An Extensible Load Balancing Approach using Column Generation in Cloud Data Centers

Hemalatha.A[1], PG Student[1], Ramadoss.P[2], Asst. Professor,
Department Of Computer Science and Engineering,
[1]PG Student, Parisutham Institute of Technology and Science, Thanjavur, Tamilnadu, India.
[2]Asst. Professor, Parisutham Institute of Technology and Science, Thanjavur, Tamilnadu, India.

***Abstract*— Cloud Computing is significant pattern has started to obtain mass appeal in corporate data centers as it enables the data center to operate like the Internet. This unique technology brings new defies, mostly in the proprieties that govern its fundamental frame. Traffic engineering in cloud data centers is one of these challenges that has fascinated consideration from the research community, particularly since the legacy properties employed in data centers offer limited and non-extensible traffic management. Many promoted for the use of VLANs as a way to provide extensible traffic management, however, finding the optimal traffic split between VLANs is the well known NP-Complete VLAN assignment problem. The size of the search space of the VLAN assignment problem is huge, even for small size networks. This paper introduce a unique decomposition approach to solve the VLAN mapping problem in cloud data centers through column generation. Column generation is an effective technique that is proven to reach optimality by exploring only a small subset of the search space. We introduce both an exact and a semi-heuristic decomposition with the objective to achieve load balancing by minimizing the maximum link load in the network.**

Keywords*— *Data centers, load balancing, optimization, column generation.

I. INTRODUCTION

Cloud computing services are currently being deployed across a range of market sectors to help improve the scalability and cost effectiveness of Information Technology (IT) services. Data centers and networking infrastructure are key enablers for the rapid rise and adoption of cloud computing as well as a wide range of other networking applications (e.g., video-streaming, data storage, web search, etc.). Typically, data centers rely on Layer 2 switched networks [1], [2] to inter-connect tens or thousands of physical servers hosting virtual machines (VMs) belonging to various cloud providers to offer their services. Such server virtualization provided by data centers addresses some limitations of traditional data centers; namely, virtualization
results in higher server utilization, reduced operational cost, better application isolation and hence better performance [3].

Scalable traffic management in data center networks has recently become a factor of utmost importance [2], [10], [11], [12], [13]. By making use of Virtual Local Area Networks (VLANs), data center networks are able to logically isolate the traffic and resources of the various applications they host. VLANs partition nodes in the network into communities of interest. Servers within one community can only communicate with servers that belong to the same community (or VLAN). Such practice allows both performance isolation and network scalability, since packets exchanged within a VLAN do not stretch to the entire network. Additionally, to exploit path redundancy in the network, the Multiple Spanning Tree Protocol (MSTP), an extension of STP, allows different VLANs to use different spanning trees traversing diverse physical links and switches. A traffic engineering framework for MSTP in large data centers is presented in [14] where the authors focused on the mapping of each VLAN into a spanning tree to improve the overall network utilization.

The VLAN assignment problem is NP-complete with a large search space [9]. For instance, for a network with v VLANs and c flows, there are v c possible mappings [10]. Finding the best mapping is therefore a large-scale combinatorial problem. Several prior works have attempted to address this or simpler variation of this problem. For instance, SPAIN [8] proposed a greedy heuristic for packing flows or paths into a minimal set of VLANs and the sub-graphs containing these VLANs are dynamically constructed. A constraint based local search based on Constraint Programming (CP) is presented in [14] for mapping flows into VLANs, assuming the set of VLANs is given. The authors of [10] used Markov Approximation techniques to solve the mapping problem and designed approximation algorithms with close to optimal performance guarantees.

