EQVMP: Energy-efficient and QoS-aware Virtual Machine Placement for Software Defined Datacenter Networks

Shao-Heng Wang, Patrick P.-W. Huang, Charles H.-P. Wen and Li-Chun Wang
Institute of Electrical and Computer Engineering
National Chiao Tung University, Taiwan
Contact Email:ryan1202.cm00g@nctu.edu.tw, kfpat1125@gmail.com, opwen@g2.nctu.edu.tw, lichun@g2.nctu.edu.tw

***Abstract*—To provide effective and reliable services, cloud datacenters need parallel computing and virtualization techniques. This work presents an improved virtual machine (VM) placement mechanism, called Energy efficiency and Quality of Service (QoS) aware 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 three key techniques: *(1) hop reduction*, *(2) energy saving* and *(3) load balancing*. Hop reduction can regroup VMs to lower the 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.**

I. 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.

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 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. Thanks for 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.

Although network performance in VM placement policy is an important issue, we still can't overlook the effects of energy consumption. A report from Intel illustrates how processor energy consumption (e.g., performance per watt) increases as server utilization increases for a typical workload. However, the overhead is large when a processor is powered on. 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%. On the other hand, 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) [14] and Routing Information Protocol (RIP) [15], provide static routing choices and 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 [16]. Traditional routing algorithms limits the capacity utilization so that it cannot reach the optimization network. In addition, the cost of maintenance of those delicate hardware is high.

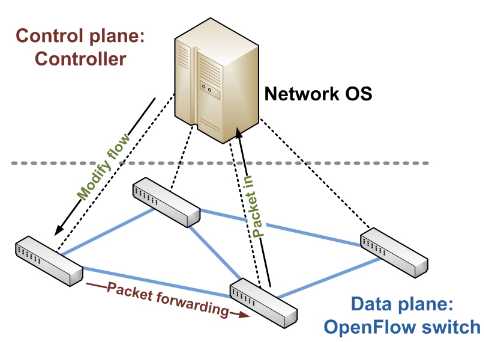

Fig. 1. Software defined network.

Software Defined Network (SDN), as shown in Fig. 1, establishes flexible and programmable network by separating the control plane and the data plane [17]. 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.

Traditional VM placement techniques only pay attention to the efficiency of resource allocation. When applying to datacenters, it may happen unexpected congestion and degradation in the network. Therefore, we propose a three-tier algorithm to take both energy efficiency and QoS into consideration. First, it partitions VMs to reduce traffic transmission across the entire datacenter. Then, it decides the minimum number of server without service-level agreement (SLA) violation. Last, the OpenFlow controller assigns the paths to avoid congestion and balance the network load.

The experimental results show that the proposed algorithm can significantly save energy and guarantee QoS 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].

The rest of this work is organized as follows. Section 2 introduces some backgrounds of our proposed approach and our problem. Then, we discuss system models of energy-efficient and QoS-aware VM placement in Section 3. Experimental results are shown in Section 4. Finally, we conclude this work in section 5.

II. Background and Problem

A. *Background*

*1) **VM placement**:* Most VM placement techniques focus on energy saving to minimize the number of servers. Nowadays, due to the rising concerns on datacenter energy and emerging bandwidth intensive applications [18], power consumption and bandwidth requirement are indeed taken into account. Therefore, some research have paid attention to finding optimal VM placement for multiple resource demands. A research [19] mentions that VM demands for certain resources are highly bursty and can be modeled as stochastic processes. They propose an algorithm to solve the traditional bin packing problem with multiple resources. On the other hand, some approaches indicate that energy consumption of network components is a vital issue in datacenter. Energy-aware VM Placement [1] presents a VM placement considering the balance between servers and network energy devices consumption. They aims at reducing energy consumption by meeting the constraints of both server-side and network data transmission. However, it may happen unpredictable congestion in the real network because they neglect routing issue in datacenter networks.

