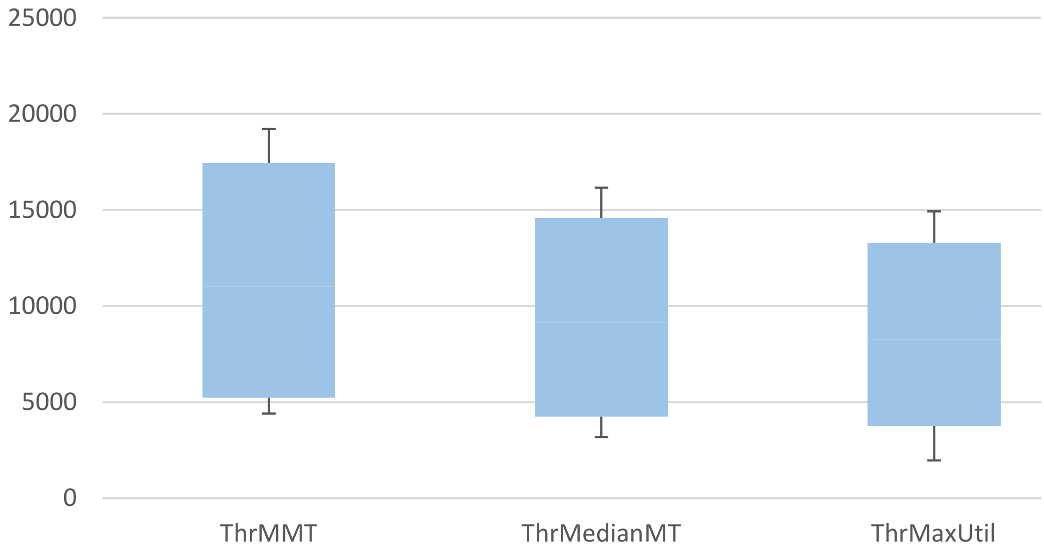


Figure 4.5: Number of hosts shutdowns when MMT, MedianMT and MaxUtil VM migration policies are deployed

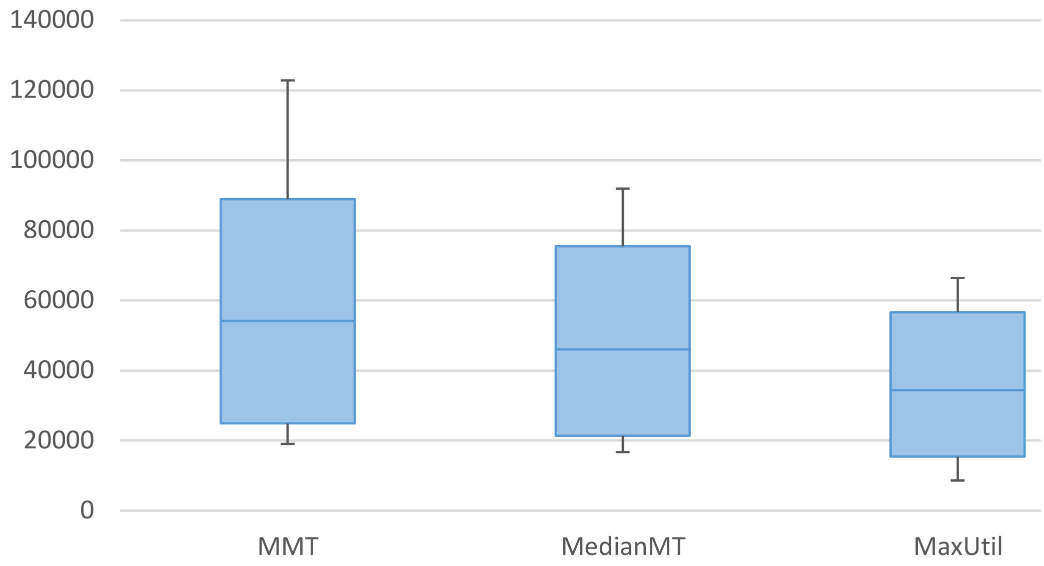


Figure 4.6: Number of VM migrations when MMT, MedianMT and MaxUtil VM migration policies are deployed

energy and time dominance given that each VM migration and host switch has time and energy drawbacks.

4.4 Discussion

VM migration proved to be an effective approach to reduce total energy consumption and to shorten mean execution time when it is deployed after VM mapping, compared to a VM mapping policy without performing VM migrations.

On average, the given VM mapping policy resulted in 5.88, 6.36 and 7 times more energy consumption than when it was enabled to perform VM migration by MMT [9], MedianMT and MaxUtil in terms of total energy consumption, respectively. In terms of mean execution time, on average, the VM mapping policy (in the absence of VM migration) resulted in 72, 93 and 126 times longer mean execution time compared with when it performed VM migrations by MMT [9], MedianMT and MaxUtil policies, respectively.

It demonstrates the impact of VM migration on reducing total energy consumption and shortening mean execution time on a given VM mapping heuristic, despite VM migration's energy and time drawbacks. Albeit, the VM migration policy performing fewer VM migrations and having fewer host switches can reduce the energy and time overhead and outperform other policies.

The comparison between the VM migration policies demonstrated that MaxUtil which selects VMs with the highest utilization level outperformed both MMT [9] and MedianMT that select VMs based on their migration time. This can be attributed to the fact that migrating VMs with high utilization levels is faster in solving the imbalance problem therefore performing fewer VM migrations. Moreover, frequent host switches negatively contribute to total energy consumption and execution time. Given that each VM migration and host switch entail energy and time drawbacks, fewer migrations and host switches can be the influencing factors in the reduction of energy consumption and the shortening of mean execution time by MaxUtil.

As a result, on average, MMT has more than 19% higher energy consumption and more than 75% longer mean execution time than MaxUtil. On average, MedianMT also has 10% and 35% more energy consumption and mean execution time than MaxUtil, respectively.

4.5 Summary

In this chapter, two VM migration policies are introduced, namely: MaxUtil and MedianMT. They are evaluated when only the VM mapping is carried out (without VM migration) and proved to lead to a substantial reduction in energy consumption and shortening mean execution time. Our proposed VM migration policies were also evaluated against a state-of-the-art VM migration policy. Both policies statistically outperformed the state-of-the-art policy significantly in terms of both total energy consumption and mean execution time.

The proposed VM migration policies optimized the VM migration process, so that the imbalance problem is handled while energy consumption is reduced and mean execution time is shortened. The proposed VM migration policies used micro level observation records, i.e. VM memory size or utilization level, as a deciding factor.

This chapter explained two VM migration policies to solve the imbalance problem in private Clouds while energy consumption and execution time are improved.

Chapter 5

ADAPTIVE ENERGY-EFFICIENT MAPPING AND MIGRATION MECHANISM FOR PRIVATE CLOUDS

VIRTUAL machine migration policies proposed in the previous chapter demonstrated the significant positive effects of VM migrations in reducing energy consumption and shortening mean execution time. In the previous chapter, we evaluated our proposed VM migration policies and proved that they outperform a state-of-the-art VM migration policy. Our proposed policies used micro level observation records, VM memory size or utilization level, as a contributing factor to seek energy efficiency.

An energy-efficient resource hyper-visor, however, can benefit from adaptivly responding to the changes in the system and resource requests based on micro level observation records. This adaptivity can be deployed when VM migration is being carried out. It enables the hyper-visor to respond to the potential changes in the system state and workload properties. In this chapter we answer two research questions. Research question 4) *How can the micro level observation records be used to enhance the VM migration process to adaptively select a VM with the highest likelihood of reducing energy consumption in private Clouds?* is answered by developing an adaptive VM migration mechanism. Our proposed mechanism includes a Bayesian inference based on micro level observation records to dynamically calculate the likelihood of a VM migration decision reducing energy consumption. The proposed migration mechanism is presented in a paper [89].

Resource hyper-visioning, however, includes both VM mapping and VM migration. In Chapter 3 the VM mapping is made adaptive using macro level observation records of total energy consumption by available VM mapping policies. In Chapter 4 the impact of VM migration on reducing total energy consumption and shortening mean execution time is presented. In order to have an adaptive energy-efficient hyper-visioning mechanism and answer research question 5) *How can the micro level observation records about the state of hosts' resources and VMs' current requests can be used to adaptively map and migrate VMs, to reduce energy consumption?*, VM mapping and VM migration should be modelled in a way that they reflect on the results of each other.

This chapter details the invention of a novel adaptive VM migration mechanism and an adaptive energy-efficient resource hyper-visioning mechanism, inclusive of both VM mapping and VM migration, based on micro level observation records. The proposed adaptive mechanisms establish a Bayesian based inference to seek energy reduction while not sacrificing the execution time. The Bayesian inference is based on micro level observation records, e.g. utilization levels, and adaptively makes VM migration and resource hyper-visioning decisions with the objective of reducing energy consumption while shortening execution time. The proposed mechanisms are evaluated against state-of-the-art heuristics. The proposed adaptive VM migration mechanism is published as a paper [89]. The results of the proposed adaptive resource hyper-visor, "Energy-efficient Adaptive Virtual Machine Migration Mechanism for Private Clouds", is published in the journal of Concurrency and Computation: Practice and Experience (CCPE) [90].

This chapter is organized as follows. Section 5.1 details our proposed adaptive migration mechanism. Then Section 5.2 describes our energy-efficient hyper-visioning mechanism inclusive of both VM mapping and VM migration. The evaluation of our proposed mechanisms against state-of-the-art heuristics is reported in Section 5.3. It is then followed by a discussion in Section 5.4 and a summary of this chapter in Section 5.5.

5.1 Adaptive VM Migration Mechanism

In order to respond to the changes in the system adaptively, a learning mechanism is needed to make the VM migration process adaptive. We propose a Bayesian inference that learns which VMs should be migrated according to the current VMs' utilization and the observed micro level records of similar migrations.

We first propose a Bayesian based heuristic that distinguishes between VMs based on their utilization (processing) level. The conditional probabilities of migrating a VM with a certain utilization level, a micro level observation, are maintained based on feedback drawn from the difference in hosts' utilization level that were involved in the migration (source and destination hosts).

The details of the proposed adaptive VM migration mechanism is explained in the following sections.

5.1.1 Notations

For every host in the system, $h_i \in H$, h_i, and VMs running on them, set of VM as VM_i running on h_i, $VM_i = \{vm_1, vm_2, \cdots, vm_m\}$. Each $vm_q \in VM_i$ has processing share (in terms of Million Instruction Per Second: MIPS) as *PS*. VM migration is $Mig = \{(vm_q, h_j) | vm_q \in VM_i, h_j \in H, i \neq j\}$, where Mig is a set of possible combinations of vm_q taken from h_i to host h_j. Mig is a finite set because VM_i (set of VMs on host h_i), VM_j (set of VMs on host h_j) and H are finite sets.

5.1.2 Bayesian Network and the Components of the Proposed VM Migration Mechanism

We propose a Bayesian based VM migration mechanism, termed as Bayesian Migration Heuristic (BMH). BMH has a Bayesian based inference and a Bayesian network to represent its inference. BMH estimates the impact of migrating a VM of a certain type (based on VM's utilization level) on the source and destination hosts' utilization level. The proposed Bayesian inference and its Bayesian network aim at reducing energy consumption and shortening execution time via arbitration of hosts' utilization levels.

The proposed Bayesian inference is illustrated by its network in Figure 5.1. The BN's hypothesis hold because there is only one contributing factor that guarantees no dependencies between factors. In Figure 5.3 the nodes labeled as R_1, R_2, R_3 and R_4 represent the possible VM utilization ranges. In the proposed Bayesian network, the ranges are defined based on the quartiles of the observed utilization values, $R = \{R_1, R_2, R_3, R_4\}$, where R_1 is a range from minimum to first quartile, R_2 is from first quartile to second quartile, R_3 is from second quartile to third quartile and R_4 is third quartile to maximum.