In this paper, we consider the problem of traffic engineering in data center networks; namely, we characterize each tenant request by a set of VMs communicating with each other. We address the problem of mapping traffic flows of each tenant (flows between VMs) into VLANs; similar to [10], a separate spanning tree is constructed per each VLAN. As the number of spanning

trees in a network could be very large (e.g., in a fully connected graph with n nodes, there are as many as nn−2 spanning trees), selecting the most promising spanning trees to map the traffic flows onto is a very challenging and complex combinatorial problem. This paper jointly addresses the problem of tree construction and flow mapping, a seemingly very large combinatorial problem. To keep track of the problem, we follow a primal-dual decomposition approach using Column Generation; here, the problem is divided into two sub-problems (Master and Pricing), the former builds sub mappings (a single spanning tree with mapped flows), and the latter, selects among the sub mappings, the global mapping of the problem. Our method proves to be highly scalable in addressing the joint VLAN mapping and tree construction problem.

The rest of the paper is organized as a discussion of the state of the art work in data centers traffic engineering, problem definition and column generation approach for solving the VLAN assignment problem.

II. BACKGROUND

Today's data center networks are expected to provide high network utilization to the large number of distinct services they support. These networks are also expected to provide guaranteed performance for the various hosted tenants sharing the same networking infrastructure. Intelligent traffic engineering mechanisms inside a data center become therefore of utmost importance to achieve better load balancing and guarantees on the quality of service. Recently, the problem of traffic management in data center networks has been receiving numerous attention with the goal of providing predictable performance. The issue of performance predictability becomes of particular interest given the nature of data center topologies, which tend to be oversubscribed [2]. The authors of [15] noted that variability in network performance harms application performance and causes service provider revenue loss. As a result, they advocated to provide tenants with virtual networks connecting their computing (VM) instances. The authors proposed a virtual network abstractions which capture a tradeoff between application performance and provider costs. The abstraction is a mean to expose the tenant requirements to the provider as well as the virtual network interconnecting the VMs. The problem reduces to a problem of resource allocation of VMs into physical servers where the network manager needs to ensure that bandwidth demands can be satisfied while maximizing the number of concurrent tenants. Ultimately, the authors have shown that tenants can get predictable performance in a multitenant shared data center network infrastructure.

Hedera [17] is a dynamic flow scheduling system for data center networks based on multistage switch topologies. Hedera depends on a central scheduler that measures link utilization in the network and move flows from highly utilized links to less utilized ones. When the scheduler detects a flow with augmenting bandwidth demand that exceeds a certain threshold, it computes a non-conflicting path to route this flow in order to improve the bandwidth-bisection in the network. Hedera uses simulated annealing and the global first fit heuristics for path computation.

Now to achieve high bisection bandwidth in data center networks, the authors of [8] proposed SPAIN, which Provides multipath forwarding using Layer-2 Ethernet switches over arbitrary topologies. Here, SPAIN aims at finding multiple paths between every pair of nodes, and then grouping these paths into a set of trees, each tree is mapped as a separate VLAN onto the physical network. The path assignment into VLANs uses a greedy packing heuristic with the objective of using the minimum number of VLANs. SPAIN's performance was validated through both simulations and experiments and has shown to achieve superior good-put over STP. Traffic engineering in data center networks has been a topic of surging interests recently; particularly, exploiting path redundancy in the network and mapping paths into VLANs to achieve higher utilization and better load balancing is addressed in [14].

Essentially, the problem considered is that of mapping routing classes to VLANs to achieve load balancing. In [20], a VLAN placement algorithm is presented for growing spanning trees to reach all switches in the network and subsequently traffic splitting onto those VLANs is presented (through linear programming) to achieve load balancing. However, the heuristic construction of spanning trees does not provide any guarantees on the quality of the generated solution. More recently, the authors of [10] considered the problem of traffic engineering with ensemble routing and noted the combinatorial challenge of optimizing the assignment of routing classes into VLANs. They used the Markov approximation framework for approaching the mapping problem and developed approximation algorithms with close to optimal performance.

III. PROBLEM DEFINITION

In this section, we present a formal definition of the VLAN assignment problem. We present computational analysis to prove the NP-Complete nature problem through an illustrative example the exponential search space that the VLAN assignment problem poses.