While some reports [13] show that up-growing data-intensive applications often need to communicate with related datacenters frequently. Therefore, the traffic load among those VMs are especially heavy. It may increase system overhead and degrades the network performance. Moreover, those energy efficient placements seek to consolidate VMs for resource utilization, which can also greatly impact network performance. This can lead to situations in which VM pairs are placed on host machines whose traffic are large. Traffic-aware VM Placement Problem [12] was proposed to solve the optimization problem based on different datacenter architecture and traffic pattern. 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 this paper illustrates the importance of network performance, it ignores energy consumption.

*2) **Dynamic Routing**:* 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 are proposed such as RIP [15], OSPF [14] and Equal-Cost Multi-Path routing (ECMP) [20]. RIP and OSPF are single-path routing which suffer from poor system throughputs. Moreover, computation capacity of OSPF routing on each individual node may degrade badly 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 uses each link for transmission in turn and may balance the traffic load and induce better network performance. However, the receiving out-of-order packets cannot be avoided and may incur more cost on the system.

While D^{2}ENDIST [16] was proposed to provide disjoint routing paths and serving as a dynamic-routing mechanism. One of the ideas originates from ENDIST [21], 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, was built upon a spanning tree algorithm that divides weighted edge nodes. In the proposed method, all of the routing paths are totally disjoint. 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 (a.k.a. D2ENDIST) was developed to eliminate the load unbalancing during data transmission in datacenter.

B. *Problem formulation*

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 a matrix form.

Datacenter studies always discuss separately the issue of resource allocation and networking. The problem of calculating the minimum number of power-on server and reducing total network delay, meanwhile maintaining 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-efficient and QoS-aware mechanism to solve the aforementioned problem.

III. Energy-Efficient and QoS-Aware VM Placement (EQVMP)

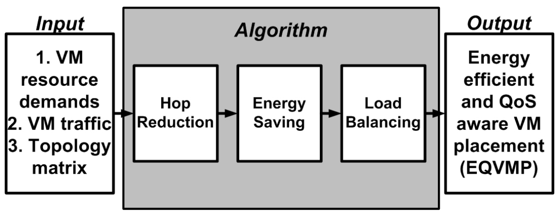

Fig. 2. Energy-efficient with QoS-aware VM Placement algorithm.

Our overall flowchart is shown in Fig. 2. We take VM resource demands, VM traffic and topology matrix as input. First, we divide VMs into groups in terms of hop count reduction. Then, energy saving process can minimize the number of power-on servers. After that, the OpenFlow controller balances traffic flow to avoid congestion based on the network status.

Step 1: Hop Reduction

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 transfer the matrix into a graph. Vertices are VMs and edges are the traffic among VMs, which can be modeled as graph partitioning problem. Graph partition problem is defined on data represented in the form of a graph G, with 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 that consists 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 our scenario, the VMs traffic information retrieved from the snapshot. Based on the collected traffic loads and datacenter topology architecture, hop reduce mechanism is proposed to partition VMs into groups that number of VM in each group is balanced and costs between different groups are minimized. Fig. 3 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 between groups is 20 so that heavy traffic load can cause great delay across the datacenter. An unequal partitioning groups, {1} and {2,3,4,5,6}, has the minimum mutual traffic load 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. 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.

VM Traffic Matrix

	1	2	3	4	5	6
1	0	1	0	0	0	0
2	1	0	6	2	0	0
3	0	6	0	0	3	0
4	0	2	0	0	8	5
5	0	0	3	8	0	2
6	0	0	0	5	2	0

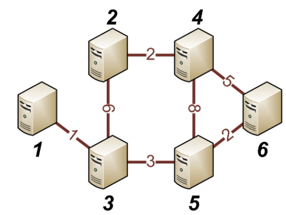

Fig. 3. VM partitioning with traffic matrix and graph.

Step 2: Energy Saving

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. Yet it is not an energy efficient VM placement, the random placement may lead to situations that some servers can not meet the resource demand 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. n VMs are placed into m servers by considering their resource demands (CPU consumptions, memory usages and bandwidth demands).

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 (M^3SBP). 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 new server can fit each of them. Then, we select the candidate VM which can be placed on the current server with minimum resource left. If there is no candidate VM in the set, it illustrates that none of existing servers has capacity to host VMs. 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 Fig. 4 shows. For the record, energy saving module provides sufficient bandwidth. However, we cannot guarantee sufficient bandwidth in datacenter networks so far.

