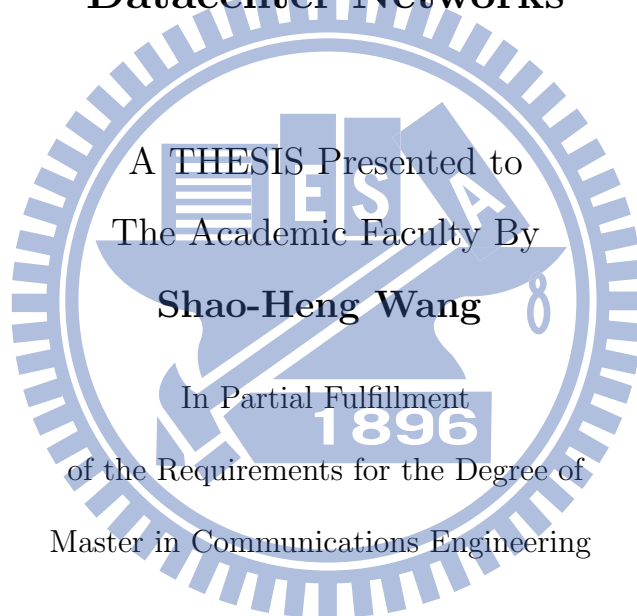


**Virtual Machine Placement for Energy
Efficiency and QoS in Software Defined
Datacenter Networks**



A THESIS Presented to
The Academic Faculty By
Shao-Heng Wang

In Partial Fulfillment
of the Requirements for the Degree of
Master in Communications Engineering

Institute of Communications Engineering

College of Electrical and Computer Engineering

National Chiao Tung University

2013

Copyright ©2013 by Shao-Heng Wang

Abstract

To provide effective and reliable services, cloud datacenters need parallel computing and virtualization techniques. This thesis presents an improved virtual machine VM placement mechanism, called Energy efficiency and Quality of Service (QoS) guarantee VM Placement (EQVMP) to overcome the problem of unbalanced traffic load in switching on and off VMs for the purpose of energy saving. EQVMP combines of three key techniques: (1)*hop reduction*, (2)*energy saving* and (3)*load balancing*. Hop reduction can regroup VMs to have lower traffic load among them. Energy saving techniques aim at choosing the appropriate servers. The proposed load balancing updates VM placement periodically. Our experimental results show that the proposed scheme can lower energy consumption and maintain QoS. We propose an evaluation score [1] to assess VM placement in terms of energy, delay and throughput. Comparing to other existing placement policies, our proposed mechanism can enhance system throughput by 25% and can have better evaluation score.

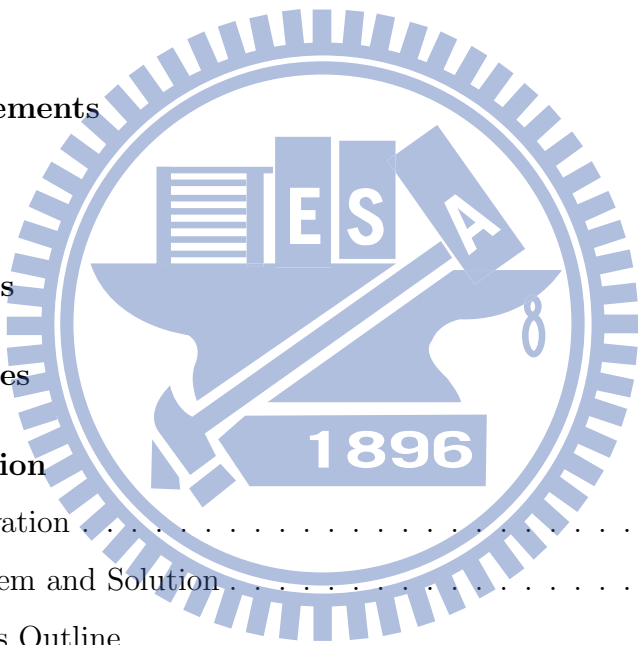
Acknowledgements

I would like to thank my parents and my older brother. They always give me endless supports. I especially thank Professor Li-Chun Wang and Charles H.-P. Wen who gave me many valuable suggestions in my research during these two years. I would not finish this work without his guidance and comments.

In addition, I am deeply grateful to my laboratory mates, Yin-Ming, Cheng-Wen, I-Cheng, I-Cen, Ssu-Han, and junior laboratory mates at Mobile Communications and Cloud Computing Laboratory at the Graduate Institute of Communications Engineering in National Chiao-Tung University. They provide me with a lot of assistance and share happiness with me.

Contents

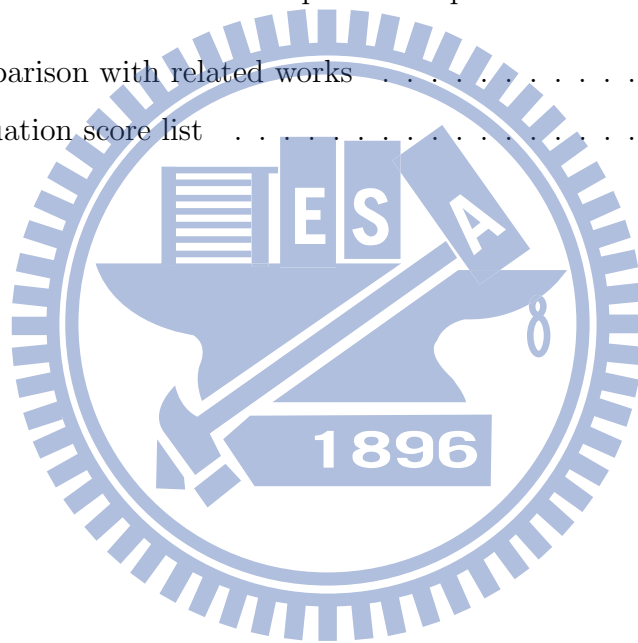
Abstract	i
Acknowledgements	ii
Contents	iii
List of Tables	v
List of Figures	vi
1 Introduction	1
1.1 Motivation	1
1.2 Problem and Solution	4
1.3 Thesis Outline	5
2 Background	6
2.1 Energy-aware VM placement	6
2.2 Delay-aware VM placement	7
2.3 Dynamic routing algorithm	8
2.4 Literature Survey	9
3 System Model and Problem Formulation	11
3.1 System model	11



3.2	Problem Formulation	14
4	Hop Reduction	16
4.1	Route Reduction	16
4.2	Graph Partition	17
4.3	Proposed Module	18
5	Energy Saving	21
5.1	Server vs Network Devices	21
5.2	energy efficiency algorithm	23
5.3	Proposed Module	24
6	Load Balancing	27
6.1	Network Management	27
6.2	Flow Routing	28
6.3	Proposed Module	30
7	Experimental Results	32
7.1	system performance	33
7.2	Update period	35
7.3	Comparison of different placement policies	37
7.4	Evaluation score	38
8	Conclusions	41
8.1	Summary	41
8.2	Future Research	42
	Bibliography	43
	Vita	47

List of Tables

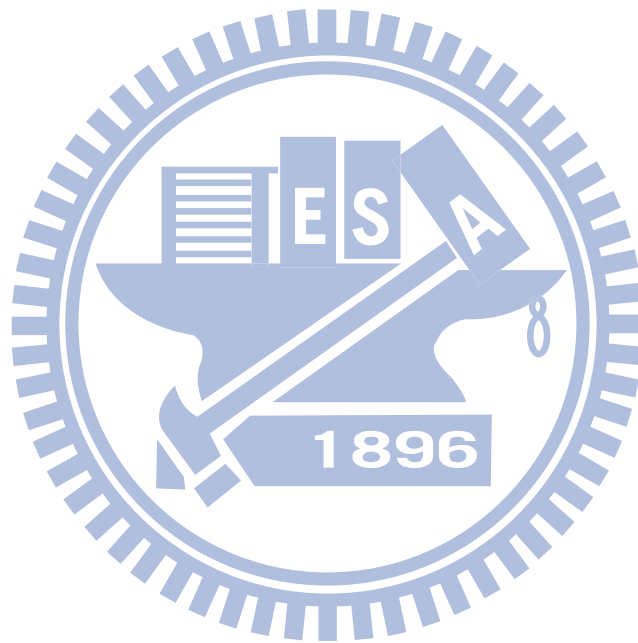
2.1	Comparison of related VM placement policies	10
7.1	Comparison with related works	38
7.2	Evaluation score list	40



List of Figures

1.1	CPU utilization and energy consumption [14].	2
1.2	Software defined network.	4
3.1	Energy-efficient with QoS-guarantee VM Placement algorithm in 3 phases.	12
3.2	Network topologies and corresponding matrices for different datacenter architectures.	13
4.1	Hop count expression in datacenter topology.	17
4.2	VM partitioning with traffic matrix and graph.	19
4.3	VM partitioning to reduce network delay.	20
5.1	VM placement before energy saving.	22
5.2	VM placement before energy saving.	25
5.3	VM placement after energy saving.	26
6.1	VM placement with load balance mechanism.	30
7.1	Throughput in different phases.	34
7.2	Delay in different phases.	34
7.3	Computational time of VM placement.	36
7.4	Performance with different update period.	36
7.5	Comparison between different VM placement policies.	37

7.6 Comparison between different awareness of VM placement. . . 39



Chapter 1

Introduction

Cloud computing research becomes a hot topic in recent years. To provide various kinds of applications and services, datacenters need sufficient bandwidth to maintain QoS for communication among millions of network components, resulting in consuming tremendous energy. Hence, how to save energy and to provision sufficient bandwidth are important issues. Finally, we propose our solution to resolve these issues.

1.1 Motivation

Datacenters are designed to provide reliable and scalable computing services for massive users. One of the most important things in datacenters is to provide efficient and fault-tolerant routing [2] [3]. Therefore, cloud computing must contain millions of servers and switches for different kinds of applications [4]. Based on U.S. Environmental Protection Agency's Data Center reports, the total power consumed by datacenters was 3 billion kWh in 2006 in the U.S., and will double in 2012 [5]. Obviously, energy consumption is an essential topic in datacenters.

VM placement is essential in datacenter [6] [7] [8]. Basically, current server hardware capacities are far beyond regular demands from users. In other words, most server resources are under-utilized. Owing to the virtualization technology, the resource utilization of physical machines can be greatly improved. Many research works on VM placement propose different approaches to improve energy efficiency [9] [10] [11]. However, most of them focus on saving energy. An aggressive placement policy [12] [13] can degrade network performance. For example, VMs with heavy traffic load can be congested in certain area of the network.