A. VLAN assignment Problem

Here, consider a data center network whose underlying physical topology is represented by a graph G(N,L), N is the set of network nodes and L is the set of communication links. Each link connects a pair of network nodes (i, j) and

has a capacity Ri,j . We assume a multi-tenant data center, each tenant requests virtual machines with computing and storage resources and specifies the complete pairwise bandwidth demand matrix between its VMs. Note that we do not address here the problem of embedding virtual machines into the data center, but rather, and similar to [10], we focus on the problem of mapping traffic flows between VMs into VLANs. We acknowledge that the maximum number of VLANs a switched Ethernet network supports is 4096, however, the number of spanning trees in the network could substantially exceed that. Given the traffic matrix, finding the most useful spanning trees to these VLANs is a combinatorially complex and challenging problem.

Problem 1: The VLAN assignment problem with the objective of maximizing the network capacity is NP-Complete.

Attempting to maximize the throughput or bisection bandwidth means admitting as many flows as possible into the network. This indeed yields better revenues for network providers by allowing them to host more cloud services. Alternatively, if one were to load balance the traffic across the network and minimize congestion to achieve better quality of service for the communicating VMs, then minimizing the maximal link load becomes the objective. This objective is necessary for building service level agreement (SLA) and QoS guarantee in data center networks. Indeed, solving the VLAN assignment problem under either of these objectives is an NP-Complete.

Problem 2: The VLAN assignment problem with the objective of minimizing the maximum link utilization is NP-Complete

We relax the VLAN assignment problem such that all links have the same capacity, and each flow will be routed on every link in the VLAN. Note that while multiple flows can be mapped to the same VLAN, the same flow cannot be mapped to more than one VLAN. Thus, in this relaxed version of the problem, the knapsacks are the VLANs and the items are the flows. Each VLAN has a capacity equal to the uniform capacity of the links and the flows each has a bandwidth requirement. Placing a flow on a VLAN will place a bandwidth load on every link in the VLAN, and thus decreases its overall capacity. Our goal is to admit as many items (flows) as possible. Since the relaxed version of the problem is NP-Complete, adding more constraints to the problem, by having specific origin and destination for each connection and that one link can belong to multiple VLANs, is therefore NP-Complete too.

The VLAN assignment problem is formulated next:

Min Max Uij

$$Uij = \left(\sum_{u \in V} t^{u}_{ij} \right) / R_{ij} \quad \forall\, v \in V, (i, j) \in L \qquad (1)$$

A connection cannot use a link in a VLAN, if that link does not belong to the spanning tree of that VLAN. Finally, constraint (1) calculates the utilization of every link in the network.

IV. DECOMPOSITION MODEL

In the following section, we present our column generation model, as well as, the intuition by which we decomposed the original VLAN assignment problem into two sub-problems towards a scalable solution to the aforementioned hurdle.

A. Column Generation

Column Generation is an efficient method for solving large LP problems [22]. Normally, given an LP, the algorithm begins with an initial subset of configurations (columns) (M0) that satisfies all the constraints. At each iteration, a new configuration ($m \in M$, where M is the set of all possible configurations or columns), that ameliorates the objective function, is added to the initial set ($M0 = M0 \cup m$). Thus, unlike the SIMPLEX method, column generation only explores a subset of the variables, instead of enumerating all of them. In every iteration of the SIMPLEX method, the goal is to find the next non-basic variable to enter the set (basis), which is the one with the minimum reduced cost coefficient. The SIMPLEX method resorts to calculating the reduced cost coefficient of all non-basic variables whereas column generation alleviates this by only finding this next entering variable. Column generation decomposes the initial problem into two sub-problems, a master (LP model) and a pricing (ILP). The master is in charge of determining if the explored configurations satisfies all the constraints. The pricing model is in charge of finding a new configuration to be passed on to the master. If the newly found configuration was deemed feasible (and will improve the objective) by the master, it will be added as a new column, hence the name of the method. The pricing's objective function is in fact the reduced cost coefficient of the master. Thus, the master model can be referred to as the primal, while the pricing as the dual. The master and the pricing problems alternates until no new configurations are found with a negative reduced cost coefficient (in the case of a minimization problem) which indicates that optimality is reached.