The probabilities related to each node - $p(R_1)$, $p(R_2)$, $p(R_3)$, $p(R_4)$ - are the likelihoods that a VM from the corresponding utilization range, R_i, is migrated.

Directed arcs connecting the nodes demonstrate dependencies between them. The strength of the dependency is quantified by conditional probabilities.

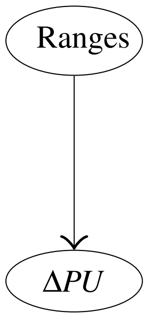
The probability of A conditional to B can be calculated as:

$$p(A|B) = \frac{p(B \cap A)}{p(B)} \tag{5.1}$$

where $p(B)$ is the prior probability for B, modelled as $p(R_i)$, and $p(B \cap A)$ is the joint probability of the two variables, $p(\Delta^+ \cap R_i)$.

We measure the impact of a VM migration decision on energy consumption by the changes in the computational resources provided to VMs. Changes in the computational resources is modeled as the node labeled ΔPU in Figure 5.1. We termed the hosts in which the VM is being migrated from and to as the source and destination hosts respectively. If, supposedly, the source host, h_i, is already running n VMs and the destination host, h_j, is running m VMs, the total processing

Utilization ranges			
R_1 :	R_2 :	R_3 :	R_4 :
$min < R \leq 1stQ$	$1stQ < R \leq 2ndQ$	$2ndQ < R \leq 3rdQ$	$3rdQ < R \leq max$
p(R_1)	p(R_2)	p(R_3)	p(R_4)



	Differences in the utilization level	
	$\Delta = (HU_i + HU_j)_{postMig} - (HU_i + HU_j)_{preMig}$	
Ranges	$\Delta^+ : \Delta \geq 0$	$\Delta^- : \Delta < 0$
R_1	p($\Delta^+ \| R_1$)	p($\Delta^- \| R_1$)
R_2	p($\Delta^+ \| R_2$)	p($\Delta^- \| R_2$)
R_3	p($\Delta^+ \| R_3$)	p($\Delta^- \| R_3$)
R_4	p($\Delta^+ \| R_4$)	p($\Delta^- \| R_4$)

Figure 5.1: Proposed Bayesian network for VM migration

units (computational resources) provided to the VMs on source and destination hosts is equal to equation 5.2.

$$PU_{Pre} = \sum_{i=1}^{n} PS_i + \sum_{j=1}^{m} PS_j \tag{5.2}$$

where PU_{Pre} is the processing units provided to the VMs on source and destination hosts before migration.

After the VM migration is carried out, the total processing units (computational resources) provided for VMs on h_i and h_j is equal to equation 5.3 when a VM is migrated from h_i to h_j.

$$PU_{Post} = \sum_{i=1}^{n-1} PS_i + \sum_{j=1}^{m+1} PS_j \tag{5.3}$$

The difference between the processing units (computational resources) provided for VMs before and after migration is

$$\Delta PU = PU_{Post} - PU_{Pre} \tag{5.4}$$

which is measured in MIPS (Million Instructions Per Second). ΔPU values greater than or equal to zero, indicate that the VM migration has solved the imbalance problem and provided VMs with more or the same processing units. If Δ represents the outcome for Bayesian, ΔPU greater than zero is termed Δ^+. The conditional probability $p(\Delta^+|R_i)$ estimates the probability of Δ^+ given a VM with utilization level in the R_i range is migrated. Migrating a VM from a certain utilization range affects the frequency of the processor and VMs' execution time that contribute to total energy consumption.

5.1.3 Calculating Bayesian Probabilities

To calculate the Bayesian probabilities for our proposed Bayesian network, prior probabilities of $P(\Delta^+)$ and $P(R_i)$ should be calculated. $P(\Delta^+)$ is the number of instances of Δ^+ per total number of migration instances. $P(R_i)$ is the number of migration instances where the migrated VM was from R_i divided by the total number of instances.

$P(R_i|\Delta^+)$ is the likelihood of observing Δ^+ if the selected VM was from range R_i. $P(R_i|\Delta^+)$ is calculated according to algorithm 5.1.

Algorithm 5.1 Calculating likelihoods, $P(R_i|\Delta^+)$

1: **for each** instance of VM migration **do**
2: **if** VM belongs to R_i **then** Increment $counter_i$
3: **if** this instance has a positive Δ value **then** increase $counter_{i,\Delta^+}$
4: $P(R_i|\Delta^+) = \dfrac{counter_{i,\Delta^+}}{counter_i}$
5:

After calculating Bayesian probabilities, they are used for calculating the likelihood of observing Δ^+ for each VM utilization range, R_1 to R_4 as the middle column in the Δ table in Figure 5.1 based on Bayesian theorem 5.1. These likelihoods (posterior probabilities) are then used to select VMs for migration. The following section describes the flow of the VM selection for the migration process based on the calculated Bayesian probabilities.

5.1.4 Flowchart of VM Migration Mechanism

The proposed VM selection for migration, Bayesian Migration Heuristic (BMH), follows a flowchart as Figure 5.2.

Our VM selection for migration heuristic, BMH, starts by finding a host with an imbalance problem. Note that this flowchart is a loop where it continually looks for imbalance in the system.

BMH selects a VM for migration based on the Bayesian probabilities calculated, i.e. posterior probabilities. Based on the Bayesian probabilities, a VM of a

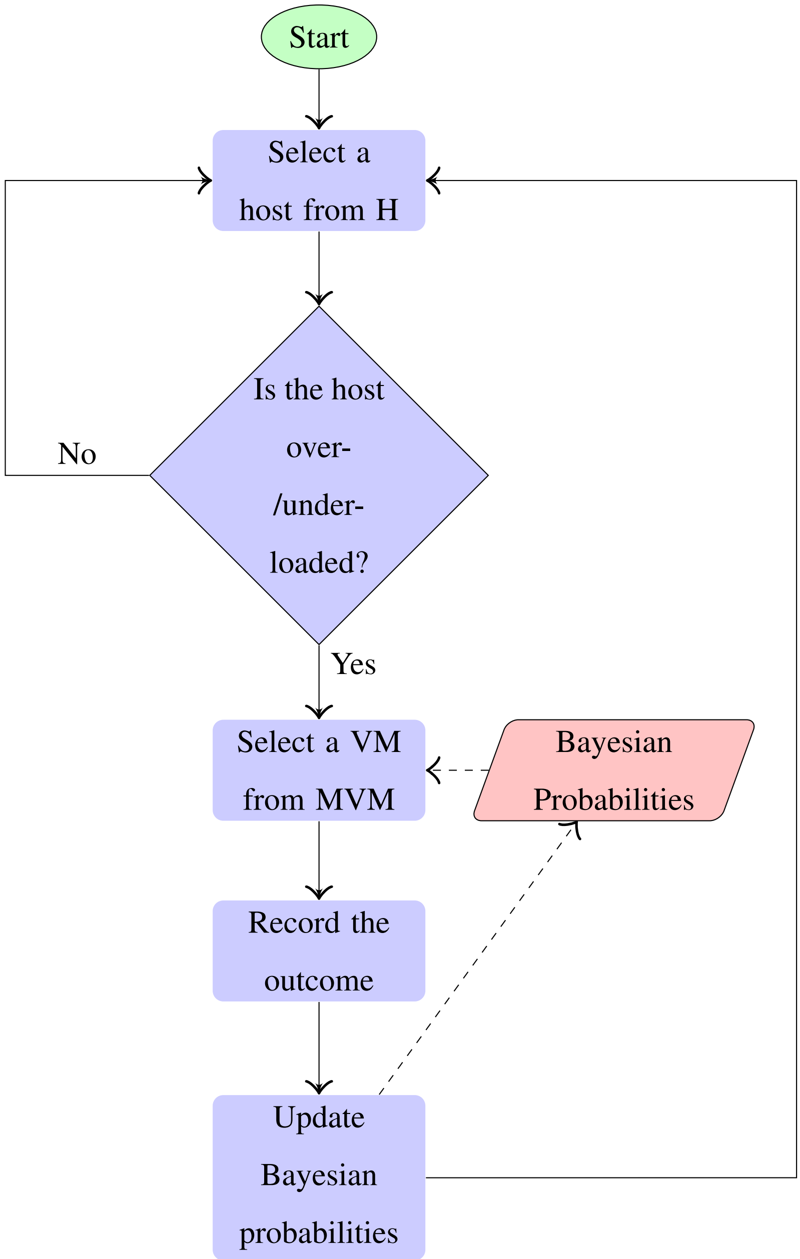


Figure 5.2: Bayesian based VM selection for migration, BMH, flowchart

utilization range with the highest probability of improving the utilization level on source and destination hosts is selected for migration. The outcome of this migration is then recorded in terms of VM's utilization level (before migration) and the difference in the hosts' (source and destination hosts) utilization level. These are then used to update Bayesian probabilities.

5.2 Adaptive Hyper-visioning Mechanism

In this section, our Bayesian based hyper-visioning mechanism, inclusive of both VM mapping and VM migration, is explained. Our Energy-efficient Adaptive Migration (EAM) mechanism extends our BMH, as detailed in Section 5.1, by incorporating an adaptive VM mapping mechanism using micro level observation records. This incorporation is not a joint effort of previously reported works but rather the integration of VM mapping and VM migration mechanisms in one adaptive model. It means that one adaptive process is developed that serves both VM mapping and VM migration. The reasons for the integration are the minimization of information log volume and deduction of computational complexity.

If VM mapping and VM migration processes were set to be separated, two information logs had to be kept, one for the learning phase of the mapping stage and another for the learning phase of the VM migration stage. That means two batches of computations have to be carried out, one for VM mapping and the other for VM migration. By incorporating both in a single mechanism the information log volume and computational complexity can be reduced.

Such incorporation is also expected to result in further optimization of energy consumption as the VM mapping and VM migration sections are representative of each other's decisions and reflect on each other's outcome as an overall hypervisor.

Our (EAM) mechanism adaptively maps and migrates VMs to minimize energy consumption. The Bayesian Network and Bayesian Inference of EAM mechanism are detailed when they address both VM mapping and VM migration. EAM mechanism's flowchart and algorithm are then explained.

5.2.1 Proposed Hyper-visioning Bayesian Network

We use Bayesian inference to make adaptive energy-efficient hyper-visioning decisions. To do so, we illustrate the relationship between the contributing factors in our Bayesian inference in a Bayesian network. The proposed Bayesian network presents the conditional dependencies between the factors and quantifies the likelihood of the contributing factors affecting the output measure. We term our Energy-efficient Adaptive hyper-visioning Mechanism as EAM. The Bayesian network of EAM is presented in Figure 5.3 to explain the relation between the contributing factors (host utilization and type of VM) and the output measure (changes on the hosts' utilizations).

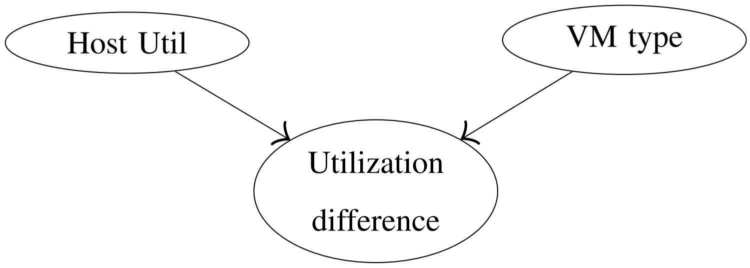
Directed arcs between nodes in Figure 5.3 represent dependencies between the contributing factors and the output measure. Conditional probabilities, shown in the table on the bottom, is a quantification of the level of dependencies associated with each combination of the contributing factors' values.