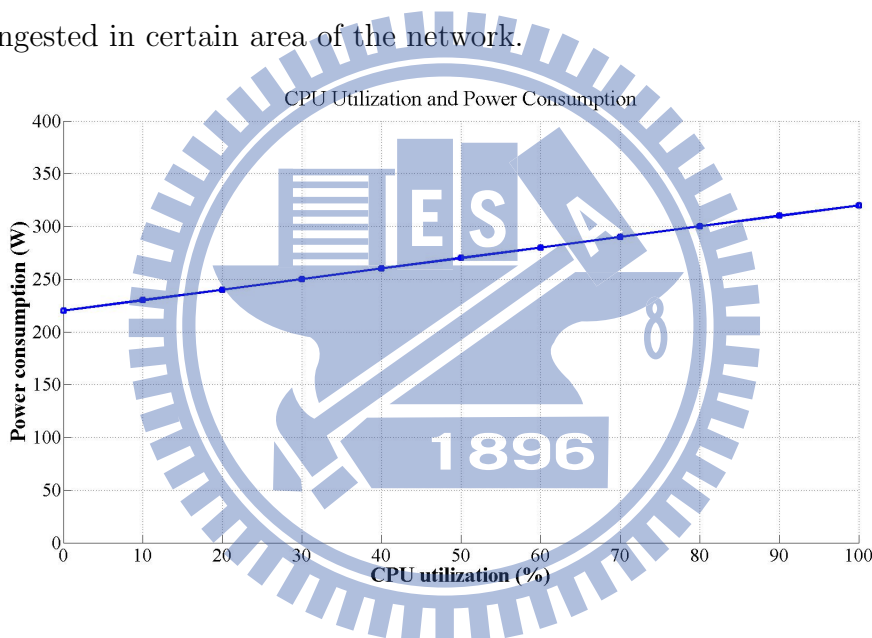


Figure 1.1: CPU utilization and energy consumption [14].

Although network performance in VM placement policy is an important issue, it is difficult to overlook the effects of energy consumption. Fig. 1.1 illustrates how processor energy efficiency (e.g., performance per watt) increases as server utilization increases for a typical workload. In the case of equal workloads, different allocations of processor utilization can greatly affect power consumption and energy efficiency. For example, the CPU utilization sum of 4 VMs is 100%. According to Fig. 1.1, the energy consumption

of 4 VMs allocated on 4 different servers is about 800 watt. It can reduce the energy consumption down to 300 watt if all of the VMs put on the same server.

However, traditional network routing algorithms, like Open Shortest Path First (OSPF) [15] and Routing Information Protocol (RIP) [16], provide static routing choices so that they are lack of flexibility to adjust flow paths for different network statuses. An energy efficient and low delay VM placement can find some bottleneck as long as they have the same source and destination nodes [17]. Traditional routing algorithms barriers the capacity utilization so that it cannot reach the optimization network. In addition, the cost of maintaining those delicate hardware is high.

Software Defined Network (SDN), as shown in Fig. 1.2, can establish flexible and programmable network by separating the control plane and the data plane [18]. OpenFlow is the protocol that implements the idea of SDN. Networks can be decomposed into a controller (a powerful network manager processing all the information of flows) and OpenFlow switches (with basic functions like receiving, lookup table, and forwarding). Using OpenFlow protocols, routing is no longer confined in an IP address or a MAC address. Controller can determine a path based on low delay, low packet loss or high security, and flow space, coarsened and fine-grained for different applications.

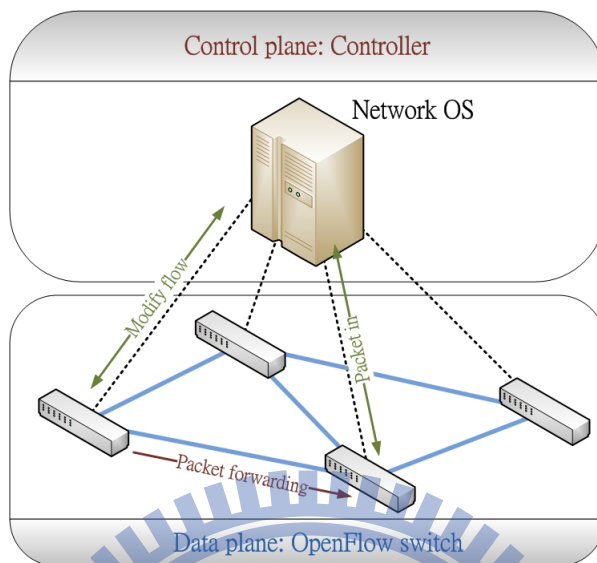


Figure 1.2: Software defined network.

1.2 Problem and Solution

Traditional VM placement techniques pay attention to the efficiency of resource allocation. When the placements apply to datacenters, it may happen unexpected congestion and degradation in the network. Regardless of the impact of network performance, VM placement can jeopardize the efficiency and utilization in the datacenter. Therefore, we design the energy efficient and QoS guarantee VM placement algorithm. Input to such a problem includes the resource demands of VMs, the traffic matrix among VMs and the cost matrix among host machines. The output to the our proposal with an OpenFlow controller dictates where VMs should be placed in order to save energy and guarantee QoS. In chapter 3, we propose a three-tier algorithm to solve the problem combining energy efficiency and QoS guarantee. It first partitions VMs to reduce traffic transmission across the entire datacenter. Then, it decides the minimum number of server without service-level agree-

ment (SLA) violation. Last, the OpenFlow controller assigns paths to avoid congestion and balance the network load balanced. The experimental results show that the proposed algorithm can significantly save energy and guarantee quality of service in comparison with other existing VM placement policies. We propose an evaluation score to assess VM placement in terms of energy, delay and throughput. Comparing to other existing placement policies, Our proposed mechanism can enhance system throughput by 25% and have better evaluation score [1].

1.3 Thesis Outline

The rest of this thesis is organized as follows. Chapter 2 introduces some VM placement policies and the background of our proposed approach. Then, we discuss system models of energy efficiency and QoS guarantee VM placement in Chapter 3. Subsequently, we explain detail of each module of our system model from Chapter 4 to Chapter 6. Experimental results are shown in Chapter 7. Finally, we conclude this thesis in Chapter 8.

Chapter 2

Background

2.1 Energy-aware VM placement

Most VM placement techniques focus on energy saving, in other words, minimizing the number of servers. To avoid SLA violation, VM placement needs to consider multi-resources of VM demands. CPU and memory are the two major considerations. Nowadays, due to the rising concerns on datacenter energy and emerging bandwidth intensive applications [19], both power consumption and bandwidth requirement are indeed taken into account when computing the placement. Therefore, some research started paying attention to finding the optimal VM placement for VMs with multiple resource demands. A research [20] mentions that VM demands for certain resources are highly bursty, so they can be modeled as stochastic processes. In datacenter networks, bandwidth demands can be approximated by the normal distribution. They propose an algorithm to solve the traditional bin packing problem with multiple deterministic and stochastic resources.

On the other hand, some approaches notice that energy consumption of network components is a vital issue in datacenter. Energy-aware VM Place-

ment [1] presents a VM placement considering the balance between server energy consumption and datacenter network energy consumption. Their goal is to reduce energy consumption in datacenters by meeting the conditions of both server-side constrains and network data transmission constrains. Due to these two conflicting objects, it applies fuzzy logic to obtain the most feasible solution. However, without topology architecture information, they mathematically derive the solution based on sufficient resource demands. Unpredictable congestion in the real data transmission may happen because they neglect routing issue in datacenter networks.

2.2 Delay-aware VM placement

In the datacenter, data-intensive applications are increasing, and often need to communicate with related data frequently. Therefore, the traffic loads among those VMs are especially heavy. The network I/O performance of the VMs can affect the performance of the applications significantly. However, the network aspects are largely ignored. This might make a VM that executes an application be placed on physical machines far away from other VMs storing the related data. It will increase system overhead and eventually the network performance deteriorates. Moreover, these energy efficiency placements seek to consolidate VMs for resource consumption saving, which can greatly impact network performance. This can lead to situations in which VM pairs with heavy traffic are placed on host machines with large network cost.

Traffic-aware VM Placement Problem [12] was proposed to solve the optimization problem based on different datacenter architectures and traffic patterns. It presents an algorithm to allocate VMs and hosts into groups,

then match them in the principles: (1) VMs pairs with heavy mutual traffic should be assigned to hosts with low-cost connections and (2) VMs with high mutual traffic should be in the same group. Although [12] illustrates the importance of network performance, energy consumption issue was not considered.

2.3 Dynamic routing algorithm

Most of the layer-3 routing algorithms are static based on the Internet Protocol (IP) and have been widely used in wide area network (WAN). Many layer-3 routing algorithms, such as RIP [16], OSPF [15] and Equal-Cost Multi-Path routing (ECMP) [21], have been proposed. Among which, RIP and OSPF are single-path routing which have been known to suffer from poor system throughputs. Moreover, computation capacity of OSPF routing on each individual node may degrade greatly when the number of the nodes increases. On the other side, ECMP is a multi-path routing and intends to effectively utilize the bandwidth of all links. ECMP takes turns to use each link for transmission so that it can result in balanced load and better network performance. However, the out-of-order problems of receiving packets cannot be avoided and may incur more cost on the system.

While D²ENDIST [17] is proposed to provide disjoint routing paths and served as a dynamic-routing mechanism. One of the ideas was originated from ENDIST [22], which provides multiple selections from numbers of divided edge nodes. It may cause overlapping paths in a symmetric datacenter network topology. Disjoint ENDIST, an improved version of ENDIST, is built upon a spanning tree algorithm that divides weighted edge nodes. In the proposed method, all of the routing paths are totally disjointed. The

other idea comes from the dynamic mechanism. Since the traffic pattern is time-invariant and under-determined, applying disjoint ENDIST can lower utilization of links. Thus, disjoint and dynamic ENDIST (D²ENDIST) is developed to eliminate the load unbalancing during data transmission in datacenter.

2.4 Literature Survey

Table 2.1 shows the difference between numerical related works and our work (EQVMP). Max-Min Multidimensional Stochastic Bin Packing (M³SBP) and Energy-aware VM Placement (EVMP) pay more attention to energy issue. Traffic-aware VM Placement (TVMP) and Network-aware VM Placement (TVMP) are designed to minimize transmission delay in datacenter networks. Walk in Line [22] is proposed to reserve bandwidth for VM migration.