Step 3: Load Balancing

Fig.5 illustrates an example that load balancing mechanism can improve network performance greatly. After the process

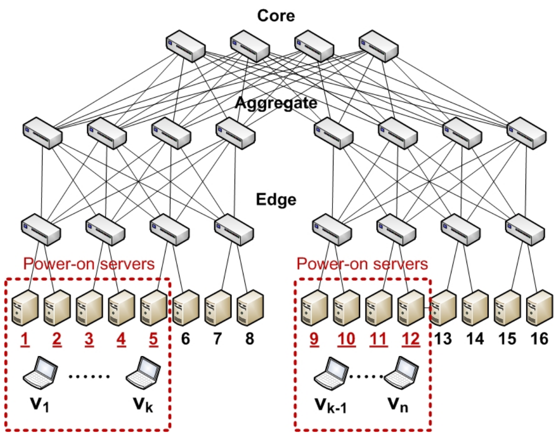

Fig. 4. VM placement after energy saving.

of hop reduction and energy saving, we assume that VMs are placed under switch 6 and switch 8. With traditional static routing algorithm, we notice that all of the traffic flow along the path 6-2-0-4-8, where congestion induced. 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.

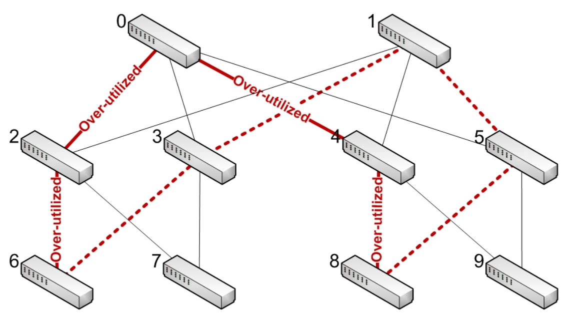

Fig. 5. VM placement with load balance mechanism.

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 datacenter, 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 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 status of energy efficiency and delay reduction. Then, load balancing repeats again.

IV. Experimental Results

Through our proposed mechanism, we may indicate that there are two parts of our experiment. One is to determine how to arrange VMs on servers based on energy saving and hop reduction. The other is to put our VM placement on the simulation tool to observe network performance. We apply Java [28] programming to compute our VM placement setting. Then, we use NS2 [29] as our simulation tool. Other Simulation tools, such as Mininet [30], is also able to create OpenFlow network environment and emulate controller's behaviour of managing the flows. However, Mininet is unable to quantify traffic and bandwidth information so that it is impossible to evaluate network performance. 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 network snapshot. The bandwidth's demands of VMs meet the uniform distribution from 0 to 100% utilization of the link bandwidth (10 Mb). Power consumptions and memory usages also meet uniform distribution $U(0, 100)$, which are represented in the utilization percentage of a server. Besides, most of the data transmission among VMs is related and confined in certain VMs [13]. We apply group traffic as our traffic pattern. In our experiment, the default simulation time of these two networks is set as 100 seconds.

A. *Performance Improvement*

In the first experiment, we implement our energy-efficient and QoS-aware VM placement to observe objectives, throughput, delay and number of power-on server, in different phases of our system model. From Fig. 6, 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 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.

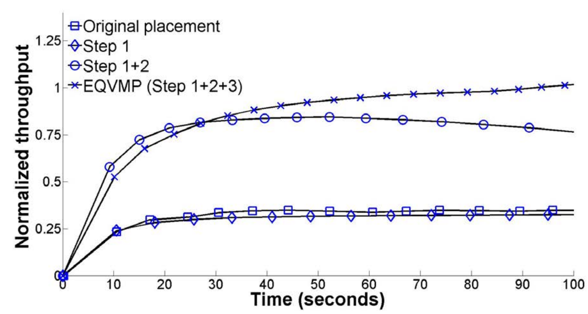

Fig. 6. Throughput in different phases.

