Journal of Computers Vol. 29 No. 1, 2018, pp. 40-46
doi:10.3966/199115992018012901004

# Mapping Virtual Tasks onto Physical Devices for Cloud Computing

Chi-Chou Kao

Department of Computer Science and Information Engineering, National University of Tainan
Tainan, Taiwan
cckao@mail.nutn.edu.tw

Received 9 September 2016; Revised 03 March 2017; Accepted 04 March 2017

**Abstract.** Cloud computing is a commercial infrastructure that eliminates the need for maintaining expensive computing facilities. In this paper, a virtual network is a set of nodes with edges that denote the communication bandwidth requirement between them, while a physical network denotes a set of physical nodes with edges that represent the available physical resources. Our goal is to map the virtual nodes of the physical nodes and find a physical resource allocation to meet the logical network demands, subject to physical network constraints. An efficient approach is proposed to achieve a feasible virtual to physical mapping. Experiments were conducted to demonstrate the efficiency of the proposed algorithms.

**Keywords**: assignment, cloud computing, flow, mapping, node

## 1 Introduction

The internet has been a great success in the past few decades and has provided a whole new way to access and exchange information. However, due to the multi-provider nature of the internet, adopting new network architecture requires not only changes in individual routers and hosts, but also joint agreements among ISPs. [1]. Generally, there are two approaches to deliver services in real life. One way is to buy new resources and build a new system according to the specific service requirements. Another way is to accomplish the required tasks using available resources in the existing system. The first way needs a great of cost so the second way is feasible in real life.

In a cloud computing environment, a virtual network could represent a set of virtual machines that must be deployed on two physical servers, where the virtual links denote the communication requirements between them. The physical network consists of the physical servers and physical communication links between them. The mapping problem then translates into assigning virtual machines onto physical servers and assigning flows in the physical network with bandwidth allocation to meet the virtual communication requirements [2-3].

In distributed computing literature, the problem that we consider is related to the problem of task/job allocation to processors [4-8]; here the goal is to assign tasks to processors to optimize various different performance metrics. In [4-5], the formulation involves processing costs for each task and communication costs for flows and the goal is to minimize the entire program completion time. In [6], the sum of execution and communication costs is minimized for a homogeneous network, while in [7], a heterogeneous network model is considered. The main difference of this literature from our problem is that we do not consider any specific optimization but rather explicitly consider linking capacity constraints on the physical network. We also address the important question of feasibility of mapping a given logical network onto a physical network by introducing feasibility checks based on logical physical graph cuts.

The load balancing problem on networks is a generalization of the load balancing problem on unrelated parallel machines [9]. A competitive strategy to minimize congestions in online virtual circuit routing was developed by Aspnes et al. to achieve a competitive ratio of $O(\log n)$ for permanent (i.e., infinite holding time) virtual circuits, where $n$ is the number of nodes in the network [10]. It was extended to the case of finite holding time circuits in [11]. The node selection and placement problem has also been studied in the context of web server replication and access network design [12-14]. Shi et al. investigated the server placement problem in overlay networks to ensure desired service quality to all its customers [15].

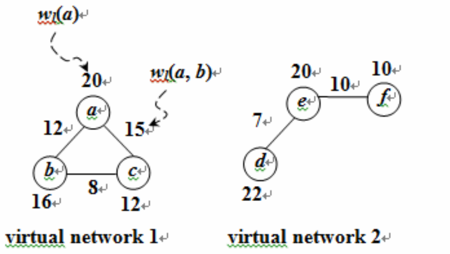
In the paper, we present an efficient approach to solve the mapping problem, namely, (1) solve for a feasible node assignment, and, (2) given the node assignment, solve for the flow assignment. In the first problem, we first present a clustering algorithm to create an augmented physical network. Next, according to the location requirement of the virtual nodes, every virtual network node is mapped onto a physical network node to satisfy capacity constraints. The latter problem can be solved using multi-commodity flow algorithms [16-17]. Experiments were conducted to demonstrate the efficiency of the proposed algorithms.

The remainder of the paper is organized as follows. In Section II we formulate the mapping problem and decompose it into a node assignment sub-problem and a flow assignment sub-problem. In Section III a novel node assignment algorithm is proposed. Section IV uses the multi-commodity flow algorithms to solve the flow assignment problem. Section V presents experimental results and finally Section VI concludes the paper.