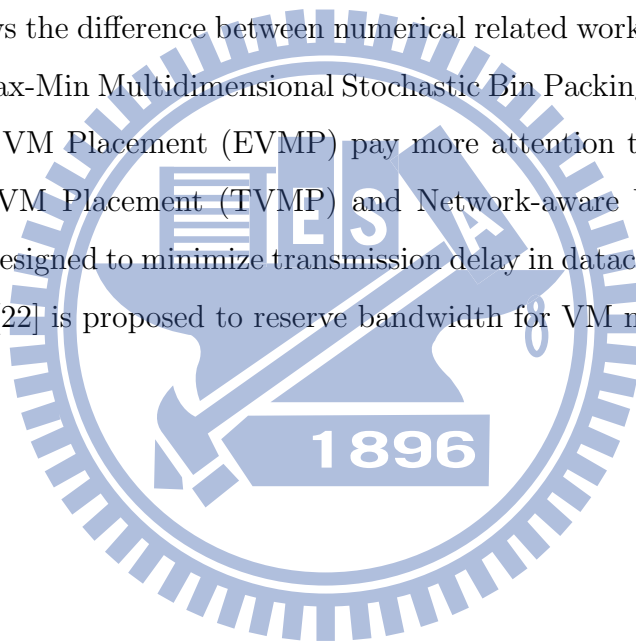


Table 2.1: Comparison of related VM placement policies

	Energy	Delay	Throughput	Feature
M ³ SBP [20]	o	x	x	Multi-resource bin-packing and stochastic analysis
EVMP [1]	o	x	x	Energy-saving on both servers and network devices
TVMP [12]	x	o	x	The impact of network due to traffic pattern and topology
NVMP [13]	x	o	x	Minimizing data transmission time among VMs
Walk in Line [22]	x	x	o	VM migration sequence order
EQVMP	o	o	o	Combination of energy and QoS issue

Chapter 3

System Model and Problem Formulation

3.1 System model

Datacenters not only provide a flexible and reliable storage space but also support underlying virtualization infrastructure. In our scenario, VMs are created and removed when users run applications. After a long period of time, network performance can degrade dramatically because the resource utilization and network traffic are unbalanced. To improve network performance, VMs should be relocated on the appropriate hosts. A snapshot records information about the VM resource demands (CPU consumption, memory usage, and bandwidth requirement) and VM traffic. We also record the topology in matrix form.

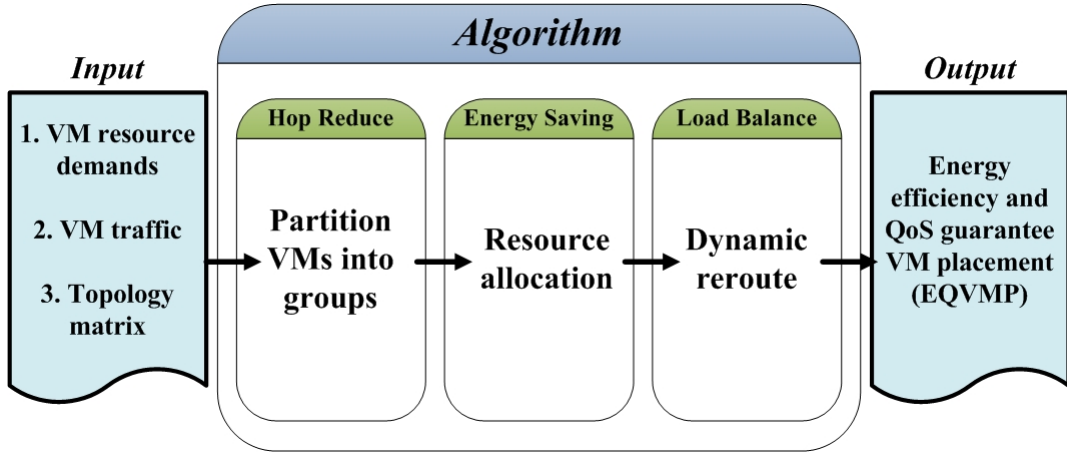


Figure 3.1: Energy-efficient with QoS-guarantee VM Placement algorithm in 3 phases.

Our system model is shown in Fig. 3.1. We take VM resource demands, VM traffic and topology matrix as input. First, we divide VMs into groups in terms of hop count reduction. Formally, datacenter architectures are usually multi-tier and symmetric, which can easily represent in matrix form, and we assume every switch causes equal delay. In the VM partitioning stage, servers is separated into different clusters. Basically, The data transmission among hosts in the same cluster traverses one hop only. However, the network cost will be higher when data transmits between the clusters. According to above assumption, we consider a datacenter network with hop count matrix H , where each element h_{ij} represents traversal hop number from host i to j in fig. 3.2.

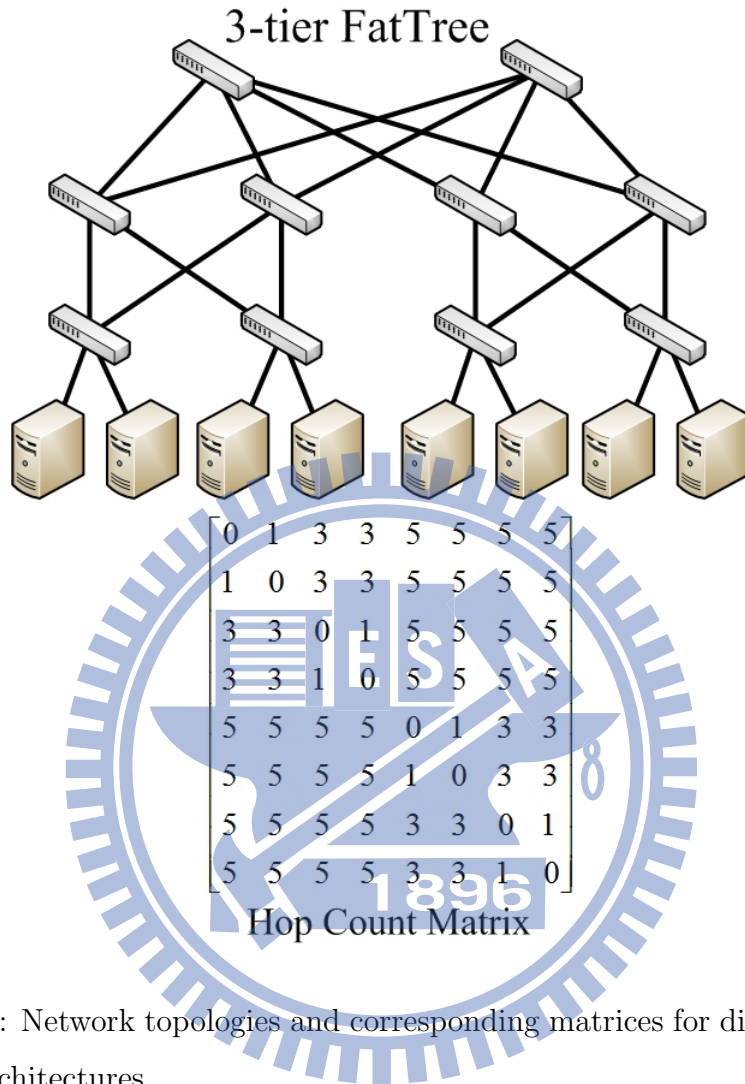


Figure 3.2: Network topologies and corresponding matrices for different datacenter architectures.

After deciding the groups of VMs, energy saving process can minimize the number of power-on servers. To deploy massive VM placement, we consider a scenario in which there are n VMs with m kinds of resources. Based on homogeneous architecture of modern datacenter, the hosts are assumed to have identical capacities. Let $V = \{v_1, v_2, \dots, v_n\}$ denotes set of VMs. The traffic load matrix L records flows between VMs and l_{ij} represents the flow size from v_i to v_j . Then CPU consumption c_i , memory usage m_i and

bandwidth demand b_i are corresponding to resource demand of v_i , which can denote $D_{v_i} = (c_i, m_i, b_i)$. Here b_i is determined by total transmission amount of v_i . For the sake of hosting VM i without violation, host h has to meet all of its resource demands D_{v_i} .

After determining VMs to corresponding servers, we solve the problem of energy consumption and routing path in the static deployment. However, we need to consider the situation that the networks occur bottleneck when data transmission starts. Network controller will monitor all the utilization status of each link. If any bottleneck is detected, it will decide an alternative path to achieve load balance. Equally allocating traffic flows to the network can reduce the probability of congestion. After a period of time, we will update VM placement based on current VM status and repeat the process above.

3.2 Problem Formulation

In datacenter networks, both of how to save energy and how to maintain QoS are crucial issues. Owing to the development of virtualization and virtual machine migration, the energy usage has become more efficient and effective. Since the requests and the applications to datacenter grow rapidly because some research [12] illustrate that the amount of services with massive bandwidth demands and strict latency constraints become huge. For these reasons, VM placement is no longer a simple problem to save energy; therefore we have to consider the traffic among VMs to prevent the congestion happening due to aggressive VM placement.

Datacenter studies always discuss the problem of resource allocation and networking issue individually. The problem of calculating the minimum number of power-on server and reducing total network delay, meanwhile main-

taining QoS, is an important issue. In this work, we consider a three-tier fat-tree topology network with VMs added and removed as real scenario in datacenter networks and propose a combination of energy efficiency and QoS guarantee mechanism to solve the aforementioned problem.



Chapter 4

Hop Reduction

4.1 Route Reduction

A network topology can be mapped into a graph with vertices and edges. The goal of routing protocol is to compute the path with the lowest cost or distance, so it becomes the shortest path problem for a graph with non-negative edge path costs. Dijkstra's algorithm is applied in routing to provide the shortest path, especially when graphs are irregular and asymmetric. Some well-known shortest path routing protocol like Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (IS-IS) are widely used in network.

Datacenters follow the multi-tier and symmetric network architecture. Each server connects to one edge switch at the bottom level. Each edge switch connects to multiple switches at the aggregation tier. Each aggregation switch is also connected with multiple switches at the core tier. In Fig. 4.1, there are edge level (1 hop), aggregation level (3 hops) and core level (5 hops). We can easily obtain routing distance by the hop count matrix H , where each element h_{ij} represents traversal hop number from host i to j .

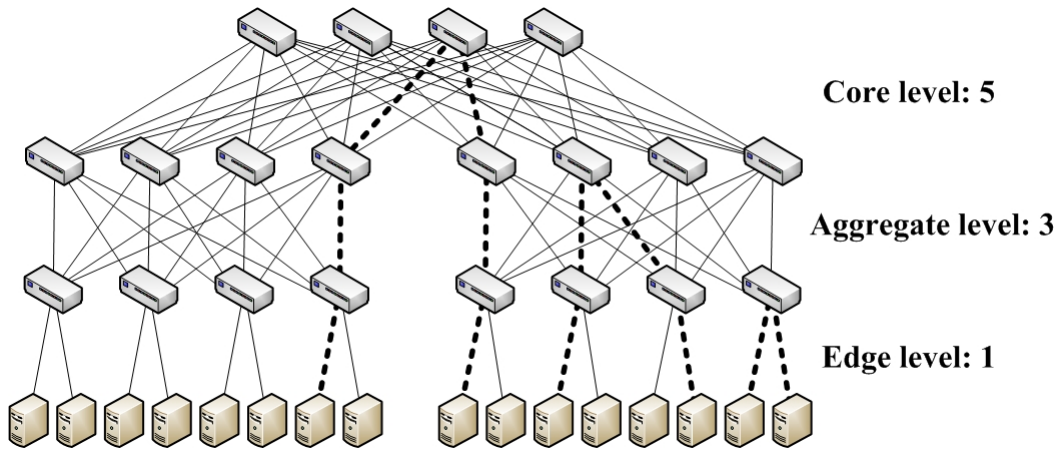


Figure 4.1: Hop count expression in datacenter topology.