B. *Impact of Update Period in Routing*

In the second experiment, we discuss the factor of update period related to the computational time. In Fig. 8, computational time of VM placement is composed of METIS

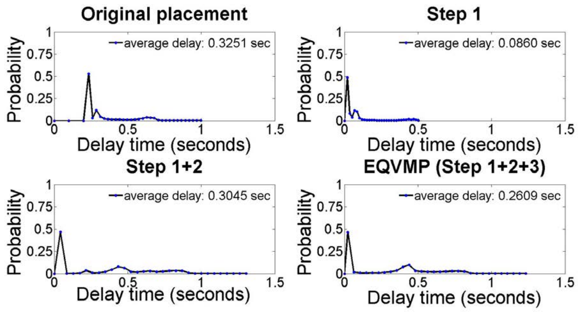

Fig. 7. Delay in different phases.

partitioning time and resource allocation time. Owing to computation efficiency of METIS, we can roughly estimate resource allocation time as 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. 9 shows the throughput comparison between different update period. Obviously, as long as update period is greater than the minimum update period, we obtain the better network performance with more frequently update.

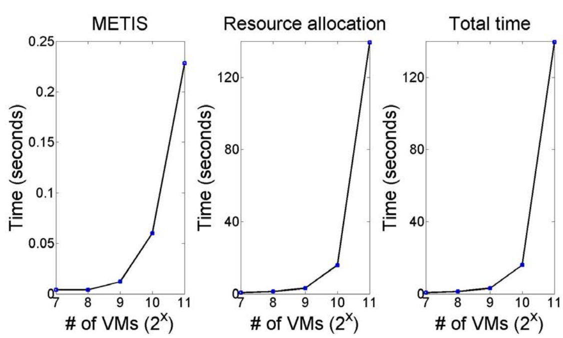

Fig. 8. Computational time of VM placement.

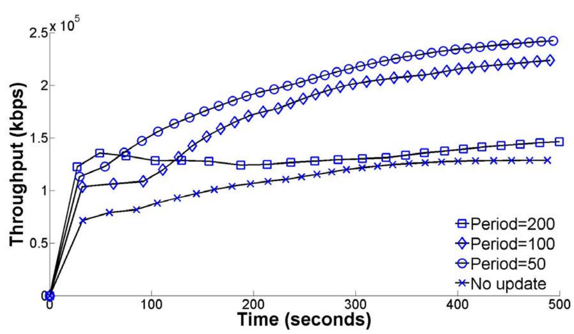

Fig. 9. Performance with different update period.

C. *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. 10, 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-efficient and QoS-aware and enhances system throughput by 25%.

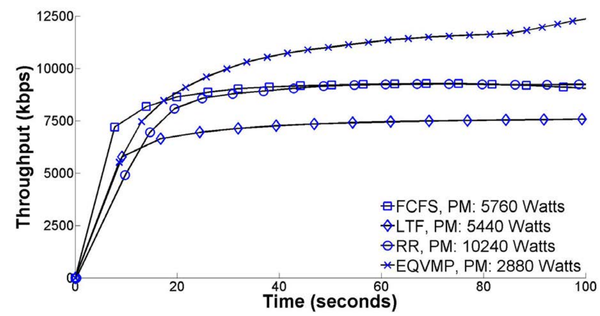

Fig. 10. Comparison between different VM placement policies.

D. *Evaluation Score*

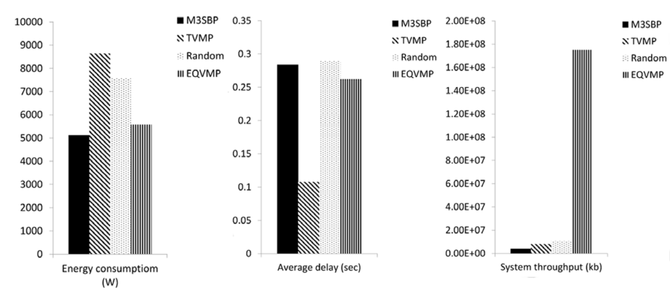

Fig. 11. Comparison between different awareness of VM placement.