The notion behind choosing VM type and current host utilization as contributing factors lies in the facts that, one, VM type determines the maximum resource requests and, two, current host utilization indicates the remaining resource capacity on the host. The difference in utilization in the system after a resource hyper-visioning decision is made, is chosen as the output measure because of the effect it has on energy consumption through the changes it entails on the frequency of the processors in hosts.

The BN's hypothesis of not having conditional dependencies between the con-

host utilization level (L)	
$l_1 : HU \leq 50\%$	$l_2 : HU > 50\%$
p(l_1)	p(l_2)

VM Type			
t_1	t_2	t_3	t_4
p(t_1)	p(t_2)	p(t_3)	p(t_4)



		$\Delta = (HU_i + HU_j)_{postMig} - (HU_i + HU_j)_{preMig}$	
L	VMType	$\Delta^+ : \Delta \geq 0$	$\Delta^- : \Delta < 0$
l_1	t1	p($\Delta^+ \mid l_1, t_1$)	p($\Delta^- \mid l_1, t_1$)
l_1	t2	p($\Delta^+ \mid l_1, t_2$)	p($\Delta^- \mid l_1, t_2$)
l_1	t3	p($\Delta^+ \mid l_1, t_3$)	p($\Delta^- \mid l_1, t_3$)
l_1	t4	p($\Delta^+ \mid l_1, t_4$)	p($\Delta^- \mid l_1, t_4$)
l_2	t1	p($\Delta^+ \mid l_2, t_1$)	p($\Delta^- \mid l_2, t_1$)
l_2	t2	p($\Delta^+ \mid l_2, t_2$)	p($\Delta^- \mid l_2, t_2$)
l_2	t3	p($\Delta^+ \mid l_2, t_3$)	p($\Delta^- \mid l_2, t_3$)
l_2	t4	p($\Delta^+ \mid l_2, t_4$)	p($\Delta^- \mid l_2, t_4$)

Figure 5.3: Bayesian network for VM mapping and VM migration when a VM is moved between h_i and h_j

tributing factors should be checked before proceeding with the proposed method. Given that host utilization and the type of the VMs are independent from each other, there exists no dependencies between them. Therefore, BN's hypothesis holds in this context.

Four VM types and two host utilization levels are considered as the values each contributing factor can be assigned to. The VM types and utilization levels can be extended to include other existing VM types. However, the deciding threshold of 50% for host utilization is driven from the idea of differentiating hosts where at least half of their resources are in use compared to those that serve their VMs with less than half their capacity. Bayesian probabilities related to $p(\Delta^+|l_i,t_j)$ is the likelihood of selecting a VM from type t_j from a host with utilization level of l_i either improves the utilization level on source and destination hosts or keeps it the same.

When a VM from a certain type is selected to be migrated or a specific host is chosen for a VM to be mapped on the hosts contribute to the changes in total utilization. Any change on utilization level in the system should be recorded after each hyper-visioning decision is made, either VM mapping or VM migration. What is of interest is to seek decisions that are more likely to lead to an increase in total utilization or at least no change.

5.2.2 Calculating Bayesian Probabilities

The proposed mechanism works with probabilities for Δ^+, an increase/ no change in utilization of hosts after a hyper-visioning decision is completed. These probabilities are in the column under Δ^+ in Figure 5.3 at the bottom of the table under the "Utilization difference" node.

According to equation 5.1 the probability of a increase/unchanged level of utilization conditional to selecting a VM from type t_j and a host with utilization level in l_i is:

$$p(\Delta^+|l_i,t_j) = \frac{p(l_i,t_j|\Delta^+).p(\Delta^+)}{p(l_i,t_j)} \tag{5.5}$$

To calculate Bayesian probabilities, $p(\Delta^+|l_i,t_j)$ for every i and j combination conditional probabilities of $p(l_i,t_j|\Delta^+)$ and prior probabilities of $p(\Delta^+)$ and $p(l_i,t_j)$, should be calculated. $p(\Delta^+)$ can be calculated by dividing the number of hyper-visioning decision instances that have positive Δ value by the total number of instances. Prior probabilities of $p(l_i,t_j)$ are calculated by dividing the number of hyper-visioning decision instances where HU is of level (l_i) and VM is from type t_j, by total number of instances.

$p(l_i,t_j|\Delta^+)$ is the likelihood of observing Δ^+ conditional to VM type t_j and host of utilization level l_i being involved in the hyper-visioning decision. These probabilities quantify the likelihood of observing Δ^+ when l_i and t_j are involved in the hyper-visioning process. That is, if a VM from type t_j is migrated from or mapped to a host with utilization level of l_i, how likely it is that this combination results in a positive Δ value? The probabilities of $p(l_i,t_j|\Delta^+)$, likelihoods, should be calculated for every t_j and l_i combination according to algorithm 5.2. Likelihoods are then used to calculate Bayesian probabilities (posterior probabilities).

Algorithm 5.2 Calculating likelihoods, $p(l_i,t_j|\Delta^+)$

1: **for each**mapping/migration instance **do**
2: **if** HU is l_i and VM belongs to t_j **then** increase $counter_ij$
3: **if** this instance has Δ^+ **then** increase $counter_{ij,\Delta^+}$;

$$p(l_i,t_j|\Delta^+) = \frac{counter_{ij,\Delta^+}}{counter_{ij}}$$

Posterior probabilities, $p(\Delta^+|l_i,t_j)$, are then calculated according to equation 5.5. Posterior probabilities provide a basis for selecting VMs for migration and all the mapping/re-mapping of VMs to hosts. For instance, when a host of utilization level l_p encounters imbalance, a VM of type t_j is selected that has the highest $p(\Delta^+|l_p,t_j)$ probability. It is then mapped to a host of level l_q if the highest posterior probability for Δ^+ combined with VM type, t_j belongs to l_q, that is, $p(\Delta^+|l_q,t_j)$ is the highest Bayesian probability for t_j combination.

5.2.3 Flowchart of Hyper-visioning Mechanism

The flowchart of our proposed Bayesian based migration mechanism, Energy-efficient Adaptive Migration (EAM) mechanism is presented in Figure 5.4 to illustrate the steps.

Figure 5.4 has six processes, two conditional point and a data node. Solid lines represent control flows and dashed lines are data flows. EAM hyper-visioning process starts when the system first commences and it continuously looks to make the required hyper-visioning decisions. It waits for new VMs to arrive for mapping. It also goes into a loop that constantly looks for an imbalanced host in the system. The loop is to guarantee that any potential imbalance is detected. Because the act of hyper-visioning resources is continuous the flowchart does not have a "Stop" node.

The flowchart starts with scanning the system for potential new VMs and hosts that are imbalanced (either over-loaded or under-loaded). If there are new VMs to map, they will be mapped based on the Bayesian probabilities for the VM type that needs to be mapped and the utilization level of hosts. If the highest probability for a given VM type is a host with less than 50% utilization level, a host of the utilization level is selected. Note that the mechanism does not differentiate the hosts that have utilization levels of the same category, l_i. EAM also looks for imbalance in the system. If there is a host that is imbalanced, either over-loaded or under-loaded, a VM from the host should be selected for migration. To select a VM from an imbalanced host, Bayesian probabilities for the host should be

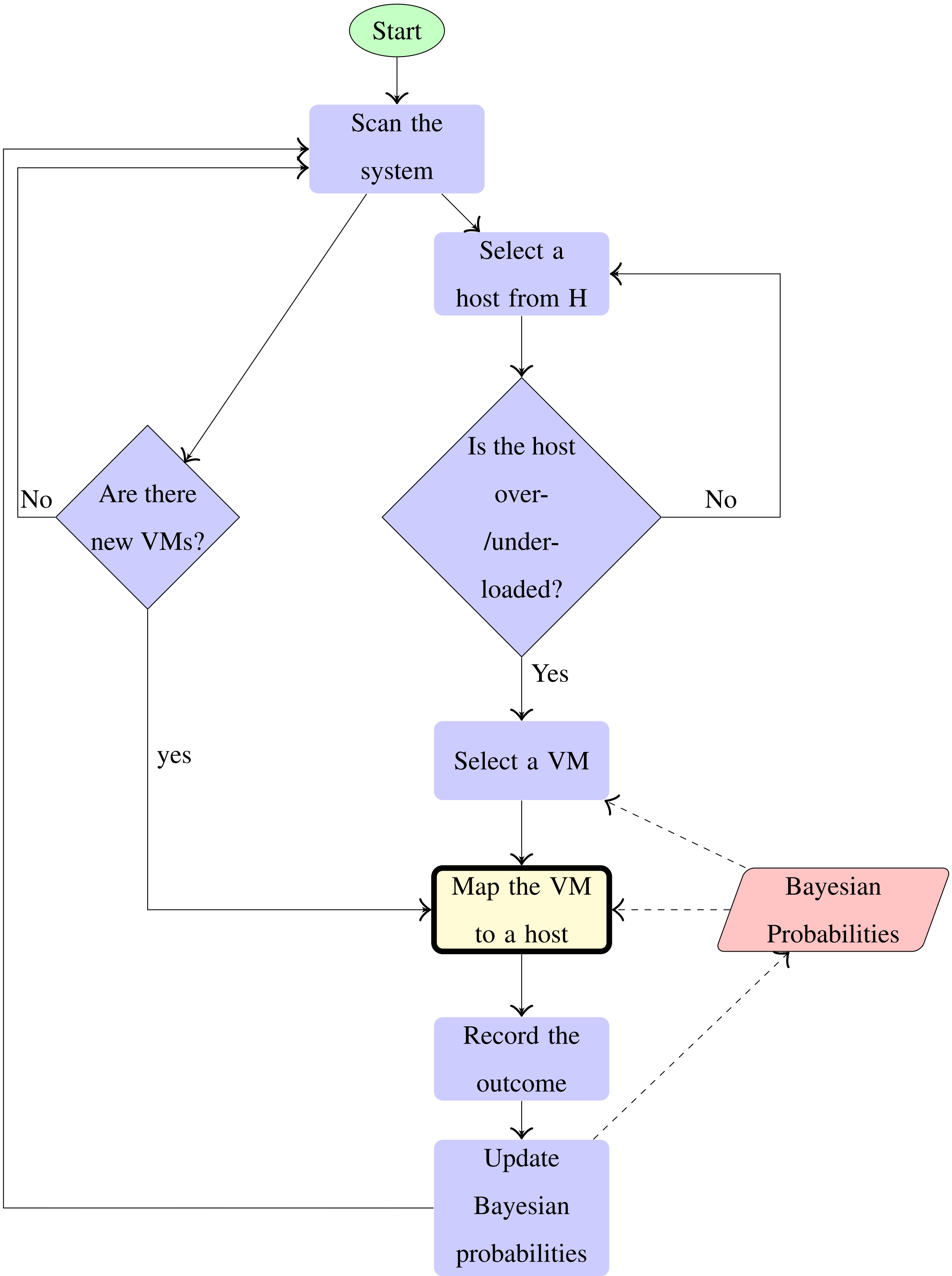


Figure 5.4: Bayesian-based VM mapping and VM migration flowchart

looked at, so that a VM with minimum probability of Δ^+ for the imbalanced host will be selected for migration. Then the selected VM will be mapped based on the Bayesian probabilities of this VM will lead to Δ^+ on a host with the highest probability, as is done for new VMs. The proposed Bayesian Inference differentiates hosts and VMs according to their level of utilization and type, respectively.