In our scenario, we assume switches cause equal delay. Therefore, the hop number of routing path plays a great role in delay. We notice that delay is determined by the distance between servers where VMs are hosted. The other advantage of reducing routes is to lower the probability of data transmission across the whole datacenter. The longer time data forwards in networks, the greater chance it happens congestion. It is important to put VMs with heavy traffic in the same group.

4.2 Graph Partition

We have the information of traffic among VMs, represented in traffic load matrix L . Each element l_{ij} shows the traffic amount between v_i and v_j . Inspired by Cluster-and-Cut [12], we can transform the matrix into a graph. Vertices are VMs and edges are the traffic among VMs, which can be modeled as graph partitioning problem. Our goal is to divide VMs into 2 groups with low mutual traffic loads among them. If we have to divide into more than 2 groups, multilevel bisection partition will be applied.

Graph partition problem is defined on data represented in the form of a graph G (vertices and edges) such that it is possible to partition a graph into smaller components with dividing policies or specific properties. For instance, a k -way partition divides the vertex set into k smaller components. A good partition is defined as one in which the number of edges running between separated components is small. Uniform graph partition is a type of graph partitioning problem consisting of dividing a graph into components, such that the components are of about the same size and there are few connections between the components.

In the studies of graph partitioning [23], it introduces many partitioning method, such as random matching, heavy edge matching and light vertex matching. However, none of them are appropriate for our work due to high computational time and massive connections between partitioning components. Metis [24] is a partition algorithms based on the multilevel graph partitioning paradigm. It has been shown to quickly produce high-quality partitionings and fill-reducing orderings. In addition to traditional partitioning objective, i.e., the number of edge and communication field, it also provides alternate partitioning objectives depending on the following factors: (i) the total communication volume; (ii) the maximum amount of data that any particular processor needs to send and receive; and (iii) the number of messages a processor needs to send and receive.

4.3 Proposed Module

In our scenario, the VMs traffic information is retrieved from the snapshot. Based on the collected traffic loads and the datacenter topology architecture, hop reduce mechanism is proposed to partition VMs into groups that the

number of VMs in each group is balanced and the costs between different groups are minimized. Fig. 4.2 indicates the traffic load among VMs. We randomly arrange VMs into two groups $\{1,2,4\}$ and $\{3,5,6\}$. Although the number of both groups are equal, the traffic load sum between groups is 20 so that heavy traffic load can cause great delay across a datacenter. The unequal partitioning groups $\{1\}$ and $\{2,3,4,5,6\}$ has the minimum mutual traffic load sum but the unbalanced division of VMs may lead to congestion in specific area of the network. The best partition is $\{1,2,3\}$ and $\{4,5,6\}$, which satisfies balanced partitioning and low cost. In our topology, Fig. 4.3, we can bisection VMs to reduce hop count. However, the partition time will be different in other topology such as VL2, BCube and PortLand.

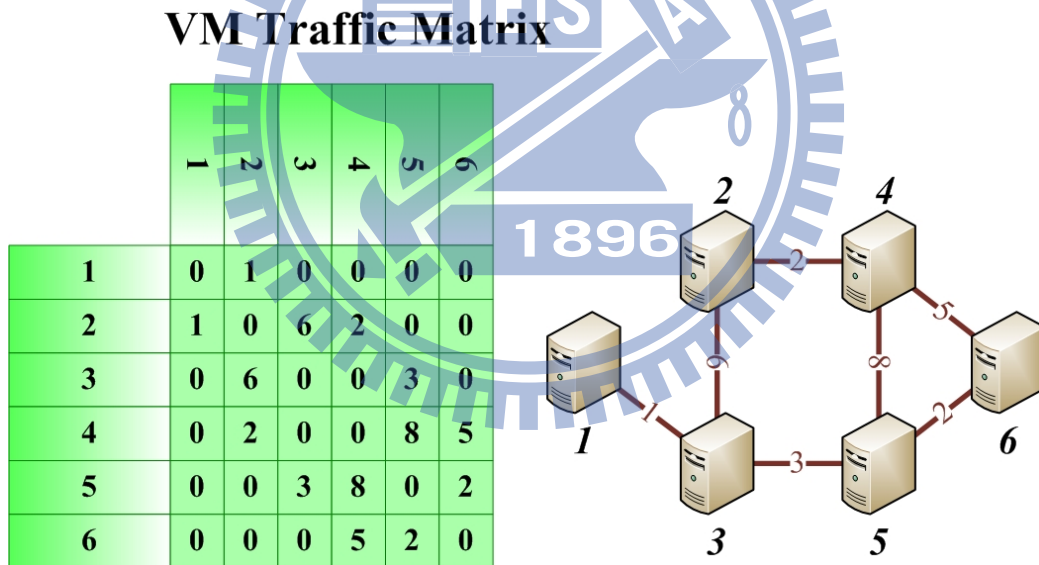


Figure 4.2: VM partitioning with traffic matrix and graph.

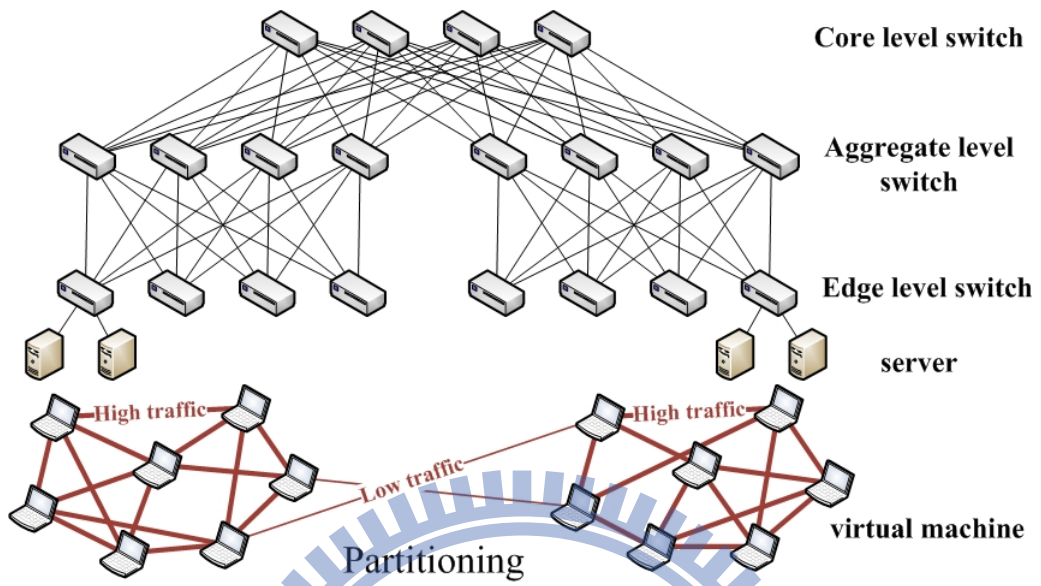
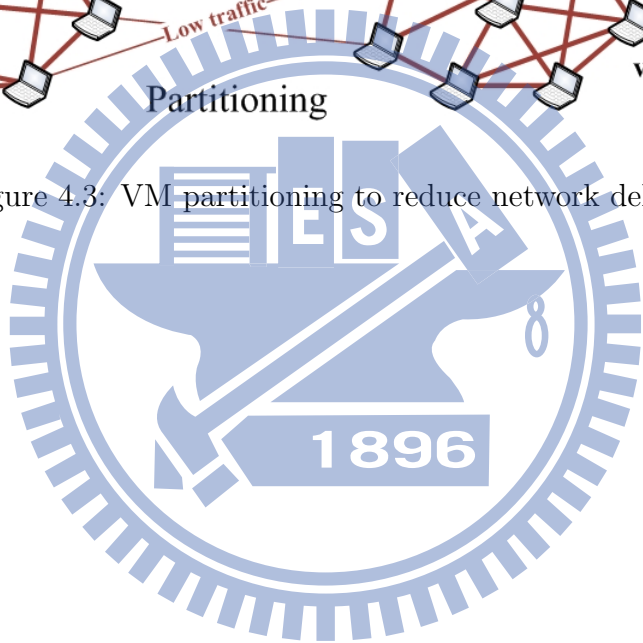


Figure 4.3: VM partitioning to reduce network delay.



Chapter 5

Energy Saving

5.1 Server vs Network Devices

A traditional datacenter architecture with 32,000 servers will consume about 8 million watts on servers at peak load. With the development of multi-root topology, some new datacenter architectures are proposed on this basis such as VL2 [25], PortLand [26] and BCube [27]. Those high-performance datacenter networks we mentioned above still have to consume 12% of overall power at full utilization. Ideally, any idle switch would consume no power, and energy consumption grows only with increasing network load. However, current network devices are not energy proportional [7] because of fixed overheads such as fans, switching fabric, and line-cards. Those components waste energy at low network loads. Most of the time, servers operate at lower levels rather than full-utilization; therefore energy proportional of the network power cannot be ignored. In Fig. 5.1, if the network system is 15% utilized and not fully energy-proportional, the network components will consume nearly 50% of overall power. At this time, energy proportional design can at least have 85% margin of power consumption to save.

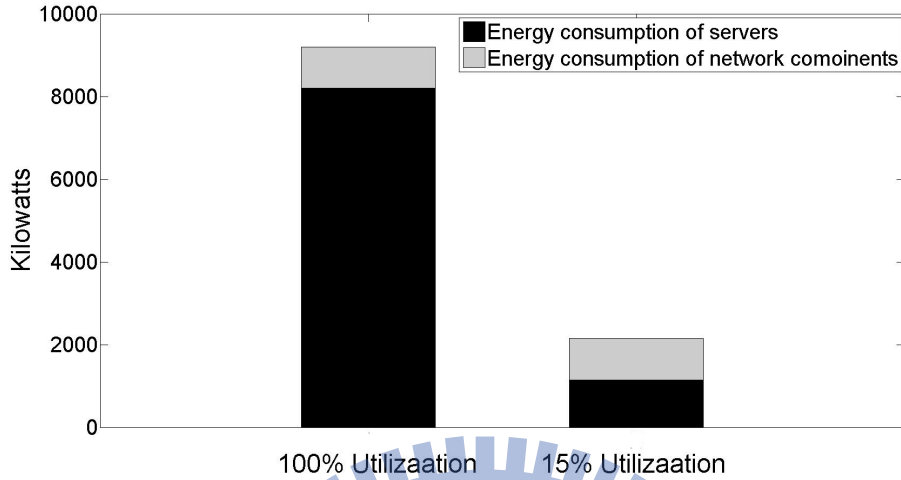


Figure 5.1: VM placement before energy saving.