B. The Intuition Behind our Decomposition Approach

Decomposing the VLAN assignment problem to be solved using the column generation method is not straight forward. To achieve a concise and vigorous decomposition, we attempted to reformulate the problem in a way that resembles the famous cutting-stock problem. In the cutting stock problem, the difficulty resides in the very large number of cuts available to choose from. This indeed resembles the VLAN assignment problem where a very large number of mapping possibilities between flows and VLANs exists. Similar to the cutting stock problem, we envision a cut in the VLAN assignment problem as a unique mapping of a subset of flows to a particular VLAN. Finding the optimal set of cuts is nothing but finding the optimal way our various flows can be mapped to a subset of the available VLANs. Following these pinpoints, we can divide our problem into two sub-problems: the pricing will be in charge of finding a new configuration, while the master will ensure that these configurations do not violate the capacity constraints.

C. The Column Generation Model

Column generation model for the VLAN assignment problem is here introduced. We have modified the problem definition to include not only finding the optimal mapping of flows to VLANs, but also finding the optimal set of VLANs (spanning trees) onto which these flows will be mapped. In fact, as the number of possible spanning trees in a network can be very large, limiting the search space to those VLANs that offer a good quality solution will greatly improve the runtime of the model. We introduce a new variable m that denotes a configuration. A configuration is defined as a VLAN v onto which flows are mapped.

Problem 3: Given a network G(N,L), and a set of flows C, each with bandwidth demand δc, find the optimal set of configurations, such that the maximum link utilization is minimized.

Algorithm: Modified Column Generation Approach
1: Given G(N,E) */*an arbitrary topology*/*
2: ST = *GenerateSpanningTrees*;
3: *RC = -μ /*μ is a very large positive number*/*
4: *M*0 = { };
5: **while** (!Terminate) **do**
6: st = ST.next();
7: */*Initialize Set of Configurations for spanning tree st*/*
8: *Mst* = { };
9: **while** ((RC ≤ 0)) **do**
10: RC = Run Master;
11: */*m is a new Configuration*/*
12: *m* = Run Pricing(RC,st);
13: RC = m.RC;
14: *Mst* = *Mst* ∪ *m*;
15: **end while**
16: **if** (*Mst*.length ≤ 1) **then**
17: Terminate = *TRUE*;
18: **end if**
19: *M*0 =*M*0 ∪ *Mst*;
20: **end while**

Algorithm illustrates the methodology of our heuristic model: Given the topology of the substrate network, represented by G(N,E), the model begins by calling the GenerateSpanningTree function to enumerate multiple spanning trees that will be stored in a dedicated set ST. As we have previously mentioned, the GenerateSpanningTree function adopts the Dijsktra algorithm with link weights increment. Next, at the beginning of every iteration of the pricing model, we provide the pricing with a new spanning tree st from the set ST, and for every tree, we initialize a set Mst that represents the list of configurations for that given tree. The master and the pricing iterate over the same given tree, as long as it produces feasible configurations (a configuration with positive reduced cost). At the end of evey iteration, the explored configuration m is added to the set Mst. Once the tree becomes non-bearing (no feasible configurations can be found), a new tree from the initial set is selected.

This procedure persists until one tree returns an infeasible configuration in the first trial, meaning that its Mst remains empty at the end of the first Master/Pricing iteration, in that case, the program terminates. Our numerical results have shown that our heuristic model achieves a great improvement in runtime and scalability over our initial exact decomposition model without sacrificing the quality of the final solution.