It is worth noting that the initial Bayesian probabilities are equal for all VM type and host utilization levels. The probabilities are then updated after a predetermined number of migration instances, n, is gathered. Bayesian probabilities are then reflective of the latest n instances. Therefore, the outcome of each hyper-visioning decision, either mapping new VMs or selecting VMs for migration and re-mapping them, should be recorded in terms of Δ value. It should then be used to update Bayesian probabilities. EAm then follows by scanning the system for required hyper-visioning decisions.

In addition to the difference in the Bayesian network of EAM mechanism and BMH, their flowcharts differ in the node highlighted in EAM's flowchart. The highlighted node in EAM's flowchart means that the selected VM will be mapped to a host of a certain utilization level that, based on Bayesian probabilities, is more likely to result in Δ^+. A host will be chosen that has the highest probability of Δ^+ for the type of the selected VM. So, for mapping a VM of type t_x, a host of utilization l_j is chosen if $p(\Delta^+|l_j,t_x)$ is the highest Bayesian probability among combinations that include t_x.

5.3 Evaluation

The proposed mechanism is evaluated on an energy-aware simulation package. The following sections detail the simulation settings and the results of the simulation.

5.3.1 Experimental Settings

Our Energy-efficient Adaptive Migration (EAM) mechanism is evaluated on the CloudSim simulation package [14, 13], an energy-aware simulation package. The evaluation workloads are included in CloudSim and are from CoMon project which is a monitoring infrastructure for PlanetLab [76]. Ten days (in April and March 2011) are selected by the Cloud Lab team [1] at University of Melbourne to be included in Cloudsim and are previously used as a benchmark in the literature [9, 87, 86, 89]. Using the same chosen dates for our evaluation enables the reproduction of the results.

Evaluation workloads describe resource requests from tasks that are being executed on real hardware, reported in intervals of five minutes in a 24 hour time frame. These workloads demonstrate the fluctuations in resource requests from VMs throughout their executions. It is an important characteristic for evaluating our adaptive mechanism as it is expected to make adaptive energy-efficient decisions especially when resource request fluctuations occur.

We consider a system with 800 hosts (heterogeneous: 400 HP ProLiant ML110 G4 and 400 HP ProLiant ML110 G5). The types and the number of hosts are set to the same numbers as [9] for fair comparison.

BMH and EAM start with equal Bayesian probabilities for every possible combination. They update their Bayesian probabilities when the first 300 instances (migration instances for BMH and hyper-visioning decisions, either mapping or migration, instances for EAM) are recorded. The simulation is run with 100, 300 and 600 migration instances for Bayesian probabilities to be calculated. The results for variations in the number of instances were consistent. Thus, the results of simulation for updating Bayesian probabilities with 300 latest migration instances are reported as a representative.

BMH determines the type of VMs based on their utilization level. For EAM, however, four VM types are considered as: 2.5 EC2 Compute Units, 0.85 GB; 2 EC2 Compute Units, 3.75 GB; 1 EC2 Compute Unit, 1.7 GB and; 0.5 EC2 Compute Unit, 0.633 GB.

The experiments are conducted on an Intel Core i7-6500 CPU machine with 2.50 GHz, 8GB of memory running Windows 10.

BMH and EAM are evaluated against a heuristic with VM mapping policy Thr, and the VM selection policy MMT [9] that is called MMT. Thr is based on threshold and it is set to the following values as in the original paper [9]: 0.6, 0.7, 0.8, 0.9 or 1. MMT, Minimum Migration Time (MMT) is reported as the best VM selection policy by Beloglazov and Buyya [9]. Our earlier VM selection policy, MaxUtil [87] with VM mapping policy Thr is reported and termed as MaxUtil.

Our proposed mechanisms are also compared with a Multi-Objective migration Heuristic (MOH) [83] with five objectives: load volume, energy consumption, thermal status, resource wastage and migration cost. In this heuristic a Static Bayesian Game - Multi-objective Genetic Algorithm is applied when objectives are in turn considered to be a player that optimizes their payoff function. The evaluation of their proposed heuristic is done when the scale of VM migrations is as big as 60000 GB of instances for the first generation of GA to be generated. However, in the actual workload of the selected benchmark the number of migratable VMs is smaller and GA section is not applicable.

EAM and BMH [89] are compared to represent the potential improvements, in terms of energy consumption and mean execution time, when adaptive energy-efficient VM mapping and VM selection for migration mechanisms are incorporated.

5.3.2 Constraints

In this set of simulations a set of constraints is applied as $C = \{c_{maxVMs}, c_{migratable}, c_{excludedHosts}\}$. Because the virtualization software used, i.e. Xen [7], the maximum number of VMs that can be mapped to a host is limited to 75 [83], $c_{maxVMs} = 75$. $c_{migratable}$ denotes that not all VMs are migratable. When migrating VMs, only migratable VMs will be chosen. $c_{excludedHosts}$ is to exclude hosts that cannot be a destination host. A host cannot be a destination host if it is already overloaded.

5.3.3 Results

The objective is to minimize energy consumption through the arbitration of hosts' utilization. Nevertheless, the execution time is also reported to guarantee that the energy efficiency has not imposed a longer execution time.

Figure 5.5 shows the results of MMT [9], MOH [83], MaxUtil [87], BMH and EAM in terms of energy consumption.

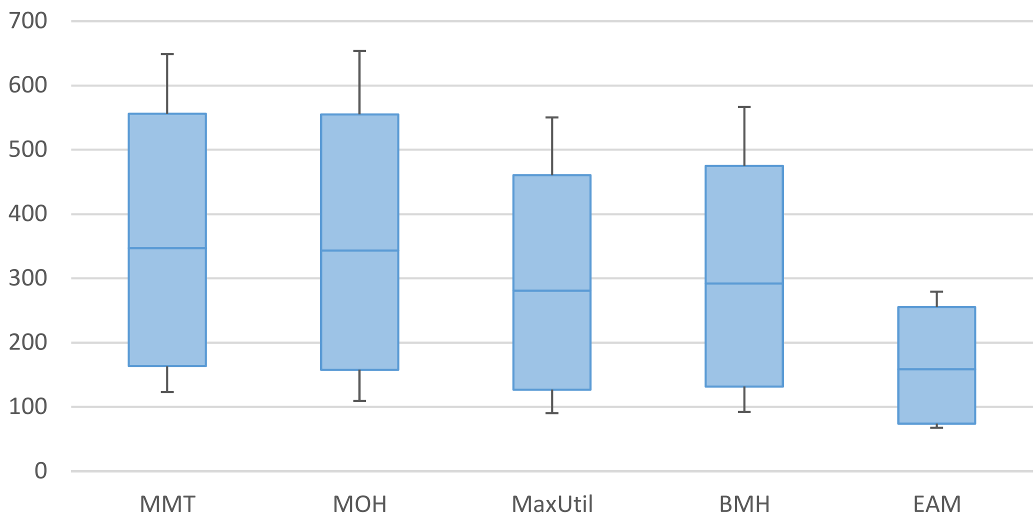


Figure 5.5: The box plots for total energy consumption (kilo Watts) of MMT, MOH, MaxUtil, BMH and EAM

According to Figure 5.5 EAM has the lowest energy consumption compared to other heuristics. To observe the performance of EAM in a greater depth, EAM is compared with a theoretically ideal case in Figure 5.6.

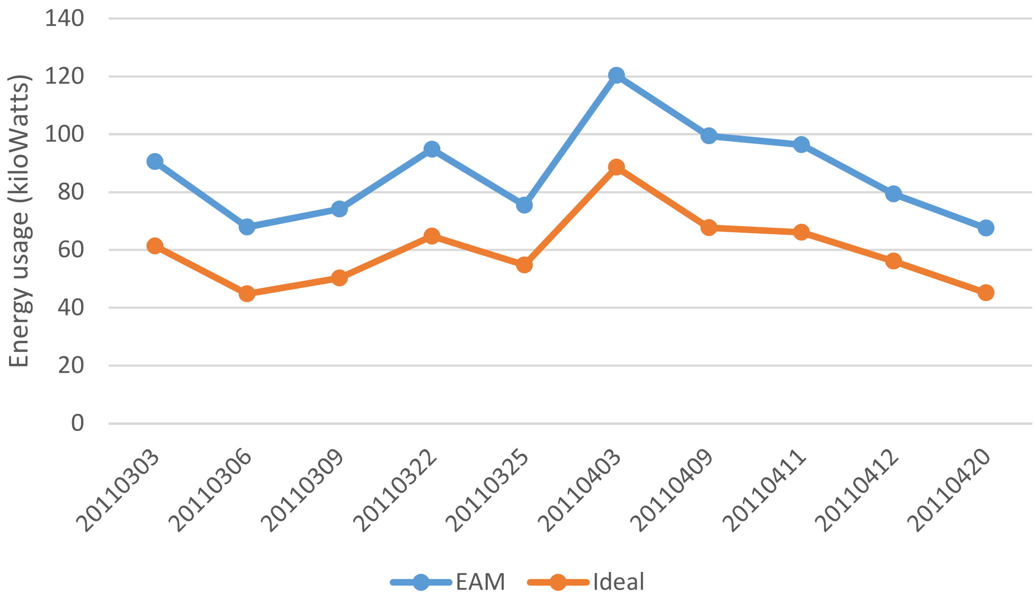


Figure 5.6: The comparison with ideal case in terms of total energy consumption (kilo Watts) for every workload set (labeled by date)

The ideal case represents a scenario where there are no resources wasted, all resources are optimally utilized, and no imbalance has occurred. Such an ideal scenario is unlikely to be achievable. However, the comparison shows how close our proposed mechanism is to the ideal, though not necessarily achievable, case.

The average result of the comparison suggests that energy consumption by EAM is relatively close to the ideal case by having only 44% more energy consumption on average. It is worth noting that the energy consumption by MMT [9] is 213% more than the ideal case.

To make sure that minimizing energy consumption has not imposed longer execution time, Figure 5.7 shows the mean time (in seconds) it took for each heuris-

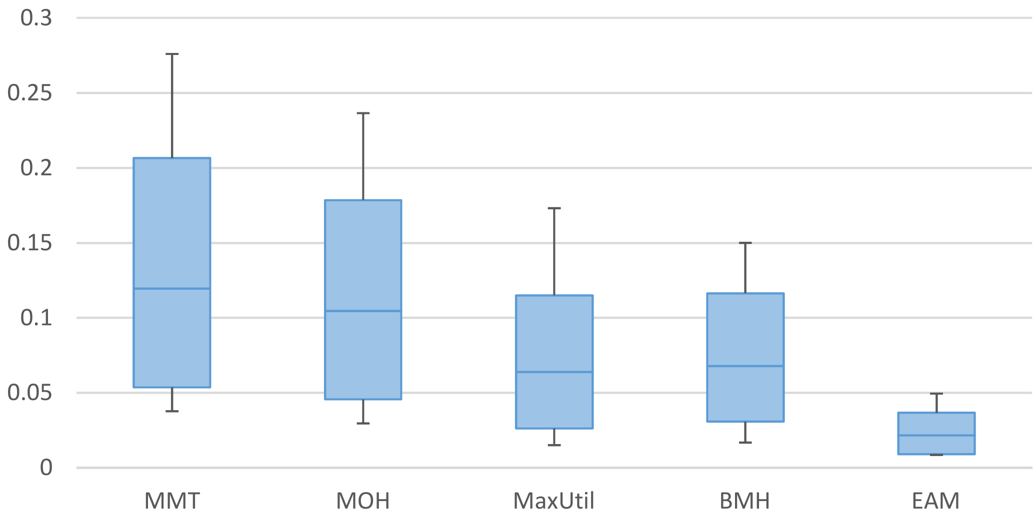


Figure 5.7: The box plots for mean execution time (in seconds) of MMT, MOH, MaxUtil, BMH and EAM

tic to execute the workloads. It is inclusive of the VM selection and VM mapping (heuristic's) time. It enables the comparison to represent any increase in the complexity of VM selection and VM mapping because of the required computations for calculating and updating Bayesian probabilities in our proposed heuristic.