Since energy consumption of network devices should be proportional, related approaches are proposed to minimize the number of powered on switches and maintain the network when switches are turned off [4]. An algorithm combines energy and routing issue. First, the route generation module selects the routing path for each flow so that the network throughput can be maximized. Then, the throughput computation module calculates the network throughput with the current topology. After that, the switch elimination module is responsible for selecting the switches which can be eliminated from the network. When some switches are shut down, routing paths should update due to the changes of the traffic matrix. The recursive process will not stop until the throughput reaches the minimum threshold.

Basically, shutting down switches could be a solution to save extra energy out of servers. However, it will face several serious problems. When a sudden burst traffic pours in the network, the mechanism can not power on switches immediately not only because the recursive algorithm computation takes time but also switches configuration needs time. Another reason is that

OpenFlow switches are important components to gather information from the network and connect to the controller. The other reason we mentioned above, energy consumption of network devices accounts to only about 15% of the full utilization datacenter energy usage. In our work, we concern both about network performance and energy efficiency, but reducing the number of switch is not an option.

5.2 energy efficiency algorithm

To reach the goal of energy efficiency, we have to minimize the number of required servers while satisfying the SLA availability guarantee. Obviously, it can be modeled as bin packing problem and it is a combinatorial NP-hard problem in computational complexity theory. In the problem description, the objects of different volumes must be packed into a finite number of bins or containers in a way that minimizes the number of bins used. The simplest approximate approach to the bin packing problem is Next Fit (NF) algorithm. The first item is assigned to bin 1 and then item 2 to item n are considered by increasing indices. Each item is assigned to the current bin if it fits; otherwise, it is assigned to a new bin, which becomes the current one. A better algorithm, First-Fit (FF), considers the items according to increasing indices and assigns each item to the lowest indexed initialized bin into which it fits. A new bin is introduced only when the current item cannot fit into any initialized bin. The other algorithm, Best-Fit (BF), is obtained from FF by assigned the current item to the feasible bin having the smallest residual capacity, which breaks the rule of choosing the lowest indexed bin. Finally, the performance improve even better when we sort the items in decreasing order, which is called Best-Fit Decreasing (BFD).

When the problem becomes more complicated such as multi-resource bin packing problem, there are many solution based on different scenarios or fitting principles. Dominant Resource First (DRF) solves the fair resource allocation problem, where bins with multiple resources are shared by different users. The dominant share of user is defined as the maximum share that the user has been allocated of any resource. DRF seeks to maximize the minimum dominant share across all users. While Max-Min Multidimensional Stochastic Bin Packing (M³SBP) solves the bin packing problem with stochastic constrains. M³SBP tackles the multi-resource allocation problem, where it indicates that some resource demands may be modeled as stochastic process. M³SBP seeks for the optimal VM to place on specific server with minimum remaining resources. Both DRF and M³SBP yield higher server utilizations and fewer servers than other naive bin packing algorithms do.

5.3 Proposed Module

Our energy saving techniques search for an energy-efficiency multi-resource placement to guarantee that each VM can meet its requirements. The placement is inspired by Best Fit Decreasing (BFD) and Max-Min Multidimensional Stochastic Bin Packing (M³SBP). BFD solves the classical bin packing problem. The process of sorting items in the decreasing order by their size determines that larger items have higher priorities in packing orders. M³SBP provides the solution for multi-resource allocation problem. It seeks for the optimal VM to place on specific server with minimum remaining resources.

Hop reduction divides VMs into groups and reduces the traffic load among groups by graph partitioning. It can localize large chunks of traffic and thus reduce load at high-level switches. Few traffic across the datacenter greatly

lower the data transmission time and average delay time. In Fig. 5.3, v_1 to v_k are on the left and the rest are on the right. Yet it is not an energy efficient VM placement, the random placement may lead to the situations that some servers can not meet the resource demands of VMs. Moreover, this placement will cause serious SLA violation. In energy saving module, we only focus on the minimum resource utilization because hop reduction has greatly lowered average delay by clustering VMs.

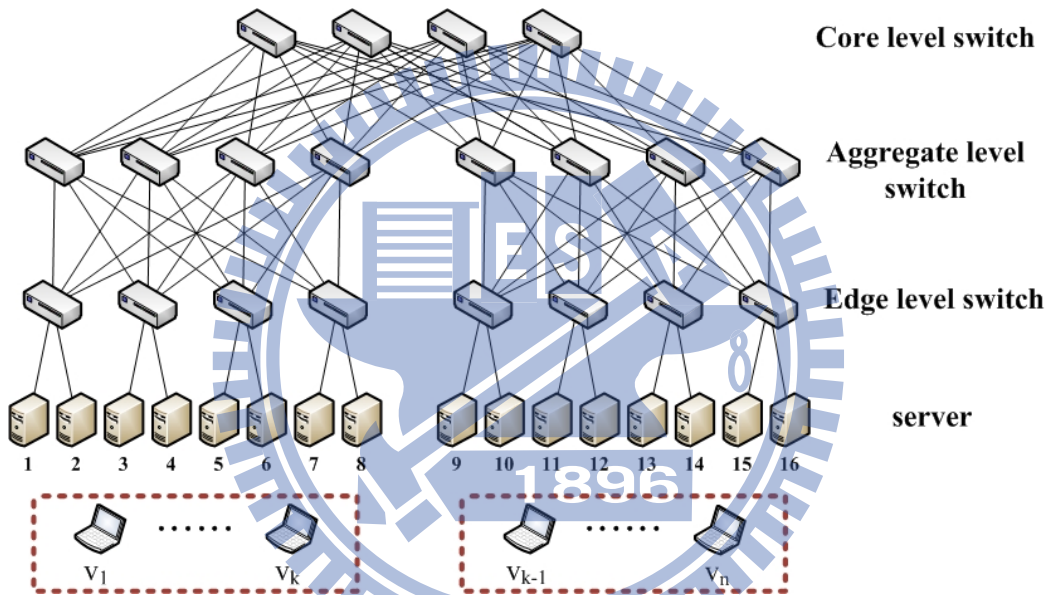


Figure 5.2: VM placement before energy saving.

n VMs are placed into m servers by considering their resource demands (CPU consumptions, memory usages and bandwidth demands). The basic idea of our energy module is as follows. First, we sort VMs in the decreasing order by the summation of their resource demands. For each newly powered-on server (current server in short), we choose a set of candidate VMs that the current server can fit each of them. Then, we select the candidate VM which can be place on the current server with the minimum resource left. If

there is no candidate VM in the set, it illustrates that none of existing VMs can be hosted on the current server. Another server will be powered on to run in iteration rounds. Energy saving module can decide the minimum number of server in the datacenter as shown in Fig. 5.3. Before moving to the next module, load balance, we need to modify the original traffic matrix L between VMs into another matrix, called physical machine (PM) traffic matrix. Let P denote the total traffic and its element defines from PM_i to PM_j . For the record, energy saving module provides sufficient bandwidth on the port of servers. However, we cannot guarantee sufficient bandwidth in datacenter networks so far.

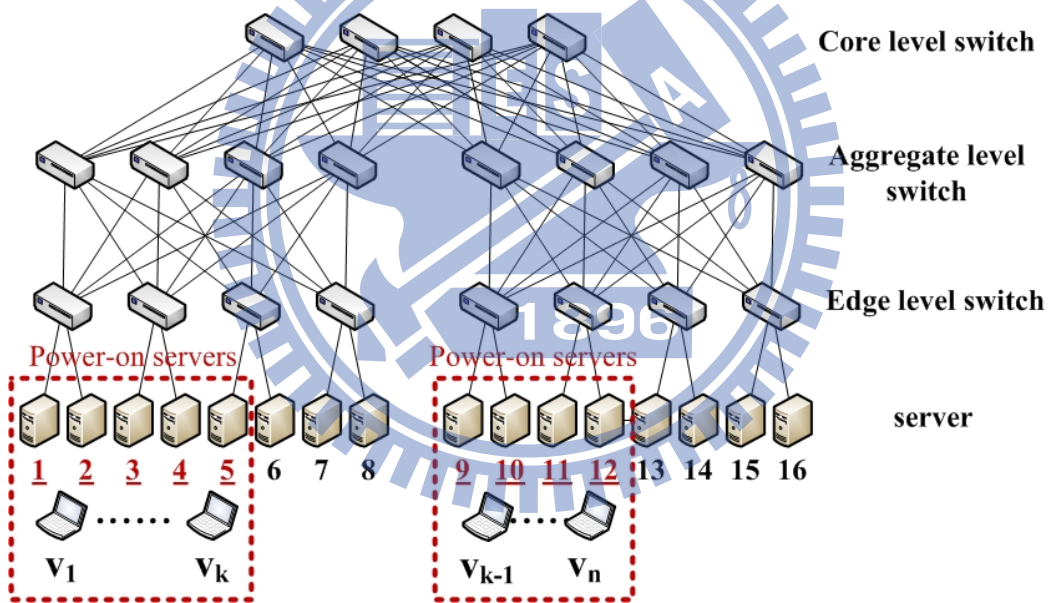


Figure 5.3: VM placement after energy saving.

Chapter 6

Load Balancing

6.1 Network Management

The Internet server programs support the mission-critical applications such as financial transactions, database access, corporate intranets, and other key functions which must run 24 hours a day, seven days a week. Networks need the ability to scale performance to handle large volumes of client requests without creating unwanted delays. For these reasons, clustering technique is of wide interest to the enterprise. Clustering enables a group of independent servers to be managed as a single system for higher availability, easier manageability, and greater scalability. Therefore, network load balancing provides scalability and high availability to enterprise-wide TCP/IP services, such as Web, terminal services, proxy, virtual private networking (VPN) and streaming media services. network load balancing brings special value to enterprises deploying TCP/IP services, such as e-commerce applications, that links clients with transaction applications and back-end databases.

Network load balancing (commonly referred to as dual-WAN routing or multihoming) is the ability to balance traffic across two WAN links without

using complex routing protocols like Border Gateway Protocol (BGP). This capability balances network sessions like Web, email, video streaming, audio streaming and file transmission. In order to spread out the amount of bandwidth used by each LAN user, it increases the total amount of bandwidth available over multiple connections. For example, a user has a single WAN connection to the Internet operating at 1.5Mbit/s. They wish to add a second broadband (cable, DSL, wireless, etc.) connection operating at 2.5Mbit/s. This would provide them with a total of 4Mbit/s of bandwidth when balancing sessions.