V. PERFORMANCE EVALUATION

Numerically evaluate the proposed VLAN mapping methods. In particular, we present comparisons between the pure ILP and its decomposition variations. We refer to the column generation models presented as CG1 and CG2. The objective of our comparisons is to study the effectiveness of the designed methods in terms of quality of the obtained solutions and runtime.

The size of the ILP pricing of CG1 is much bigger and thus less scalable in terms of running time. Figure 3 compares the run times of both CG1 and CG2 over larger clique topologies of 3 to 15 nodes with all to all connection demands.

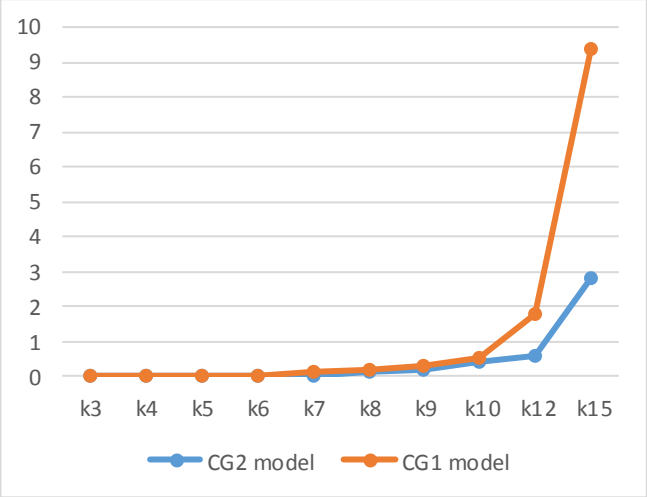

Figure 1: Runtime results comparison between CGI models

The figure shows that the runtime of CG2 is much faster than CG1, in fact as we move towards cliques with more than 8 nodes, CG2 becomes at least three times faster.

Finally, we examine the quality of the obtained solutions. We show, in Table I, the % gap of the solution obtained by CG2 and CG1 to the one obtained by the pure ILP. In most cases, the gap is less than 2% for CG1 and 4% for CG2. The gap between the solution of CG1 and that of the pure ILP is attributed to the manner through which we obtained the ILP solution for CG1.

VI. CONCLUSION AND FUTURE WORK

In this paper, we introduced a unique column generation approach for solving the VLAN assignment problem in cloud data centers. We present two decomposition approaches: an exact, as well as a semi-heuristic model to attain better runtime and scalability. We compare both models against the pure ILP model of the VLAN assignment problem, and prove that our approach yields a substantial decrease in the size of the explored search space with encouraging optimality gap.

We also compared our decomposition approach against state of the art protocols in traffic engineering, our comparative analysis has shown that our model outperforms its peers in most network topologies in terms of link load, attainable gap from lower bound LP solution, as well as in goodput. As we have previously mentioned, employing a more effective technique to go from the relaxed LP solution to the integral ILP solution can potentially improve our model's optimality gap. Hence, for our future work, we aim to employ ranchand-bound to affirm this proclamation. Also, in our current work, we have assumed that the traffic demands are given. VLAN assignment problem with unknown traffic demands is a more challenging problem, thus as future work, we aim to build on our current findings and devise an efficient online decomposition model that is suitable to the bursty, sporadic and unpredictable nature of demands in cloud data centers. For future work, we also plan to extend our mapping framework to consider delay sensitive flows, multicast flows, dynamic flows, etc.

ACKNOWLEDGMENT

I would like to thank my guide Prof. P.Ramadoss, Asst. Prof., Computer Science and Engineering Department, Parisutham Institute of Technology and Science, Thanjavur for his help and guidance to enable me to propose this system

REFERENCES

[1] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," ACM SIGCOMM Computer Commun. Rev., vol. 38, pp. 63–74, 2008.

[2] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. Maltz, P. Patel, and S. Sengupta, "VL2: a scalable and flexible data center network," ACM SIGCOMM Computer Commun. Rev., vol. 39, pp. 51–62, 2009.