Tables 5.1 and 5.2 represent statistical description of total energy consumption and mean execution time of the competing heuristics, respectively.

	Mean	St. Deviation	Minimum	Maximum
MMT	189.1982	37.99089	123.15	301.60
MOH	188.7884	42.74991	109.40	310.26
MaxUtil	157.5686	38.59788	90.28	269.25
BMH	162.2480	38.84498	92.23	274.57
EAM	86.5630	16.04903	67.53	120.31

Table 5.1: Statistical description of total energy consumption

	Mean	St. Deviation	Minimum	Maximum
MMT	.0725614	.02713448	.03780	.15634
MOH	.0620034	.02126852	.02965	.13176
MaxUtil	.0402022	.01787994	.01522	.10914
BMH	.0404848	.01386231	.01687	.08215
EAM	.0130658	.00418710	.00867	.02784

Table 5.2: Statistical description of mean execution time

To test the significance of difference between the competing heuristics, Table 5.3 presents the results of Wilcoxon Signed Ranks test and in the cases of energy consumption and execution time.

	Asympt. Sig. (2-tailed), total energy consumption	Asympt. Sig. (2-tailed), total execution time
EAM - MMT	.000 (< .05)	.000 (< .05)
EAM - MOH	.000 (< .05)	.000 (< .05)
EAM - MaxUtil	.000 (< .05)	.000 (< .05)
EAM - BMH	.000 (< .05)	.000 (< .05)
BMH - MMT	.000 (< .05)	.000 (< .05)
BMH - MOH	.000 (< .05)	.000 (< .05)
BMH - MaxUtil	.228 (> .05)	.357 (> .05)

Table 5.3: Results of the Wilcoxon Signed Ranks test for energy consumption (kW) and mean execution time (seconds)

Based on Table 5.3, almost all pairwise comparisons concluded in favour of EAM over MMT [9], MOH [83], MaxU [87] and BMH [89]. The only exception is the comparison between BMH and MaxUtil where they are not statistically different.

To investigate the underlying elements of our proposed mechanisms' dominance, the number of host shutdowns and VM migrations are reported in Figures 5.8 and 5.9. EAM performs the fewest number of VM migrations and host shutdowns compared to other heuristics. Given the energy and time drawbacks of VM migration and host shutdowns, it can be inferred that EAM's energy reduction and shortening of mean execution time are related to its fewer VM migration and host shutdowns. That is, efficient hyper-visioning decisions by EAM migrates VMs that will solve the imbalance problem relatively fast and causing fewer host shutdowns.

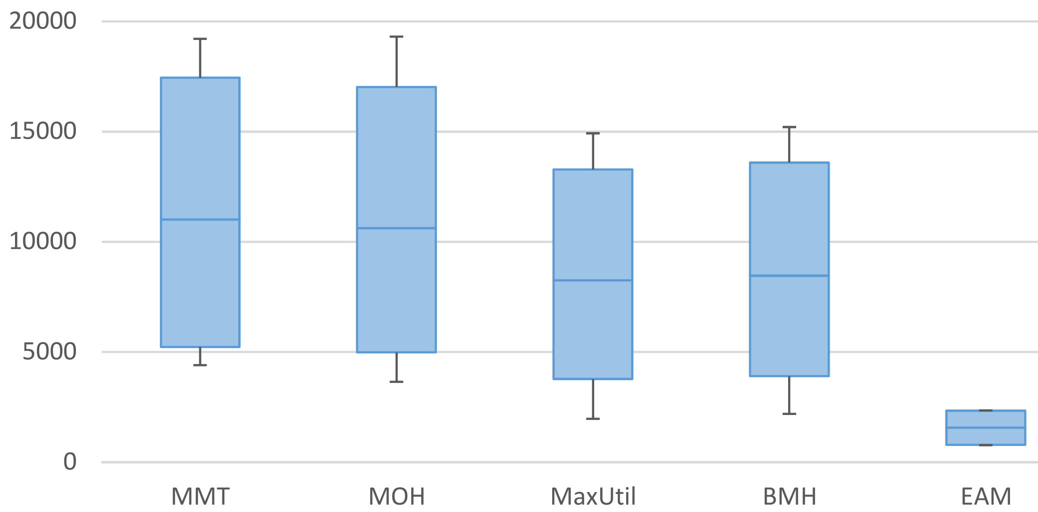


Figure 5.8: The box plots for the number of host shutdowns by MMT, MOH, MaxUtil, BMH and EAM

5.4 Discussion

The difference is significant as the average energy consumption by MMT is more than 116% higher than EAM. On average, EAM has 98.92 kiloWatt less energy

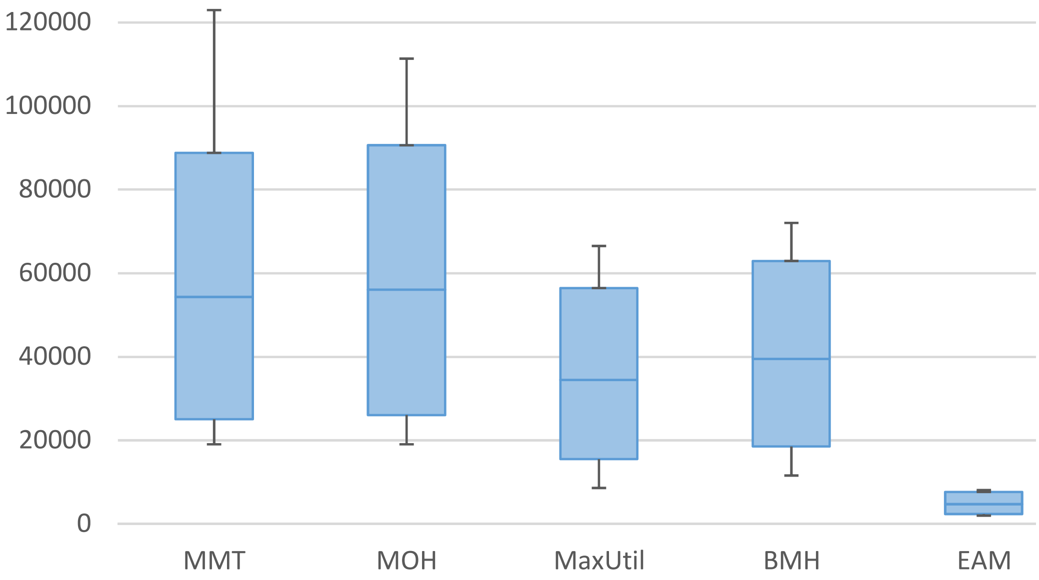


Figure 5.9: The box plots for the number of VM migrations by MMT, MOH, MaxUtil, BMH and EAM

consumption than MMT [9], which can be translated into the deduction of more than 38 metric tons carbon emission annually [1]. This level of energy saving is on a system with only 800 hosts. For systems with more hosts this can add up to a substantial energy saving and Carbon emission deduction. Moreover, on average, MMT takes 5.39 times longer than our proposed EAM mechanism to execute VMs.

On average, MOH [83], MaxUtil [87] and BMH [89] have 118%, 81% and 88% higher energy consumption than EAM and take 3.7, 2 and 1.9 times longer than EAM to execute the workloads (mean execution time), respectively.

BMH and EAM both present an adaptive approach in response to the changes in resource requirements and available resources using micro level observation records of VMs and hosts. EAM demonstrated strong dominance over competing strategies by minimizing energy consumption and execution time simultaneously.

[1] According to the ratio reported in [23].

Such dominance can be associated with the inclusiveness of both VM mapping and VM migration where they are modelled in a single Bayesian network. A single integrated Bayesian network enables resource hyper-visioning decisions, either VM mapping or VM migration, to reflect on previous hyper-visioning decisions, VM mapping or VM migration, for its decision process. As a result, EAM showed closeness to theoretical optimal energy consumption. It also shortened mean execution time significantly compared to the competing heuristics.

5.5 Summary

In this chapter, an adaptive energy efficient resource hyper-visioning mechanism is proposed, inclusive of both VM mapping and VM migration, based on micro level observation records of VM types and utilization levels. The proposed novel adaptive energy-efficient resource hyper-visor works by establishing an inference to relate the micro level observation records of VM types and utilization levels to the effect they have on hosts' utilization and consequently the energy consumption.

Our novel adaptive energy-efficient hyper-visor is evaluated against state-of-the-art heuristics from the literature and heuristics proposed in the earlier chapters where micro level observations were used. It represented a statistically significant reduction in energy consumption and shortening of mean execution time.

This chapter described the enhancement of the VM migration process to adaptively migrate a VM that is the most likely VM to reduce energy consumption. It also presented the invention of a novel adaptive resource hyper-visioning mechanism, inclusive of VM mapping and VM migration that significantly reduced total energy consumption and shortened mean execution time.

Chapter 6

Conclusions and Future Research Directions

THE technical details of our proposed hyper-visioning strategies are explained in previous chapters. This chapter summarizes the research done on energy-efficient resource hyper-visioning in private Cloud proposed in this thesis and highlights the main findings. It then reviews open research problems in the area and outlines potential future research directions.

This chapter is organized as follows. Section 6.1 presents an overview of the proposed energy-efficient policies and mechanisms in private Clouds. The main contributions of this thesis is summarized in Section 6.2. Section 6.3 explains the limitations of the research. It is then followed by future research directions in Section 6.4 and final remarks in Section 6.5.

6.1 Overview of Thesis

The research aimed at answering five research questions regarding energy-efficient resource hyper-visioning in private Cloud. To answer the research questions, energy-efficient resource hyper-visioning policies and adaptive mechanisms are proposed using macro and micro level observation records. The overview of the thesis is as follows:

- **Chapter 1** introduced energy-efficient resource hyper-visioning in private Cloud. The problems associated with high level of energy consumption is discussed. Chapter 1 also detailed the key elements of this research: scope of the research, research questions, hypotheses, methodology and contributions.

- **Chapter 2** reviewed the related literature on resource hyper-visioning in Cloud, in general. First common research objectives are noted and their relationship explained. Then their association with energy consumption is clarified. Later energy contributors, measures and models are discussed. Research conducted in reducing energy consumption by arbitrating the available resources are reviewed. Reviewed resource hyper-visioning studies are inclusive of VM mapping and VM migration related studies.

- **Chapter 3** investigated the outcome of six VM mapping policies, in terms of energy consumption, when workload properties are arbitrarily altered. It is then proven that a single VM mapping policy has its strengths and weaknesses based on the state of the system and workload properties. Therefore, a dominating VM mapping policy might be dominated by other VM mapping policy when the settings are changed. Chapter 3 then presented a novel Bayesian based VM mapping mechanism that adaptively switches between available VM mapping policies in order to reduce total energy consumption using the macro level observation records of energy consumption levels. The presented Bayesian inference relates the system's current state, in terms of average utilization; workload property (average arrival rate); and the deployed VM mapping policy to the observed energy consumption level. The results indicated that our adaptive VM mapping mechanism has similar results to the best performing VM mapping policy, on average.

- **Chapter 4** covered the introduction of two VM migration policies for when the system encounters an imbalance problem. The outcome proved a significant reduction of total energy consumption and shortening of mean execution time in comparison with a situation where no migration was carried out. The evaluation against state-of-the-art policies demonstrated statistically significant improvements in terms of total energy consumption and mean execution time.

