Computer Networks 139 (2018) 70–80

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

A spectral clustering approach to network-aware virtual request partitioning[☆]

Lingnan Gao[*], George N. Rouskas

North Carolina State University, Raleigh, NC 27695-8206, USA

ARTICLE INFO

Article history:
Received 20 December 2016
Revised 2 April 2018
Accepted 3 April 2018
Available online 5 April 2018

ABSTRACT

Virtual request partitioning is an essential subproblem of two common problems in virtual networks, namely, virtual network embedding (VNE) and virtual machine placement (VMP). In this study, we consider a network-aware variant of the problem where the objective is to partition a virtual request so as to minimize the total amount of inter-cluster traffic. This problem is equivalent to the (k, v)-balanced partitioning problem, an NP-complete problem. To handle the inherent complexity of this problem, we develop a spectral clustering-based partitioning scheme that produces good solutions in a reasonable amount of time. Our solution consists of several components: (a) spectral clustering, (b) a Constrained k-means partitioning algorithm that ensures that capacity limits for clusters are met, and for which we present a polynomial-time greedy algorithm, and (c) a greedy refinement algorithm using simulated annealing to further improve the clustering solution. Simulation results indicate that our algorithm outperforms existing partitioning schemes in terms of inter-cluster traffic minimization.

1. Introduction

Network virtualization is seen as crucial way in reshaping the Internet architecture and introducing diversity into the current network [1]. With network virtualization, conventional providers are decoupled into infrastructure providers (InP), who mainly focus on the management of the infrastructure, and service providers (SP), who are responsible for the creation of the network and provide end-to-end service to end users. Such an environment allows the deployment of network architectures regardless of the underlying infrastructure, and thus facilitates the evolution of network architecture [2]. The cloud computing paradigm also employs virtualization techniques. Data centers aggregate all the computing resources (including CPU, memory, and storage), and provide services to the end users in the form of virtual machines (VM). Server virtualization allows multiple VMs to co-locate on the same physical server to increase utilization and lower the operational cost [3].

A key challenge for network virtualization and cloud computing is resource allocation. In network virtualization, resource allocation arises in the context of the virtual network embedding (VNE) problem, where the objective is to embed the virtual network to the substrate network so as to maximize the benefit from the existing hardware [4]. In the area of cloud computing architecture, the related virtual machine placement (VMP) problem arises, whereby the objective is to optimally assign the VMs to physical hosts so as to utilize the available resources without performance degradation [5].

In either area, mapping virtual request to physical resources may involve partitioning of the virtual request. For the VNE problem, mapping virtual requests to multiple domains may be required for various reasons, including load balancing [6] and managing the embedding cost [7]; for the VMP problem, VMs must be mapped onto underlying physical resources that may span across physical hosts, racks, even data centers [8]. Therefore, for communication-intensive applications, mapping virtual requests to physical resources must be accomplished in a manner that satisfies capacity constraints and takes into account the communication cost and quality of service (QoS) requirements [8,9].

In this work, we consider the problem of virtual request partitioning and present an algorithm inspired by spectral clustering to partition the set of virtual nodes under capacity constraints. This algorithm produces high quality solutions, compares favorably to existing algorithms, and scales well; simulation experiments indicate that it can tackle virtual networks consisting of hundreds of nodes within a few seconds. Following the introduction, we review previous work in this topic in Section 2. In Section 3, we formally define the problem and present the various components of the virtual request partitioning algorithm. In Section 4, we present the results of simulation experiments we have conducted to compare

[☆] This work was supported in part by the National Science Foundation under Grant CNS-1111088.

[*] Corresponding author.
E-mail address: glingna@ncsu.edu (L. Gao).

https://doi.org/10.1016/j.comnet.2018.04.005
1389-1286/© 2018 Elsevier B.V. All rights reserved.

the performance of this algorithm to existing algorithms. We conclude the paper in Section 5.

2. Related work

Several studies [6,7,9] have addressed the virtual request partitioning problem using max-flow, min-cut schemes. With existing algorithms, it is possible to compute efficiently the maximum flow between a pair of nodes and obtain the minimum cut between them. The work in [7] recursively uses the max-flow, min-cut approach to partition the network into the desired number of clusters. In [6,9], a clustering approach based on Gomory–Hu trees is explored. A Gomory–Hu tree represents the $n-1$ minimum $s-t$ cuts in a graph of n nodes. By removing the $k-1$ least weight edges of this tree, a partition of the n nodes into k clusters is obtained that is close to optimal. The shortcoming of this approach is that the resulting clusters may be highly imbalanced, as the cluster capacity is not taken into account. In order to enforce the capacity constraints, further partitioning of an overloaded cluster and combination of small clusters is necessary. For instance, in an extreme case, when recursively using the max-flow min-cut approach to partition the network requests, one may keep obtaining the result of one node in one part while all the rest goes to the other; while for a Gomory–Hu tree, it is possible to end up with a star topology, and by removing one edge, each time, we can only obtain a cluster with a single node. However, there is no guarantee that the combination of those small clusters would lead to a small amount of inter-cluster traffic. In both cases, when we group those singleton nodes into a cluster, there is no evidence that the traffic among those nodes is high. As intra-cluster traffic is not necessarily low, this, in turn, implies a potentially high inter-cluster traffic.