[3] M. F. Bari, R. Boutaba, R. Esteves, L. Granville, M. Podlesny, M. Rabbani, Q. Zhang, and M. F. Zhani, "Data center network virtualization: a survey," 2012.

[4] U. Hlzle, "Google's data centers: an inside look." Available: http://googleblog.blogspot.ca/2012/10/googles-data-centers-inside-look.html, Oct. 2012.

[5] R. McMillan, "Amazon cloud powered by almost 500,000 servers." Available: http://www.wired.com/wiredenterprise/2012/03/amazon- c2/, 2012.

[6] C. Guo, H. Wu, K. Tan, L. Shiy, Y. Zhang, and S. Lu, "Dcell: a scallable and fault-tolerant network structure for data centers," ACM SIGCOMM Computer Commun. Rev., vol. 38, pp. 75–86, 2008.

[7] W. J. Dally and B. P. Towles, Principles and Practices of Interconnection Networks. Access Online via Elsevier, 2004.

[8] J. Mudigonda, P. Yalagandula, M. Al-Fares, and J. C. Mogul, "Spain: COTS data-center ethernet for multipathing over arbitrary topologies," in Proc. 2010 USENIX Conference on Networked Systems Design and Implementation, pp. 265–280.

[9] R. N. Mysore, A. Pamboris, N. Farrington, N. Huang, P. Miri, S. Radhakrishnan, V. Subramanya, and A. Vahdat, "Portland: a scalable faulttolerant layer 2 data center network fabric," ACM SIGCOMM Computer Commun. Rev., vol. 39, pp. 39–50, 2009.

[10] Z. Shao, X. Jin, W. Jiang, M. Chen, and M. Chiang, "Intra-data-center traffic engineering with ensemble routing," in Proc. 2013 IEEE confocom.

[11] M. Schlansker, Y. Turner, J. Tourrilhes, and A. Karp, "Ensemble routing for datacenter networks," in Proc. 2010 ACM/IEEE Symposium on Architectures for Networking and Communications Systems, p. 23.

[12] T. Benson, A. Anand, A. Akella, and M. Zhang, "MicroTE: fine grained traffic engineering for data centers," in Proc. 2011 COnference on Emerging Networking EXperiments and Technologies, p. 8.

[13] C. Kim, M. Caesar, and J. Rexford, "Floodless in SEATTLE: a scalable Ethernet architecture for large enterprises," ACM SIGCOMM Computer Commun. Rev., vol. 38, pp. 3–14, 2008.

[14] H. T. Viet, Y. Deville, O. Bonaventure, and P. Francois, "Traffic engineering for multiple spanning tree protocol in large data centers," in Proc. 2011 IEEE Internationsl Teletraffic Congress, pp. 23–30.

[15] H. Ballani, P. Costa, T. Karagiannis, and A. Rowstron, "Towards predictable datacenter networks," in Proc. 2011 ACM SIGCOMM, vol 11, pp. 242–253.

[16] A. Shieh, S. Kandula, A. Greenberg, C. Kim, and B. Saha, "Sharing the data center network," in Proc. 2011 USENIX Conference on Networked Systems Design and Implementation, pp. 23–23.

[17] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, "Hedera: dynamic flow scheduling for data center networks," in NSDI vol. 10, pp. 19–19, 2010.
[18] S. Sen, D. Shue, S. Ihm, and M. J. Freedman, "Scalable, optimal flow routing in datacenters via local link balancing," under submission, 2013.

AUTHOR DETAILS

A.Hemalatha received B.Tech (IT) from Periyar Maniammai college of Technology for Women, Anna University in 2005 and received MBA (Technology Management), Anna University in 2010. She is currently pursuing M.E – Computer Science in Parisutham Institute of Technology and Science, Anna University.

P.Ramadoss.,M.Sc.,M.Phil.,M.Tech., working as a Asst.Professor, Dept. of Computer Science and Engineering, Parisutham Institute of Technology and science, Thanjavur.