- **Chapter 5** proposed two adaptive mechanisms: adaptive VM migration mechanism and adaptive resource hyper-visioning mechanism (inclusive of both VM mapping and VM migration). Both mechanisms used micro level observation records in their Bayesian inference. However, Bayesian inferences are made based on different observations from the system. And the adaptive energy-efficient resource hyper-visor is inclusive of both VM mapping and VM migration in its inference. The results strongly support the improvement made by the mechanisms. Nevertheless, our adaptive energy-efficient resource hyper-visioning mechanism demonstrated strong dominance over state-of-the-art heuristics from the literature and our proposed VM migration policy and adaptive mechanisms.

6.2 Contributions

The significance of this study can be categorized into the three areas where we suggest a different perspective in the field of energy-efficient resource hyper-visioning. These categories are as follows:

- *Sub-optimality of the policies.* Simulation of the deployment of multiple VM mapping policies in a small scale private Cloud proved that changes in the resource requirements by VMs and alteration of average arrival rate changes the outcome, in terms of energy consumption. A dominant result by a VM mapping policy is significantly dominated by another VM mapping policy when the settings, in terms of system and workload, are changed. We suggested that a well performing VM mapping policy might not be the optimal VM mapping policy in all settings of the system and workload.

- *The need for adaptability and learning.* Because of the dynamic nature of a Cloud system, private Cloud included, it is essential for an energy-efficient resource hyper-visor to be adaptive and be able to learn from the observed records of the system and the outcome of hyper-visioning decisions. A resource hyper-visioning mechanism that is adaptive and has a learning ability is more likely to serve the system in an energy-efficient manner. Our proposed adaptive mechanisms on VM mapping, VM migration and resource hyper-visioning (inclusive of both VM mapping and VM migration in a single adaptive learning phase) proved that the adaptability and learning can improve the outcome of resource hyper-visioning decisions significantly.

- *Importance of inclusion of VM mapping and VM migration in a single mechanism.* In this research we proposed adaptive energy-efficient mechanisms for VM mapping and VM migration, separately. However, the results, in terms of energy consumption and mean execution time, were substantially improved when VM mapping and VM migration are modelled as an integrated adaptive energy-efficient resource hyper-visioning mechanism. The reason lies in the idea that modelling VM mapping and VM migration in a single mechanism allows the model to reflect on any resource hyper-visioning decision, either VM mapping and VM migration.

6.3 Limitations

This thesis has the following limitations:

- The results are simulated values for energy consumption and mean execution time. Although the simulated values are driven from the actual energy consumption level by the presumed types of hosts, actual implementation of the proposed policies and mechanisms could provide a more accurate set of values.

- Because of some limitations on the simulation packages (not allowing change of VM mapping policy at run-time in one simulation package and not supporting VM migration in the other simulation package), two packages are used for simulation purposes. They use similar values for energy and execution time estimations but slight variations are possible.

- Proposed VM mapping, VM migration and hyper-visioning mechanisms are centralized. They are evaluated when there are a certain number of hosts. It has not been tested whether a dramatic increase in the number of hosts might add to the required calculation in mechanisms' learning phase.

6.4 Future Research Directions

Despite contributions of the research in the field of energy-efficient resource hyper-visioning in private Clouds, there are open challenges for future research to further advance the field of study.

6.4.1 Learning Methodologies

In this research Bayesian inference is used to provide the system with learning ability to adaptively respond to the changes in the system and workload. However, there are other methodologies than can be utilized and their strength in efficiently reducing energy consumption level tested in the field of study.

6.4.2 Adaptive and Distributed

Proposed policies and mechanisms are centralized approaches where a focal decision making point is in charge of resource hyper-visioning. It has its potential complications in the case of a large scale system. Although private Clouds do not necessarily exhibit very large scale systems, a distributed resource hyper-visioning can potentially reduce the computations, in the focal point, and further improve energy consumption as well as execution time.

6.4.3 Inclusion of Network Topology

In this research, the network bandwidth available to hosts was one of the deciding factors when VMs were migrated between the hosts. However, the network topology determines the physical connections and distance between the hosts. Multiple combinations of hosts can have the same network bandwidth available between them but the actual cost of the migration depends on how far the hosts are in terms of the nodes in the network and the physical distance between them.

6.5 Final Remarks

Clouds have transformed the IT industry by providing platforms, in different levels and by providing services for science, business and personal applications. The energy-efficient hyper-visioning of resources in a private Cloud, in particular the adaptive resource hyper-visioning, investigated in this thesis, will enable private Clouds to adaptively provide resources for the workloads assigned to them with low energy consumption and therefore low costs and carbon dioxide emissions. Research, such as that presented in this thesis, will motivate further investigation and innovation in the context of adaptive energy-efficient resource hyper-visioning in private Cloud.

Bibliography

[1] The Cloud Computing and Distributed Systems (CLOUDS) Laboratory, University of Melbourne. `http:\www.cloudbus.org`. accessed: Feb-2017.

[2] Abdelzahir Abdelmaboud, Dayang NA Jawawi, Imran Ghani, Abubakar Elsafi, and Barbara Kitchenham. Quality of service approaches in cloud computing: A systematic mapping study. *Journal of Systems and Software*, 101:159–179, 2015.

[3] Sherif Abdelwahed, Nagarajan Kandasamy, and Sandeep Neema. Online control for self-management in computing systems. In *Proceedings of the 10th IEEE Symposium on Real-Time and Embedded Technology and Applications*, pages 368–375. IEEE, 2004.

[4] Aldeida Aleti and Indika Meedeniya. Component deployment optimisation with bayesian learning. In *Proceedings of the 14th international ACM Sigsoft symposium on Component based software engineering*, pages 11–20. ACM, 2011.

[5] Hakan Aydin, Rami Melhem, Daniel Mossé, and Pedro Mejía-Alvarez. Power-aware scheduling for periodic real-time tasks. *IEEE Transactions on Computers*, 53(5):584–600, 2004.

[6] Sławomir Bąk, Marcin Krystek, Krzysztof Kurowski, Ariel Oleksiak, Wojciech Piątek, and Jan Wąglarz. GSSIM–a tool for distributed computing experiments. *Scientific Programming*, 19(4):231–251, 2011.

[7] Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield. Xen and the art of virtualization. *ACM SIGOPS Operating Systems Review*, 37(5):164–177, 2003.

[8] Anton Beloglazov and Rajkumar Buyya. Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers. In *Proceedings of the 8th International Workshop on Middleware for Grids, Clouds and e-Science*, page 4. ACM, 2010.

[9] Anton Beloglazov and Rajkumar Buyya. 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*, 24(13):1397–1420, 2012.

[10] Norman Bobroff, Andrzej Kochut, and Kirk Beaty. Dynamic placement of virtual machines for managing sla violations. In *Proceedings of 10th IFIP/IEEE International Symposium on Integrated Network Management*, pages 119–128. IEEE, 2007.

[11] Richard Brown et al. Report to congress on server and data center energy efficiency: Public law 109-431. *Lawrence Berkeley National Laboratory*, 2008.

[12] Iksen Çaglar and Deniz Turgay Altilar. An energy efficient vm allocation approach for data centers. In *Proceedings of 2nd IEEE 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), pages 240–244. IEEE, 2016.

[13] Rodrigo N Calheiros, Rajiv Ranjan, Anton Beloglazov, César AF De Rose, and Rajkumar Buyya. 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, 2011.

[14] Rodrigo N Calheiros, Rajiv Ranjan, César AF De Rose, and Rajkumar Buyya. CloudSim: A novel framework for modeling and simulation of cloud computing infrastructures and services. *arXiv preprint arXiv:0903.2525*, 2009.

[15] Michael Cardosa, Madhukar Korupolu, and Aameek Singh. Shares and utilities based power consolidation in virtualized server environments. In *Proceedings of 11th IFIP/IEEE International Symposium on Integrated Network Management*, pages 327 – 334. IEEE, 2009.

[16] Feifei Chen, John Grundy, Yun Yang, Jean-Guy Schneider, and Qiang He. Experimental analysis of task-based energy consumption in cloud computing systems. In *Proceedings of 4th ACM/SPEC International Conference on Performance Engineering*, pages 295–306. ACM, 2013.

[17] Susanta Nanda Tzi-cker Chiueh and Stony Brook. A survey on virtualization technologies. *RPE Report*, pages 1–42, 2005.

[18] Christopher Clark, Keir Fraser, Steven Hand, Jacob Gorm Hansen, Eric Jul, Christian Limpach, Ian Pratt, and Andrew Warfield. Live migration of virtual machines. In *Proceedings of the 2nd conference on Symposium on*

Networked Systems Design & Implementation-Volume 2, pages 273–286. USENIX Association, 2005.

[19] Nicola Cordeschi, Mohammad Shojafar, and Enzo Baccarelli. Energy-saving self-configuring networked data centers. *Computer Networks*, 57(17):3479–3491, 2013.

[20] Ewa Deelman, James Blythe, Yolanda Gil, Carl Kesselman, Gaurang Mehta, Sonal Patil, Mei-Hui Su, Karan Vahi, and Miron Livny. Pegasus: Mapping scientific workflows onto the grid. In *Proceedings of Grid Computing*, pages 11–20. Springer, 2004.

[21] Ewa Deelman and Ann Chervenak. Data management challenges of data-intensive scientific workflows. In *Proceedings of 8th IEEE International Symposium on Cluster Computing and the Grid*, pages 687–692. IEEE, 2008.

[22] Ewa Deelman, Gurmeet Singh, Miron Livny, Bruce Berriman, and John Good. The cost of doing science on the cloud: the montage example. In *Proceedings of the ACM/IEEE Conference on Supercomputing*, pages 1–12. IEEE Press, 2008.

[23] Pierre Delforge. America's data centers consuming and wasting growing amounts of energy. *Natural Resource Defence Councle*, 2014. `http://www.nrdc.org/energy/data-center-efficiency-assessment.asp`, accessed: Jun-2016.

[24] Eugen Feller, Louis Rilling, and Christine Morin. Energy-aware ant colony based workload placement in clouds. In *Proceedings of 12th IEEE/ACM International Conference on Grid Computing*, pages 26–33. IEEE Computer Society, 2011.

[25] Wu-chun Feng, Xizhou Feng, and Rong Ge. Green supercomputing comes of age. *IT professional*, 10(1):17–23, 2008.

[26] Mauro Gaggero and Luca Caviglione. A predictive control approach for energy-aware consolidation of virtual machines in cloud computing. In *Proceedings of 53rd IEEE Conference on Decision and Control*, pages 5308–5313. IEEE, 2014.

[27] Mauro Gaggero and Luca Caviglione. Model predictive control for the placement of virtual machines in cloud computing applications. In *Proceedings of American Control Conference (ACC)*, pages 1987–1992. IEEE, 2016.

[28] Anshul Gandhi, Mor Harchol-Balter, Rajarshi Das, and Charles Lefurgy. Optimal power allocation in server farms. In *Proceedings of the 11th International Joint Conference on Measurement and Modeling of Computer Systems*, pages 157 – 168. ACM, 2009.

[29] Yongqiang Gao, Haibing Guan, Zhengwei Qi, Yang Hou, and Liang Liu. A multi-objective ant colony system algorithm for virtual machine placement in cloud computing. *Journal of Computer and System Sciences*, 79(8):1230–1242, 2013.

[30] Saurabh Kumar Garg, Chee Shin Yeo, Arun Anandasivam, and Rajkumar Buyya. Energy-efficient scheduling of HPC applications in cloud computing environments. *arXiv preprint arXiv:0909.1146*, 2009.