In cloud datacenters, applications with massive bandwidth demands and strict latency constraints grow rapidly recent years. In order to meet the requirements of different of application, cloud datacenter networks are equipped with centralized mechanisms, called network management system (NMS), to adjust the networking related components to achieve specific network performance. In software defined datacenter networks, the controller integrates all of the tasks from NMS including failure recovery, traffic information collection, and utilization detection. However, the controller can do even more to modify flow header so that flows with the same source and destination can forward to different paths. Moreover, it can also allocate bandwidth by slicing the network so it is able to guarantee application QoS.

6.2 Flow Routing

Traditional routing mechanisms provide single route path on the same source and destination due to the fixed algorithm written in the network devices. Therefore, it can not compute fine-grained routing path based on the network status. After applying energy saving, nearly all of the power-on servers are

in full utilization. In other words, VMs take up all the bandwidth resource on some specific path. In the traditional static routing, there are numbers of data transmission on the same path so that it definitely occurs bottleneck at the lower switch level. Besides, the traditional routing algorithm is a distributed system. In such huge scale of datacenter networks, routing tables are updated dynamically by obtaining the network information from other routers. Routers in the network must constantly update the information of the changes in the topology. Routers may be added or removed, or routers may be out of function due to failures in the physical links. This situation may lead to failure of routing path decision. Flow routing can provide a way to route in alternative path when failures and congestions happen unexpectedly.

In traditional networks, the concept of flow routing has been proposed already. Edge Node Divided Spanning Tree (ENDIST) [22] is also a shortest-path routing algorithm. It divides edge nodes into sub-nodes, which helps to assign MAC address for each sub-node. ENDIST avoids the discipline of single path in spanning tree protocol by adopting flow-basis selection. Hash-Based Routing (HBR) [28] is another method applying flow routing. It constructs a routing path hash table as MAC address. The advantage of HBR is fast table look-up mechanism. HBR defines flows by the port they are received and decides the corresponding output ports for routing. Many different strategies, such as Round Robin, can be incorporated with HBR. It is applied onto a two-stage and fully-mesh topology and thus the flow can be transmitted effectively. A new routing algorithm, Dynamic & Disjoint ENDIST-based (D^2 ENDIST), is proposed to support various types of multi-layer scale network topologies, dynamic adjustment of traffic imbalance and fast recovery from link failure and VM migration. It consists of two main

stages: (1) routing by disjoint ENDIST and (2) rerouting with dynamic reweights.

6.3 Proposed Module

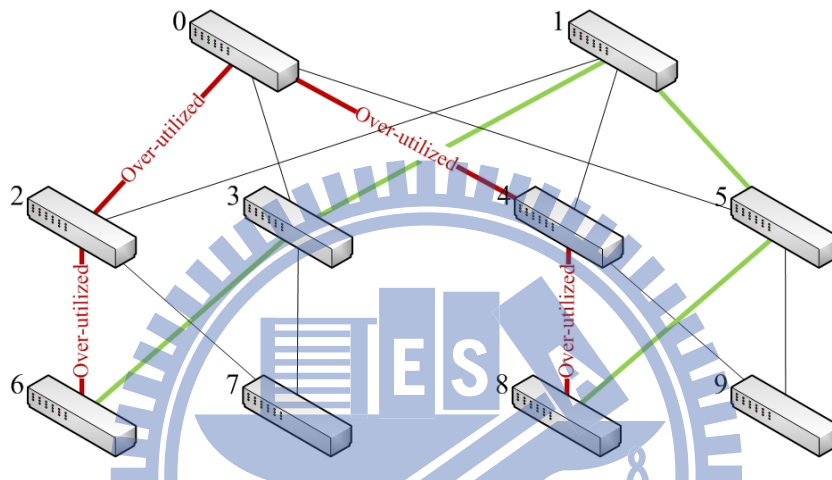
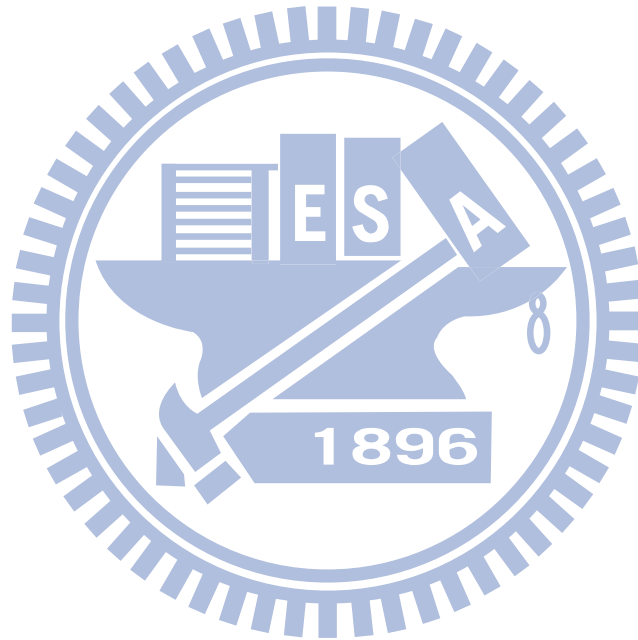


Figure 6.1: VM placement with load balance mechanism.

Fig. 6.1 is an illustrative example that load balancing mechanism can improve network performance greatly. After the process of hop reduction and energy saving, we assume that VMs are hosted under switch 6 and switch 8. With traditional static routing algorithm, we notice that all of the traffic flows follow the path 6-2-0-4-8, which happens congestion. The purpose of load balancing is to detect the over-utilized links so that it can decide alternative path 6-3-1-5-8 to ease congestion. It helps to reach a balanced network.

Now, we have decided which host VMs should be put with the constraints of energy efficiency and hop reduction. Owing to the advantage of SDN, load balancing attempts to achieve flow transmission in networks without congestion. In SDN datacenters, the controller can assign flows to different

routing paths although they have the same source and destination. The controller monitors the utilization of every link in datacenter networks. Once controller detects that a link reaches to the threshold, like 90 % of the maximum capacity, it will immediately assign another low utilization path and move certain portion of flows on it to balance the traffic. When VMs are randomly added or removed with time, we will periodically compute the new placement to maintain the network in the constraints of energy efficiency and delay reduction. Then, load balancing repeats again.



Chapter 7

Experimental Results

Through our proposed system model, we can notice that there are two parts of our experiment. One is to determine how to arrange VMs on servers based on energy efficiency and hop reduction. The other is to put our VM placement on the simulation tool to observe network performance. In the former part, we apply Java [29] programming to compute our VM placement setting. In the later part, we use NS2 [30] as our simulation tool.

Although there exist some simulation tools, such as Mininet [31], to create OpenFlow network environment and emulate the behavior of the controller managing the flows. However, Mininet is unable to quantify the traffic and the bandwidth information so that it is impossible to evaluate network performance. It is originally designed to perform how the controller modifies flow headers to manage OpenFlow networks. On the contrary, NS2 provides source routing which can designate the routing path of each flow like OpenFlow controller does. That is why NS2 is adopted in our work.

In our experiment, given that topology architecture is a 3-tier fat-tree datacenter network, consisting 16 core-level, 32 aggregate-level and 32 edge-level switches. Each edge-level switch can connect 8 servers, and each server

can host 4 VMs. We assume 256 VMs that power consumptions, memory usages and bandwidth demands are given from the network snapshot. The bandwidth demands of VMs meet the uniform distribution from 0 to 100% utilization of the link capacity (10 Mb). Power consumptions and memory usages also meet uniform distribution $U(0, 100)$, which are represented in the utilization percentage of a server. When we run the simulation in 10000 VMs, it occurs the problem of insufficient memory. Therefore, our model supports to the maximum size of 10000 VMs in datacenters. Some research show that communication intensive applications [12] appear more often nowadays. Therefore, a large number of FTP traffic flows are generated to represent the real situations. Besides, most of the data transmission among VMs is related and confined in certain VMs [13]. We apply the group traffic as our traffic pattern. In our experiment, the default simulation time of networks is set as 100 seconds.

7.1 system performance

In the first experiment, we implement our energy efficiency and QoS guarantee VM placement to observe objectives, throughput, delay and number of power-on server, in different phases of our system model. From Fig. 7.1, network throughput of original VM placement is the lowest, and in addition, not only all the servers are powered on but average delay is the highest. After hop reduce, Fig. 7.2 indicates that network throughput does not improve much. However, average delay greatly drops from 0.3251 to 0.086. Although delay increases because of aggressive energy efficiency placement, it is still lower than the original placement. Finally, we show that complete model with periodically reroute successfully maintains QoS no matter in throughput or

delay.

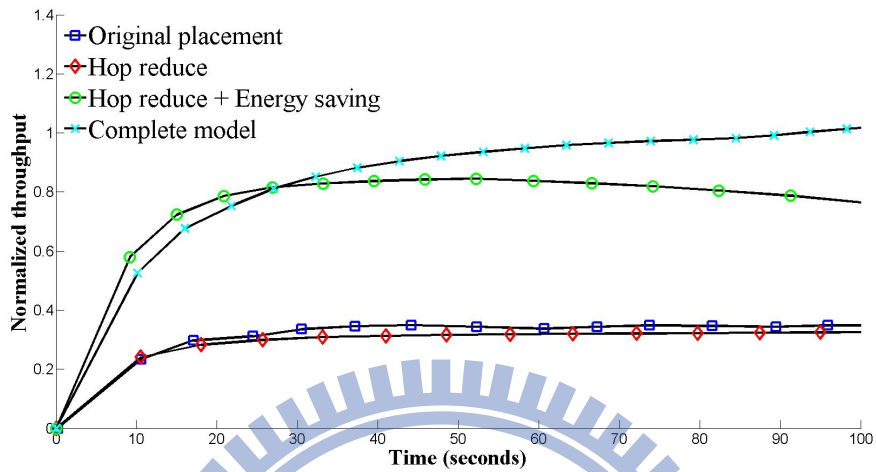


Figure 7.1: Throughput in different phases.