Fig. 11 illustrates the comparison among Max-Min Multi-dimensional Stochastic Bin Packing (M^{3}SBP), 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 closed 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.

$$ES = \alpha \cdot \frac{E_{min}}{E} + \beta \cdot \frac{D_{min}}{D} + \gamma \cdot \frac{T}{T_{max}}. \quad (1)$$

To provide a reliable and general evaluation score of a VM placement on the aspect of energy and QoS, we propose an equation 1 to determine if the placement is good or not based on the theory of linear programming [1]. We can use different weight to observe the tendency of placement policies. Given that E_{min} is the minimum number of power-on servers in the M^3SBP and the TVMP only partitions VMs with 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 throughout. The weights of each terms, α, β and γ, satisfy that α+β+γ=1.

Table I indicates our proposed has the best evaluation score in the balance weights (α=β=γ). When we apply different weights (Energy-critical: α=0.5, β=0.25, γ=0.25; Delay-critical: α=0.25, β=0.5, γ=0.25; Throughput-critical: α=0.25, β=0.25, γ=0.5;), our evaluation score is still superior than others because of the excellent performance in throughput.

TABLE I
EVALUATION SCORE LIST

Placement	Energy-aware	Delay-aware	Random	Ours
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

V. CONCLUSION

Both of energy and QoS are critical issue in datacenter networks. Owing to the application's massive bandwidth demand and strict delay constraint, it's important to maintain effective datacenter networks condition with minimum resources. Many previous works proposed on VM placement policy guarantee VMs to have sufficient resources and utilize the network resources more effectively. However, unbalance and aggressive placement still induces severe congestion in datacenter networks. Therefore, we propose the energy-efficient and QoS-aware VM placement (EQVMP) mechanism.

Experiments prove that EQVMP can provide better system throughput than other VM placement strategies. It determines a good VM placement considering energy consumption, hop delay and network throughout. Although our energy and delay performance are the second best, EQVMP outperforms other placement schemes and achieve 10 times throughput than energy-aware and delay-aware placement. To make comparison, we propose an evaluation score to indicate the score of VM placement policies, and EQVMP is superior on every viewpoints of our consideration. Our computation time of the new VM placement configuration is large. However, based on our scenario, we provide a long update period to relocate VM placement rather than frequently adjusting them.

REFERENCES

[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 centernetworks," 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, Mov 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 Interna- tional Conference on Grid and Cloud Computing, 2010.

[14] RFC 2328 OSPF Version 2. The Internet Society. OSPFv2. Retrieved.

[15] RFC 1058, Routing Information Protocol, C. Hendrik, The Internet Society.

[16] 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.

[17] 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.

[18] J. Kleinberg, Y. Rabani, and E. Tardos, "Allocating bandwidth for bursty connections," SIAM Journal on Computing, vol. 30, pp. 191- 217, 2000.

[19] H. Jin, D. Pan, J. Xu, and N. Pissinou, "Efficient VM placement with multiple deterministic and stochastic resources in data centers," GLOBECOM 2012, 2012.

[20] Thaler, D. and C. Hopps, Multipath Issues in Unicast and Multicast, RFC 2991, November 2000..

[21] C. Suh, K. Kim, , and J. Shin, "Endist: Edge node divided spanning tree," vol. 1, pp. 802-807, Feb 2008.

[22] 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.

[23] METIS - Serial Graph Partitioning and Fill-reducing Matrix Ordering. [Online]. Available: http://glaros.dtc.umn.edu/gkhome/metis/metis/overview

[24] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "Vl2: a scalable and flexible data center network," Commun. ACM, vol. 54, pp. 95-104, Mar 2011.

[25] 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.

[26] 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.

[27] M. Schlansker, J. Tourrilhes, Y. Turner, and J. Santos, "Killer fabrics for scalable datacenters," Communications (ICC), 2010 IEEE Interna- tional Conference, pp. 23-27, May 2010.

[28] Java: general-purpose, concurrent, class-based, object-oriented computer programming language. [Online]. Available: http://docs. oracle.com/javase/tutorial/

[29] The ns-2 network simulator. http://www.isi.edu/nsnam/ns

[30] Mininet: Rapid prototyping for software defined networks. http://yuba.stanford.edu/foswiki/bin/view/OpenFlow/Mininet.