[31] Rahul Ghosh, Vijay K Naik, and Kishor S Trivedi. Power-performance trade-offs in iaas cloud: A scalable analytic approach. In *Proceedings of 41st International IEEE/IFIP Conference on Dependable Systems and Networks Workshops (DSN-W)*, pages 152–157. IEEE, 2011.

[32] Robert P Goldberg. Survey of virtual machine research. *Computer*, 7(6):34–45, 1974.

[33] Tarun Goyal, Ajit Singh, and Aakanksha Agrawal. CloudSim: simulator for cloud computing infrastructure and modeling. *Procedia Engineering*, 38:3566–3572, 2012.

[34] Qiang Guan, Ziming Zhang, and Song Fu. Ensemble of bayesian predictors for autonomic failure management in cloud computing. In *Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN)*, pages 1–6. IEEE, 2011.

[35] Qiang Guan, Ziming Zhang, and Song Fu. Ensemble of bayesian predictors and decision trees for proactive failure management in cloud computing systems. *Journal of Communications*, 7(1):52–61, 2012.

[36] J Michael Harrison. Dynamic scheduling of a multiclass queue: Discount optimality. *Operations Research*, 23(2):270–282, 1975.

[37] Jerry Honeycutt. Microsoft virtual PC 2004 technical overview. *Microsoft, Nov*, 2003.

[38] Ching-Hsien Hsu, Shih-Chang Chen, Chih-Chun Lee, Hsi-Ya Chang, Kuan-Chou Lai, Kuan-Ching Li, and Chunming Rong. Energy-aware task consolidation technique for cloud computing. In *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on*, pages 115–121. IEEE, 2011.

[39] Ching-Hsien Hsu, Kenn Slagter, Shih-Chang Chen, and Yeh-Ching Chung. Optimizing energy consumption with task consolidation in clouds. *Information Sciences*, 258:452–462, 2014.

[40] International Energy Agency (IEA). Medium term renewable energy market report - executive summary, 2014. `https://www.iea.org/Textbase/npsum/MTrenew2014sum.pdf,accessed:August-2016`.

[41] International Energy Agency (IEA). World energy outlook 2015, 2015. `http://www.worldenergyoutlook.org/weo2015/`, accessed: August-2016.

[42] VMWare Inc. http://www.vmware.com/. accessed: July-2016.

[43] Sandy Irani, R Gupta, and S Shukla. Competitive analysis of dynamic power management strategies for systems with multiple power savings states. In *Proceedings of the Conference on Design, Automation and Test in Europe*, page 117. IEEE, 2002.

[44] Leila Ismail and Abbas Fardoun. Eats: Energy-aware tasks scheduling in cloud computing systems. *Procedia Computer Science*, 83:870–877, 2016.

[45] Russ Joseph and Margaret Martonosi. Run-time power estimation in high performance microprocessors. In *Proceedings of the International Symposium on Low Power Electronics and Design*, pages 135–140. ACM, 2001.

[46] Gaganjot Kaur. Study of comparison of various cloud computing simulators. In *Proceedings of 2nd National (INDIAN) Conference in Intelligent Computing & Communication, April 26-27*. Organized by Departement of IT, GCET Greater Nodia, India, 2013.

[47] Harmeet Kaur and Rama Krishna Challa. A new hybrid virtual machine scheduling scheme for public cloud. In *Proceedings of 5th International Conference on Advanced Computing & Communication Technologies (ACCT)*, pages 495–500. IEEE, 2015.

[48] Gunjan Khanna, Kirk Beaty, Gautam Kar, and Andrzej Kochut. Application performance management in virtualized server environments. In *Network Operations and Management Symposium, 2006. NOMS 2006. 10th IEEE/IFIP*, pages 373–381. IEEE, 2006.

[49] Kyong Hoon Kim, Anton Beloglazov, and Rajkumar Buyya. Power-aware provisioning of cloud resources for real-time services. In *Proceedings of the 7th International Workshop on Middleware for Grids, Clouds and e-Science*, pages 1–6. ACM, 2009.

[50] Kyong Hoon Kim, Anton Beloglazov, and Rajkumar Buyya. Power-aware provisioning of virtual machines for real-time cloud services. *Concurrency and Computation: Practice and Experience*, 23(13):1491–1505, 2011.

[51] Kyong Hoon Kim, Rajkumar Buyya, and Jong Kim. Power aware scheduling of bag-of-tasks applications with deadline constraints on DVS-enabled clusters. In *Proceedings of CCGRID*, pages 541–548, 2007.

[52] Nakku Kim, Jungwook Cho, and Euiseong Seo. Energy-credit scheduler: an energy-aware virtual machine scheduler for cloud systems. *Future Generation Computer Systems*, 32:128–137, 2014.

[53] Avi Kivity, Yaniv Kamay, Dor Laor, Uri Lublin, and Anthony Liguori. KVM: the linux virtual machine monitor. In *Proceedings of the Linux Symposium*, volume 1, pages 225–230, 2007.

[54] Dzmitry Kliazovich, Pascal Bouvry, and Samee Ullah Khan. GreenCloud: a packet-level simulator of energy-aware cloud computing data centers. *The Journal of Supercomputing*, 62(3):1263–1283, 2012.

[55] Dzmitry Kliazovich, Pascal Bouvry, and Samee Ullah Khan. Simulating communication processes in energy-efficient cloud computing sys-

tems. In *Proceedings of 1st International Conference on Cloud Networking (CLOUDNET)*, pages 215–217. IEEE, 2012.

[56] Jonathan Koomey. Growth in data center electricity use 2005 to 2010. *Oakland, CA: Analytics Press. August*, 1:2010, 2011.

[57] Jonathan G Koomey. Estimating total power consumption by servers in the us and the world, 2007. *Lawrence Berkeley National Laboratory, Berkeley, CA, available at: http://hightech.lbl.gov/documents/DATA_CENTERS/svrpwrusecompletefinal.pdf*, 2007.

[58] Kevin B Korb and Ann E Nicholson. *Bayesian artificial intelligence*. CRC press, 2010.

[59] J Kovar. Data center power consumption grows less than expected: Report, 2011.

[60] P Venkata Krishna. Honey bee behavior inspired load balancing of tasks in cloud computing environments. *Applied Soft Computing*, 13(5):2292–2303, 2013.

[61] Krzysztof Kurowski, Jarek Nabrzyski, Ariel Oleksiak, and Jan Weglarz. Grid scheduling simulations with GSSIM. In *Proceedings of International Conference on Parallel and Distributed Systems*, pages 1–8. IEEE, 2007.

[62] Dara Kusic, Jeffrey O. Kephart, James E. Hanson, Nagarajan Kandasamy, and Guofei Jiang. Power and performance management of virtualized computing environments via lookahead control. In *Proceedings of International Conference on Autonomic Computing*, pages 3–12, June 2008.

[63] Dara Kusic, Jeffrey O Kephart, James E Hanson, Nagarajan Kandasamy, and Guofei Jiang. Power and performance management of virtualized computing environments via lookahead control. *Cluster computing*, 12(1):1–15, 2009.

[64] Young Choon Lee and Albert Y Zomaya. Energy efficient utilization of resources in cloud computing systems. *The Journal of Supercomputing*, 60(2):268–280, 2012.

[65] Jiandun Li, Junjie Peng, Zhou Lei, and Wu Zhang. An energy-efficient scheduling approach based on private clouds. *Journal of Information & Computational Science*, 8(4):716–724, 2011.

[66] Seung-Hwan Lim, Bikash Sharma, Gunwoo Nam, Eun Kyoung Kim, and Chita R Das. MDCSim: A multi-tier data center simulation, platform. In *Proceedings of International Conference on Cluster Computing and Workshops*, pages 1–9. IEEE, 2009.

[67] Minghong Lin, Adam Wierman, Lachlan LH Andrew, and Eno Thereska. Dynamic right-sizing for power-proportional data centers. *IEEE/ACM Transactions on Networking (TON)*, 21(5):1378–1391, 2013.

[68] Somnath Mazumdar and Marco Pranzo. Power efficient server consolidation for cloud data center. *Future Generation Computer Systems*, 70:4–16, 2017.

[69] Peter Mell and Tim Grance. The NIST definition of cloud computing. national institute of standards and technology. *Information Technology Laboratory, Version 15*, 2009. http://www.csrc.nist.gov/groups/SNS/cloudcomjputing/index.html.

[70] Haibo Mi, Huaimin Wang, Gang Yin, Yangfan Zhou, Dianxi Shi, and Lin Yuan. Online self-reconfiguration with performance guarantee for energy-efficient large-scale cloud computing data centers. In *Proceedings of IEEE International Conference on Services Computing*, pages 514–521, 2010.

[71] Rich Miller. Google's energy story: High efficiency, huge scale. *Retrieved*, 12(10):2013, 2011.

[72] Justin D Moore, Jeffrey S Chase, Parthasarathy Ranganathan, and Ratnesh K Sharma. Making scheduling "Cool": Temperature-aware workload placement in data centers. In *Proceedings of USENIX Annual Technical Conference, General Track*, pages 61–75, 2005.

[73] Ahuva W. Mu'alem and Dror G. Feitelson. Utilization, predictability, workloads, and user runtime estimates in scheduling the IBM SP2 with backfilling. *IEEE Transactions on Parallel and Distributed Systems*, 12(6):529–543, 2001.

[74] Tom Oinn, Matthew Addis, Justin Ferris, Darren Marvin, Martin Senger, Mark Greenwood, Tim Carver, Kevin Glover, Matthew R Pocock, Anil Wipat, and Peter Li. Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 20(17):3045–3054, 2004.

[75] Rina Panigrahy, Kunal Talwar, Lincoln Uyeda, and Udi Wieder. Heuristics for vector bin packing. *research. microsoft. com*, 2011.

[76] KyoungSoo Park and Vivek S Pai. CoMon: a mostly-scalable monitoring system for planetlab. *ACM SIGOPS Operating Systems Review*, 40(1):65–74, 2006.

[77] C Greg Plaxton, Yu Sun, Mitul Tiwari, and Harrick Vin. Reconfigurable resource scheduling. In *Proceedings of the 18th Annual ACM Symposium on Parallelism in Algorithms and Architectures*, pages 93–102. ACM, 2006.

[78] Ramya Raghavendra, Parthasarathy Ranganathan, Vanish Talwar, Zhikui Wang, and Xiaoyun Zhu. No power struggles: Coordinated multi-level power management for the data center. In *Proceedings of ACM SIGARCH Conference on Computer Architecture News*, pages 48–59. ACM, 2008.

[79] Veena Reddy Reguri, Swetha Kogatam, and Melody Moh. Energy efficient traffic-aware virtual machine migration in green cloud data centers. In *Proceedings of 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)*, pages 268–273. IEEE, 2016.

[80] Shenyuan Ren, Ligang He, Huanzhou Zhu, Zhuoer Gu, Wei Song, and Jiandong Shang. Developing power-aware scheduling mechanisms for computing systems virtualized by xen. *Concurrency and Computation: Practice and Experience*, 2016. DOI: 10.1002/cpe.3888.

[81] Christian Robert and William M Boldstad. Understanding computational bayesian statistics. *CHANCE*, 25(2):57–58, 2012.

[82] Mohsen Amini Salehi and Rajkumar Buyya. Adapting market-oriented scheduling policies for cloud computing. In *Proceedings of the International Conference on Algorithms and Architectures for Parallel Processing*, pages 351–362. Springer, 2010.

[83] Ahmed Sallam and Kenli Li. A multi-objective virtual machine migration policy in cloud systems. *The Computer Journal*, 57(2):195–204, 2014.