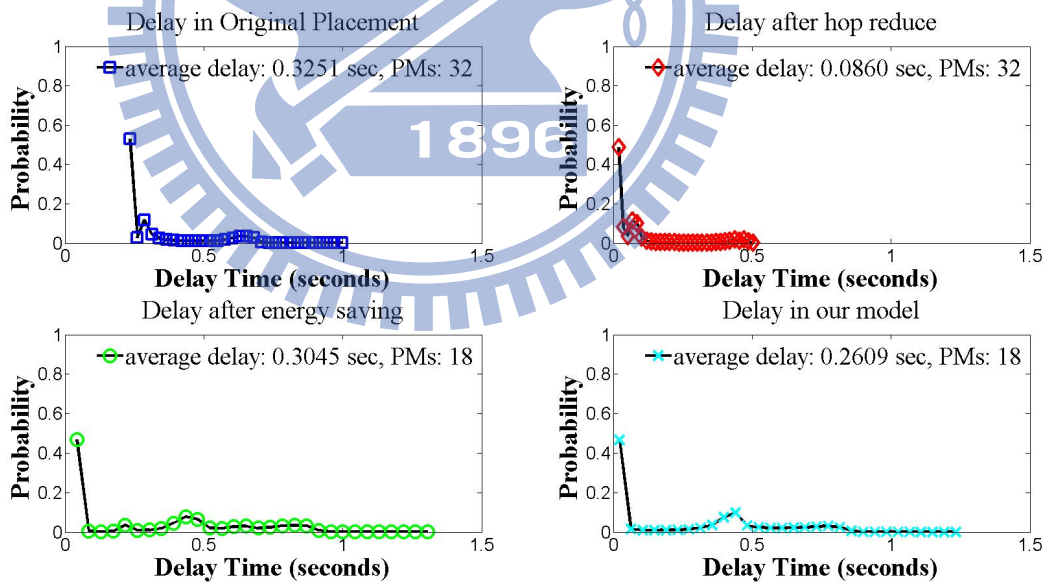


Figure 7.2: Delay in different phases.

7.2 Update period

In the second experiment, we discuss the factor of update period related to the computational time. In Fig. 7.3, computational time of VM placement is composed of METIS partitioning time and resource allocation time. Owing to computation efficiency of METIS, we can roughly estimate resource allocation time as the total computational time. Time complexity of our computation is linear time. For example, a PC with 4-core CPU needs about 0.5 hour to update a 10000 VMs placement, which decides the minimum update period. In our simulation, the number of VM is 256 and the minimum update period is 1.3 sec. As simulation starts running, we will randomly create or remove VMs to emulate the real datacenter scenario. We follow Poisson process and VM inter-arrival rate distribution is $A(t) = \lambda e^{-\lambda t}$, where $\lambda = \frac{1}{10}$ per second. The rate of removing VMs follows the same Poisson process. Fig. 7.4 shows the throughput comparison between different update period. Obviously, as long as update period is greater than the minimum update period, we obtain the fact that the network performance is better with updating more frequently.

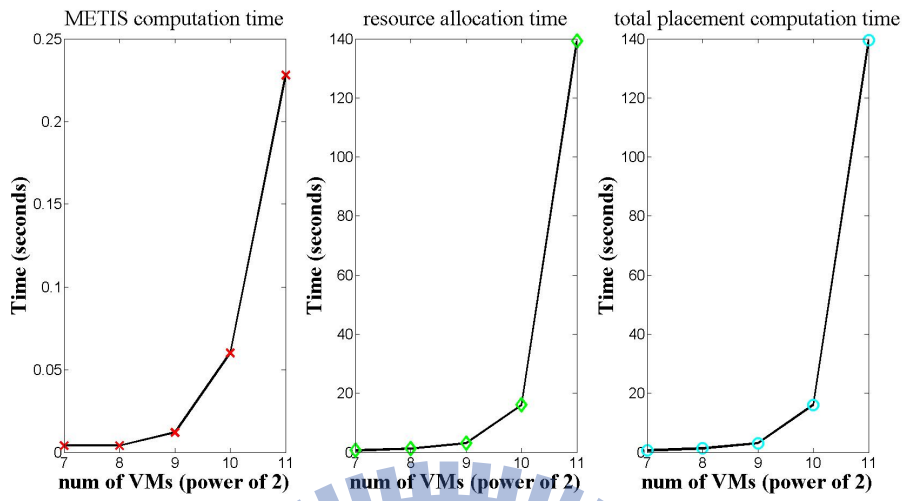


Figure 7.3: Computational time of VM placement.

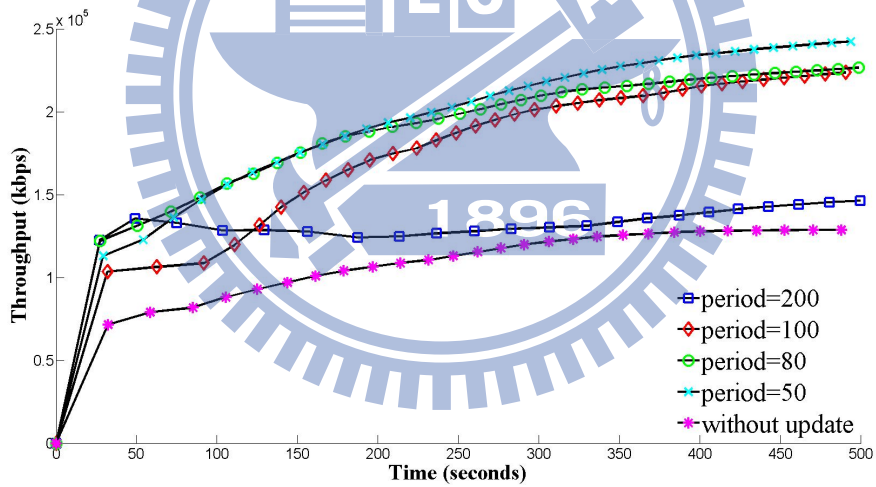


Figure 7.4: Performance with different update period.

7.3 Comparison of different placement policies

The following experiment introduces some existing VM allocation methods, such as First Come First Serve (FCFS), Largest Task First (LTF), Round Robin (RR). FCFS is to place VM in the order of their arrival time. To reach the most efficient resource utilization, LTF is to allocate VMs with heavy resource demand on the same server without SLA violation. While RR considers the fairness in networks, VMs are placed equally on servers. From Fig. 7.5, LTF has the lowest throughput because the aggressive placement with bandwidth demand causes bottleneck. Although FCFS and RR have the same throughput, FCFS is superior than RR in energy aspect. Eventually, it shows the excellent performance of our method both in energy efficiency and QoS guarantee and enhances system throughput by 25%.

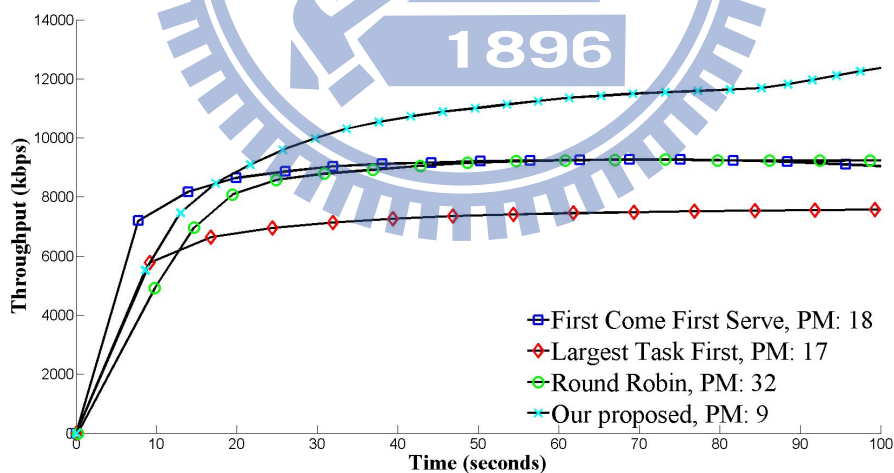


Figure 7.5: Comparison between different VM placement policies.

7.4 Evaluation score

Table 7.1: Comparison with related works

	Energy consumption (Watt)	Average delay (sec)	System throughput (kbps)
M ³ SBP [20]	5120	0.2839	4.10x10 ⁶
TVMP [12]	8640	0.1078	8.05x10 ⁶
EQVMP	5580	0.2621	1.75x10 ⁸

Since current VM placement policies focus on one objective such as energy or delay, EQVMP can both save energy and guarantee QoS. In our forth experiment, we investigate the energy consumption, average delay and system throughput in datacenter networks with different VM placement policies. Table 7.1 shows that M³SBP is superior to the minimum energy consumption. On the other hand, M³SBP overlooks the impact of networks so that it has poor performance on delay and throughput. TVMP attempts to lower the network delay by putting VMs with heavy traffic together. Obviously, TVMP has the minimum delay. In some traffic patterns, VMs can be uniformly allocated on servers causing excessive energy consumption. However, Table 7.1 illustrates that EQVMP has balanced performance on energy consumption, delay and throughput.

$$EV = \alpha \cdot \frac{E_{min}}{E} + \beta \cdot \frac{D_{min}}{D} + \gamma \cdot \frac{T}{T_{max}}. \quad (7.1)$$

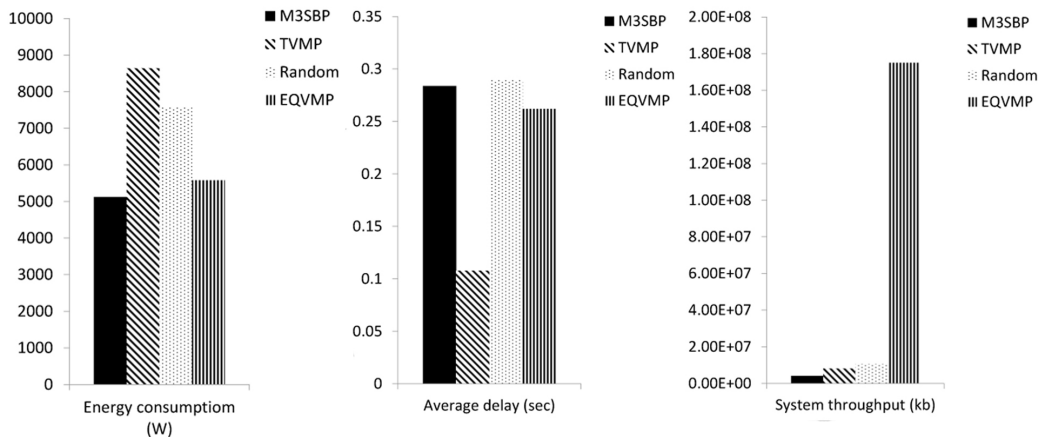


Figure 7.6: Comparison between different awareness of VM placement.