The virtual network embedding problem across multiple domains has been considered in [10], where it was proposed to use iterative local search (ILS) to partition the virtual request. For this problem, ILS starts with a random clustering, following which a sequence of solutions is generated by randomly remapping some of the nodes to other clusters. Of these solutions, the one that improves upon the current solution the most is kept, and the algorithm iterates until a stopping criteria is met. Despite the simplicity of this method, it is hard to guarantee the quality of the solution within a limited time. In a related study, a general procedure for resource allocation in distributed clouds was presented in [8]. The objective was to select the data centers, the racks, and processors with the minimum delay and communication costs, and then to partition the virtual nodes by mapping them onto the selected data center and processors.

In [25], a series of spectral partitioning algorithms are reviewed and summarized. These spectral partitioning techniques generally use the median of the Fiedler vector, the second smallest eigenvalue of the Laplacian matrix for the traffic matrix, to partition the graph into two parts. Then, by recursively applying this method, one can partition the graph into $2^n, n \geq 1$, parts. These algorithms have two limitations. First, the node weights related to load demands are not taken into consideration; consequently, the load may not be balanced well across the various clusters. Second, the number of clusters is limited to powers of two. In [26], a spectral partitioning based method that makes use of multiple eigenvectors of the Laplacian matrix is proposed. While this work considers the node weight balancing and makes use of multiple eigenvectors to produce a solution with a high-quality, it only partitions a graph into two, four or eight parts.

In contrast to the existing works, our algorithm exploits multiple eigenvectors of the Laplacian matrix to partition the network requests into arbitrary number of clusters. Such eigenvectors would form a k-dimensional space known as the eigenspace, and the Euclidean distance would reflect the traffic intensities. One can arrive at a high quality solution by clustering on Euclidean distance. Unlike Gomory–Hu tree based approaches, while assigning data points to the different partitions, the minimization of the inter-cluster traffic and the cluster capacities are jointly considered. This allows network requests to be partitioned into arbitrary clusters under the capacity constraints.

3. Virtual request partitioning

Virtual request partitioning is required in both the VNE and VMP problems, whereby the objective is to partition the virtual network into a set of clusters in order to minimize the inter-cluster traffic. Fig. 1(a) shows a set of virtual nodes that have been partitioned into three clusters such that inter-cluster traffic is minimum. In the VNE scenario of Fig. 1(b), mapping each of the clusters to a different domain will minimize inter-domain traffic (which presumably is more expensive than intra-domain traffic). In the context of the VMP problem in Fig. 1(c), assuming that each cluster is assigned to a different processor or even rack, optimal partitioning of the virtual request minimizes the traffic that has to be handled by the aggregate and core switches of the data center network, hence improve the scalability and stability of the network.

In this section, we formally define the virtual request partitioning problem which could be applied both to the VNE and VMP problem, and then present an algorithm based on spectral clustering to solve this problem.

3.1. Problem statement

We model the communication between virtual nodes as a traffic matrix $W = [w_{ij}]_{n\times n}$, where element w_{ij} represents the amount of traffic from virtual node i to j. Each virtual node is associated with a resource requirement r_i, and each cluster h is associated with a capacity threshold Cap_h.

With these definitions, partitioning the set of virtual nodes into k clusters so as to minimize the inter-cluster traffic can be formulated as the following Integer Linear Programming (ILP) problem:

$$\textbf{minimize} \quad \sum_k \sum_{i,j} w_{ij}(1 - y_{ij}^k) \tag{1}$$

$$\textbf{subject to} \quad \sum_i r_i x_i^k \leq Cap_k, \qquad \forall k \tag{2}$$

$$x_i^k + x_j^k \leq y_{ij}^k + 1 \qquad \forall i, j, k \tag{3}$$

$$y_{ij}^k \leq x_i^k \qquad \forall i, j, k \tag{4}$$

$$\sum_k x_i^k = 1, \qquad \forall i \tag{5}$$

$$x_i^k = \{0, 1\}, y_{ij}^k = \{0, 1\} \tag{6}$$

The binary variable $x_i^k \in \{0, 1\}$ here indicates if virtual node i is assigned to cluster k while binary variable $y_{ij}^k \in \{0, 1\}$ indicates if virtual nodes i and j are both mapped onto cluster k.

Constraint (2) ensures that the amount of resources assigned to each cluster will not exceed its capacity limit. Constraint (3) and constraint (4) guarantees consistency between decision variable x and y. Constraint (5) makes sure that virtual machine i is assigned to exactly one cluster. This formulation is equivalent to the (k, v)-balanced partitioning problem, which is an NP-complete problem [16]. We also note that by replacing "virtual node" with "VM" and cluster with "processor," the above formulation also expresses the