[84] Chiyoung Seo, Sam Malek, and Nenad Medvidovic. Component-level energy consumption estimation for distributed java-based software systems. In *Proceedings of the International Symposium on Component-Based Software Engineering*, pages 97–113. Springer, 2008.

[85] Oshin Sharma and Hemraj Saini. Vm consolidation for cloud data center using median based threshold approach. *Procedia Computer Science*, 89:27–33, 2016.

[86] Sahar Sohrabi. The application of bayesian heuristic for scheduling in real-time private clouds. *World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering*, 10(3):505–512, 2016.

[87] Sahar Sohrabi and Irene Moser. The effects of hotspot detection and virtual machine migration policies on energy consumption and service levels in the cloud. *Procedia Computer Science*, 51:2794–2798, 2015.

[88] Sahar Sohrabi and Irene Moser. A survey on energy-aware cloud. *European Journal of Advances in Engineering and Technology*, 2:80–91, 2015.

[89] Sahar Sohrabi, Anthony Tang, Irene Moser, and Aldeida Aleti. Adaptive virtual machine migration mechanism for energy efficiency. In *Proceedings of the 5th International Workshop on Green and Sustainable Software*, pages 8–14. ACM, 2016.

[90] Sahar Sohrabi, Yun Yang, Irene Moser, and Aldeida Aleti. Energy-efficient adaptive virtual machine migration mechanism for private clouds. *Concurrency and Computation: Practice and Experience*, 2017. 10.1002/cpe.4170.

[91] Thamarai Selvi Somasundaram and Kannan Govindarajan. Cloudrb: a framework for scheduling and managing high-performance computing (hpc) applications in science cloud. *Future Generation Computer Systems*, 34:47–65, 2014.

[92] Mahalaxmi Sridhar and G. Rama Mohan Babu. Hybrid particle swarm optimization scheduling for cloud computing. In *Proceedings of IEEE International Conference on Advance Computing Conference (IACC)*, pages 1196–1200. IEEE, 2015.

[93] Shekhar Srikantaiah, Aman Kansal, and Feng Zhao. Energy aware consolidation for cloud computing. In *Proceedings of the Conference on Power Aware Computing and Systems*, pages 1–5, 2008.

[94] A Suresh and P Vijayakarthick. Improving scheduling of backfill algorithms using balanced spiral method for cloud metascheduler. In *Proceedings of International Conference on Recent Trends in Information Technology (ICRTIT)*, pages 624–627. IEEE, 2011.

[95] Chunqiang Tang, Malgorzata Steinder, Michael Spreitzer, and Giovanni Pacifici. A scalable application placement controller for enterprise data centers. In *Proceedings of the 16th International Conference on World Wide Web*, pages 331–340. ACM, 2007.

[96] Qinghui Tang, Sandeep Kumar S Gupta, and Georgios Varsamopoulos. Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: A cyber-physical approach. *IEEE Transactions on Parallel and Distributed Systems*, 19(11):1458–1472, 2008.

[97] Haluk Topcuoglu, Salim Hariri, and Min-you Wu. Performance-effective and low-complexity task scheduling for heterogeneous computing. *IEEE Transactions on Parallel and Distributed Systems*, 13(3):260–274, 2002.

[98] Jordi Toress. Green computing: The next wave in computing. *UPC Technical University of Catalonia*, 2010. `0.http://seminarisempresa.fib.upc.edu/aulesempresa/2010/programes/1231640F.Palmierietal.BSC/Llista-documents/02/document/5.%20BSC%20GreenComputing.AulesEmpresa%20FOB.2010.pdf,accessed:December-2016.`

[99] Dan Tsafrir, Yoav Etsion, and Dror G Feitelson. Backfilling using system-generated predictions rather than user runtime estimates. *IEEE Transactions on Parallel and Distributed Systems*, 18(6):789–803, 2007.

[100] Rich Uhlig, Gil Neiger, Dion Rodgers, Amy L Santoni, Fernando CM Martins, Andrew V Anderson, Steven M Bennett, Alain Kagi, Felix H Leung, and Larry Smith. Intel virtualization technology. *Computer*, 38(5):48–56, 2005.

[101] Akshat Verma, Gargi Dasgupta, Tapan Kumar Nayak, Pradipta De, and Ravi Kothari. Server workload analysis for power minimization using consolidation. In *Proceedings of Conference on USENIX Annual technical conference*, pages 28–28. USENIX Association, 2009.

[102] Amandeep Verma and Sakshi Kaushal. Bi-criteria priority based particle swarm optimization workflow scheduling algorithm for cloud. In *Proceedings of IEEE Conference on Recent Advances in Engineering and Computational Sciences (RAECS)*, pages 1–6. IEEE, 2014.

[103] Amandeep Verma and Sakshi Kaushal. Cost minimized pso based workflow scheduling plan for cloud computing. *International Journal of Information Technology and Computer Science*, pages 37–43, 2015.

[104] Gregor Von Laszewski, Lizhe Wang, Andrew J Younge, and Xi He. Power-aware scheduling of virtual machines in DVFS-enabled clusters. In *Proceedings of IEEE International Conference on Cluster Computing and Workshops*, pages 1–10. IEEE, 2009.

[105] William Voorsluys, James Broberg, and Rajkumar Buyya. Introduction to cloud computing. *Cloud computing: Principles and paradigms*, pages 1–41, 2011.

[106] Lee Wang, Howard Jay Siegel, Vwani P Roychowdhury, and Anthony A Maciejewski. Task matching and scheduling in heterogeneous computing environments using a genetic-algorithm-based approach. *Journal of parallel and distributed computing*, 47(1):8–22, 1997.

[107] Wei Wang, Guosun Zeng, Daizhong Tang, and Jing Yao. Cloud-DLS: Dynamic trusted scheduling for cloud computing. *Expert Systems with Applications*, 39(3):2321–2329, 2012.

[108] Xianbin Wang, Guangjie Han, Xiaojiang Du, and Joel JPC Rodrigues. Mobile cloud computing in 5G: Emerging trends, issues, and challenges [guest editorial]. *Network, IEEE*, 29(2):4–5, 2015.

[109] Andrew Whitaker, Marianne Shaw, and Steven D Gribble. Denali: Lightweight virtual machines for distributed and networked applications. Technical report, Citeseer, 2002. `https://pdfs.semanticscholar.org/d4f4/1241d9814f06054454ede2b8dcfa3fe47730.pdf, accessed:December-2016`.

[110] Ying Yidu Xiong and Yan Yan Wu. Cloud computing resource schedule strategy based on pso algorithm. In *Proceedings of Applied Mechanics and Materials*, pages 1332–1336. Trans Tech Publications, Switzerland, 2014.

[111] Jia Yu and Rajkumar Buyya. Scheduling scientific workflow applications with deadline and budget constraints using genetic algorithms. *Scientific Programming*, 14(3-4):217–230, 2006.

[112] Dong Yuan, Yun Yang, Xiao Liu, and Jinjun Chen. A data placement strategy in scientific cloud workflows. *Future Generation Computer Systems*, 26(8):1200–1214, 2010.

[113] Dong Yuan, Yun Yang, Xiao Liu, and Jinjun Chen. On-demand minimum cost benchmarking for intermediate dataset storage in scientific cloud workflow systems. *Journal of Parallel and Distributed Computing*, 71(2):316–332, 2011.

[114] Wei Zheng and Rizos Sakellariou. Budget-deadline constrained workflow planning for admission control. *Journal of Grid Computing*, 11(4):633–651, 2013.

[115] Linan Zhu, Qingshui Li, and Lingna He. Study on cloud computing resource scheduling strategy based on the ant colony optimization algorithm. *IJCSI International Journal of Computer Science Issues*, 9(5):1694–0814, 2012.

Appendix A

NOTATIONS

Notation	Definition
AR	set of average arrival rates, $AR = \{AR_1, AR_2, \cdots, AR_i\}$
C	set of constraints
$c_{excludedHosts}$	constraint on excluding hosts as candidate hosts to receive the migrating virtual machine
c_{maxVMs}	constraint on the maximum number of virtual machines on a host
$c_{migratable}$	constraint on selecting only the virtual machines that can be moved (migrated)
CU	current utilization level
H	set of available hosts, $H = \{h_1, h_2, \cdots, h_n\}$, $n \in N$
h_i	host i
HU_i	CPU utilization of h_i
L	set of possible host utilization levels, $L = \{l_1, l_2\}$
l_i	host utilization in l_i level
M_i	memory share of virtual machine i, vm_i
MC_i	memory capacity of h_i

Mig	set of possible virtual machine and hosts combinations for migration. $Mig = \{mig_1, mig_2, \cdots, mig_p\}$, $p \in N$
mig	a possible combination of virtual machine and hosts for migration. $mig = (vm_q, h_j), vm_q \in VM_i, VM_i$ set of virtual machines on h_i, $h_i \in H$, $h_j \in H$ and $i \neq j$
MT	the migration time of a VM
NB_i	network bandwidth available to h_i
$p(\Delta^+)$	the probability of Δ^+
$p(l_i)$	probability of l_i
$p(t_i)$	probability of t_i
$p(l_i, t_j)$	probability of migrating a VM from type t_j from/to a host with l_i utilization level
$p(l_i, t_j \vert \Delta^+)$	probability of Δ^+ given l_i and t_j
$p(\Delta^+ \vert l_i, t_j)$	probability of l_i and t_j resulting in Δ^+
$p(AR_i)$	probability of AR_i
$p(CU_i)$	probability of CU_i
$p(pol_i)$	probability of pol_i
$p(AR_i, CU_j, Pol_k)$	probability of observing average arrival rate of AR_i, current utilization level of CU_j when VM mapping policy pol_i was deployed
$p(AR_i, CU_j, Pol_k \vert EC_l)$	probability of average arrival rate of AR_i, current utilization level of CU_j when VM mapping policy pol_i was deployed given the observed energy consumption level was in EC_1 range
$p(EC_l \vert AR_i, CU_j, Pol_k)$	Probability of observing an energy consumption level of EC_1 given average arrival rate of AR_i, current utilization level of CU_j when VM mapping policy pol_i was deployed

$p(R_i)$	probability of VMs having a utilization level in R_i range
pol_i	a VM mapping policy
PS	processing share of a virtual machine
R_i	a utilization range for VMs
t_i	type of a VM
VM_i	set of virtual machines, $VM = \{vm_1, vm_2, \cdots, vm_m\}$, $m \in N$ on host h_i
vm_i	virtual machine i
$VMType$	set of available virtual machine types, $VMType = \{t_1, t_2, t_3, t_4\}$
Δ	$\Delta = (HU_i + HU_j)_{postMigration} - (HU_i + HU_j)_{preMigration}$. The difference between host utilizations of the source and destination hosts in the migration process. If vm_q is migrated from h_i (meaning: h_i had imbalanced problem and $vm_q \in VM_i$ before migration) to h_j (vm_q is added to VM_j after migration)
Δ^+	$\Delta \geq 0$
Δ^-	$\Delta < 0$

Appendix B

ABBREVIATIONS

Abbreviation	Meaning
BI	Bayesian Inference
BN	Bayesian Network
ASM	Adaptive Switching Mechanism
MU	Maximum Utilization VM mapping policy
IB	Intensity Based VM mapping policy
GD	Greedy Deadline VM mapping policy
IGD	Intensity-based Greedy Deadline VM mapping policy
EL	Equal Load VM mapping policy
IEL	Intensity-based Equal Load VM mapping policy
AR	task Arrival Rate
CU	current CPU Utilization
EC	Energy Consumption
MMT	Minimum Migration Time VM migration policy
MedianMT	Median Migration Time VM migration policy
MaxUtil	Maximum Utilization VM migration policy
Asympt. Sig.	asymptotic significance