Fig. 7.6 illustrates the comparison among Max-Min Multidimensional Stochastic Bin Packing (M³SBP), Traffic-aware VM Placement (TVMP), Random VM Placement (RVMP) and our proposed placement (EQVMP) in bar charts. We observe that the energy consumption of EQVMP is close to the optimal consumption. Although the delay of EQVMP is almost twice as much as DVMP, it is still lower than EVMP and RVMP. Due to dynamic reroute and periodical update mechanism, the throughput of EQVMP overwhelms other VM placements. To provide a reliable and general evaluation score of a VM placement on the aspects of energy and QoS, we propose an equation 7.1 to determine if the placement is good or not based on the theory of linear programming [1]. We can use different weights to observe the tendency of placement policies. Given that E_{min} is the minimum power consumption of power-on servers in the M³SBP. TVMP only partitions VMs with the minimum cost so that D_{min} represents the minimum delay of it. Owing to our load balance module, we assume that all the links are fully utilized. As a result, the throughput of our system model can be regarded as the maximum system throughput. The weights of each terms, α , β and

γ , satisfy that $\alpha+\beta+\gamma=1$. Table 7.2 indicates our proposed has the best evaluation score in the balance weights ($\alpha=\beta=\gamma$). When we apply different weights (Energy-critical: $\alpha=0.5, \beta=0.25, \gamma=0.25$; Delay-critical: $\alpha=0.25, \beta=0.5, \gamma=0.25$; Throughput-critical: $\alpha=0.25, \beta=0.25, \gamma=0.5$), our evaluation score is still superior than others because of the excellent performance in throughput.

Table 7.2: Evaluation score list

Placement Name	Energy-aware	Delay-aware	Random	Our Proposed
Balance	0.4677	0.5133	0.3415	0.7667
Energy Critical	0.6008	0.5115	0.4043	0.7973
Delay Critical	0.4457	0.6365	0.3491	0.6779
Throughput Critical	0.3566	0.3980	0.2711	0.8250

Chapter 8

Conclusions

8.1 Summary

Both energy and QoS are critical issues in datacenter networks. Owing to the applications of massive bandwidth demands and strict delay constraints, how to maintain effective datacenter network condition with the minimum resources is an important issue. Many previous works proposed on VM placement policy guarantee VMs to have sufficient resources and utilize the network resources more effectively. However, they still suffer from unbalance and aggressive placement so that they will lead to severe congestion in datacenter networks. Therefore, in this thesis, we propose the energy efficiency and QoS guarantee VM placement (EQVMP) mechanism.

Experiments show that our approach can provide better system throughput than other VM placement strategies. EQVMP determines a good VM placement considering energy consumption, hop delay and network throughput. Although our energy and delay performance are the second best, EQVMP outperforms other placement schemes by achieving 10 times more throughput than the energy-aware placement and the delay-aware placement. To

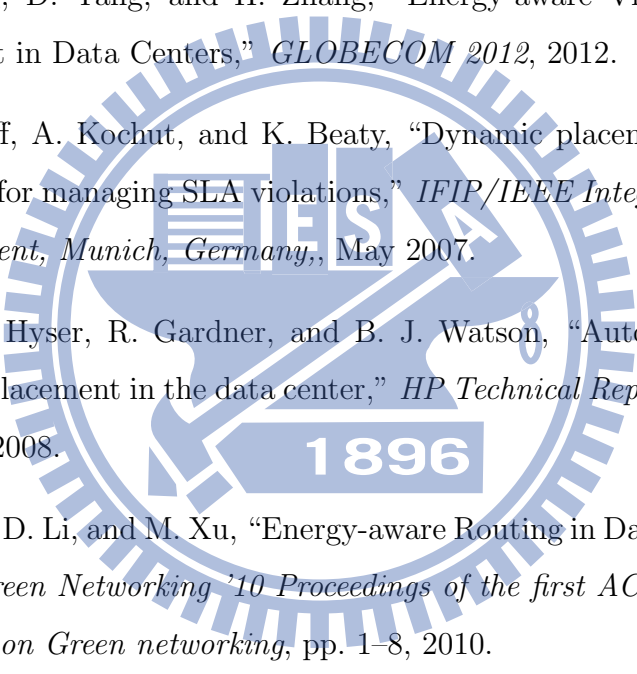
make comparisons, we propose an evaluation score to indicate the score of VM placement policies, and EQVMP is superior on every viewpoints of our considerations. Our computation time of new VM placement configuration is large. However, based on our scenario, we provide a long update period to relocate VM placement rather than adjusting them frequently.

8.2 Future Research

For the future research of the thesis, we provide the following suggestions to extend our work:

- We determine the optimal placement based on current resource demands of VMs and topology information, yet we have not consider the original placement. It is a trade-off between the optimal placement and the minimum migration distance.
- Based on the concept of evaluation value, we set different weights to the modules of our system model so that we can provide a VM placement inclined to certain characteristic.
- Application-aware is an important issue in datacenter networks. With different application characteristics, file transmission can tolerate delay but packet loss while video streaming allows packet loss with any delay. This will make VM placement more flexible and effective.

Bibliography

- 
- [1] D. Huang, D. Yang, and H. Zhang, “Energy-aware Virtual Machine Placement in Data Centers,” *GLOBECOM 2012*, 2012.
- [2] N. Bobroff, A. Kochut, and K. Beaty, “Dynamic placement of virtual machines for managing SLA violations,” *IFIP/IEEE Integrated Network Management, Munich, Germany*, May 2007.
- [3] B. M. C. Hyser, R. Gardner, and B. J. Watson, “Autonomic virtual machine placement in the data center,” *HP Technical Report HPL-2007-189*, Feb 2008.
- [4] Y. Shang, D. Li, and M. Xu, “Energy-aware Routing in Data Center Network,” *Green Networking '10 Proceedings of the first ACM SIGCOMM workshop on Green networking*, pp. 1–8, 2010.
- [5] U.S. Environmental Protection Agency, Data Center Report to Congress. [Online]. Available: <http://www.energystar.gov>.
- [6] A. Greenberg, J. Hamilton, D. Maltz, , and P. Patel, “The Cost of a Cloud: Research Problems in Data Center Networks,” *ACM SIGCOMM CCR*, Jan 2009.
- [7] D. Abts, M. R. Marty, P. M. Wells, P. Klausler, and H. Liu, “Energy proportional datacenter,” *ACM ISCA, Saint-Malo, France*, Jun 2010.

- [8] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. Mckeown, "Elastictree: saving energy in data center networks," *USENIX NSDI, San Jose, CA*, Apr 2010.
- [9] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild," *ACM IMC, Melbourne, Australia*, Nov 2010.
- [10] M. Chen, H. Zhang, Y. Y. Su, X. Wang, G. Jiang, and K. Yoshihira, "Effective VM sizing in virtualized data centers," *IFIP/IEEE Integrated Network Management (IM), Dublin, Ireland*, May 2011.
- [11] S. Kandula, S. Sengupta, A. Greenberg, P. Patel, and R. Chaiken, "The nature of datacenter traffic: measurements and analysis," *ACM IMC, Chicago, IL*, Nov 2009.
- [12] X. Meng, V. Pappas, L. Zhang, and I. T. W. R. Center, "Improving the Scalability of Data Center Networks with Traffic-aware Virtual Machine Placement," *IEEE INFOCOM 2010 proceedings*, 2010.
- [13] J. T. Piao and J. Yan, "A Network-aware Virtual Machine Placement and Migration Approach in Cloud Computing," *2010 Ninth International Conference on Grid and Cloud Computing*, 2010.
- [14] Energy Efficiency for Information Technology by Lauri Minas and Brad Ellison. [Online]. Available: <http://www.intel.com.tw/>
- [15] RFC 2328 OSPF Version 2. The Internet Society. OSPFv2. Retrieved.
- [16] RFC 1058, Routing Information Protocol, C. Hendrik, The Internet Society.

- [17] G. H. Liu, H. P. W. Charles, and L. C. Wang, "D2ENDIST: Dynamic and Disjoint ENDIST-based Layer-2 Routing Algorithm for Cloud Datacenters," *IEEE GLOBECOM 2012*, pp. 1611–1616, Dec 2012.
- [18] N. McKeown, T. Anderson, G. P. H. Balakrishnan, L. Peterson, J. Rexford, S. Shenker, , and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," *SIGCOMM Comput. Commun. Rev.*, pp. 69–74, 2008.
- [19] J. Kleinberg, Y. Rabani, and E. Tardos, "Allocating bandwidth for bursty connections," *SIAM Journal on Computing*, vol. 30, pp. 191–217, 2000.
- [20] H. Jin, D. Pan, J. Xu, and N. Pissinou, "Efficient VM Placement with Multiple Deterministic and Stochastic Resources in Data Centers," *GLOBECOM 2012*, 2012.
- [21] Thaler, D. and C. Hopps, Multipath Issues in Unicast and Multicast, RFC 2991, November 2000.
- [22] C. Suh, K. Kim, , and J. Shin, "Endist: Edge node divided spanning tree," vol. 1, pp. 802–807, Feb 2008.
- [23] Z. Anyu, W. Huiqiang, and P. Song, "A Study on Matching Algorithm in Multilevel K-way for Partitioning Topology under The Cognitive Network Environment," *20 II International Conference on Computer Science and Network Technology*, pp. 24–26, Dec 2011.
- [24] METIS - Serial Graph Partitioning and Fill-reducing Matrix Ordering. [Online]. Available: <http://glaros.dtc.umn.edu/gkhome/metis/metis/overview>

- [25] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "V12: a scalable and flexible data center network," *Commun. ACM*, vol. 54, pp. 95–104, Mar 2011.
- [26] R. N. Mysore, A. Pamboris, N. Farrington, P. M. N. Huang, S. Radhakrishnan, V. Subramanya, and A. Vahdat, "Portland: a scalable fault-tolerant layer 2 datacenter network fabric," *SIGCOMM Comput. Commun. Rev.*, vol. 39, pp. 39–50, Aug 2009.
- [27] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu, "Bcube: a high performance, server-centric network architecture for modular datacenters," *SIGCOMM Comput. Commun. Rev.*, vol. 39, pp. 63–74, Aug 2009.
- [28] M. Schlansker, J. Tourrilhes, Y. Turner, and J. Santos, "Killer Fabrics for Scalable Datacenters," *Communications (ICC), 2010 IEEE International Conference*, pp. 23–27, May 2010.
- [29] Java: general-purpose, concurrent, class-based, object-oriented computer programming language. [Online]. Available: <http://docs.oracle.com/javase/tutorial/>
- [30] The ns-2 network simulator. [Online]. Available: <http://www.isi.edu/nsnam/ns>
- [31] Mininet: Rapid prototyping for software defined networks. [Online]. Available: <http://yuba.stanford.edu/foswiki/bin/view/OpenFlow/Mininet>

Vita

Shao-Heng Wang

He was born in Taiwan, R. O. C. in 1988. He received a B.S. in Communications Engineering from Chiao-Tung University of Technology in 2011. From July 2011 to August 2013, he worked his Master degree in the Mobile Communications and Cloud Computing Lab in the Department of Communication Engineering at National Chiao-Tung University. His research interests are in the field of wireless communications and mobile computing.