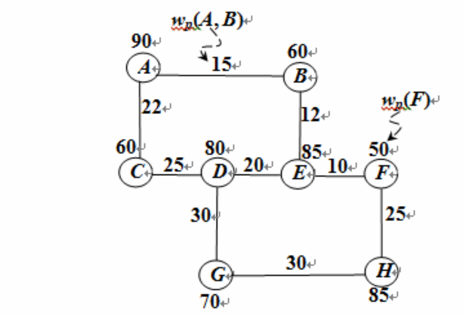
## 2 Terminology and Problem Formulation

A weighted undirected graph, $G_v = (V_v, E_v, W_v)$, which is used to represent given tasks, is a virtual network. A vertex in $V_v$ represents an implementation of a functional task and an undirected edge $e_v(i, j)$ exists in $E_v$ if there are communications between the task $i$ and the task $j$. In the virtual network, the communication requirement (bandwidth) between the task $i$ and the task $j$ is denoted by the edge weight $w_v(i, j) \in W_v$ and the storage requirement of the task $i$ is denoted by the weight $w_v(i) \in W_v$ of the node $v_v(i)$. The logical tasks need to be mapped onto a given real physical device (computers/processors), which is represented as a physical network. Let $G_p = (V_p, E_p, W_p)$ be a weighted undirected graph that represents the physical network. A vertex in $V_p$ represents a real physical device and an undirected edge $e_p(i, j)$ exists in $E_p$ if the physical device $i$ can be linked physically to the task $j$. Let $w_p(i, j) \in W_p$ denote the edge weight, which represents the communication capacity of a physical link from node $v_p(i)$ to node $v_p(j)$ in $E_p$. In the physical network, the storage capacity of the physical device $i$ is denoted by the weight $w_p(i) \in W_p$ of the node $v_p(i)$. Note that multiple logical nodes can be mapped into one physical node if the total storage requirements of the logical nodes do not exceed the storage capacity of the physical device. Fig. 1 illustrates a virtual and physical network. There are two virtual networks as shown in Fig. 1(a) Every virtual network is composed of three virtual tasks with communication requirements among them. It can be seen that there is a bandwidth requirement of 15Mbps between nodes $v_v(a)$ and $v_v(c)$ and a storage requirement of 20Mbytes for the task $a$. In Fig. 1(b), the physical network composed of eight physical devices $A$, $B$, $C$, $D$, $E$, $F$, $G$, and $H$. The communication capacities between the physical devices and the storage capacities are given in the edge weights and node weights, respectively.

The mapping problem is to map virtual nodes onto physical nodes such that the required communication and storage can be satisfied. In this paper, we allow that multiple virtual nodes can be accommodated with one physical node and allow also multi-path routes flow routing, subject to the link capacity limits in the physical network. Fig. 2 is mapping results of Fig. 1. The virtual nodes $v_v(a)$, $v_v(b)$, and $v_v(c)$ in the virtual network 1 are mapped onto the physical nodes $v_p(C)$, $v_p(H)$, and $v_p(B)$ in the given physical network, respectively. The virtual nodes $v_v(d)$, $v_v(e)$, and $v_v(f)$ in the virtual network 2 are mapped onto the physical nodes $v_p(A)$, $v_p(D)$, and $v_p(H)$ in the given physical network, respectively. Please note that since multiple logical nodes can be accommodated to one physical node and multi-path routing is allowed, the virtual nodes $v_v(b)$ and $v_v(f)$ can be mapped simultaneously onto the physical node $v_p(H)$and the communication requirements between $v_v(a)$ and $v_v(b)$ in the virtual network 1 and between $v_v(e)$ and $v_v(f)$ in the virtual network 2 are achieved with aggregated bandwidth of 22 Mbps, which is implemented to the path *CDGH* in the physical network. As a result, we say that the physical nodes $v_p(A)$, $v_p(B)$, $v_p(C)$, $v_p(D)$, and $v_p(H)$ are mapped nodes and $e_p(A, B)$, $e_p(B, E)$, $e_p(E, F)$, $e_p(F, H)$, $e_p(G, H)$, $e_p(D, G)$, $e_p(C, D)$, and $e_p(A, C)$ are mapped edges in Fig. 2.



(a) two given virtual networks



(b) a given physical network

**Fig. 1.** An example of weighted undirected graphs

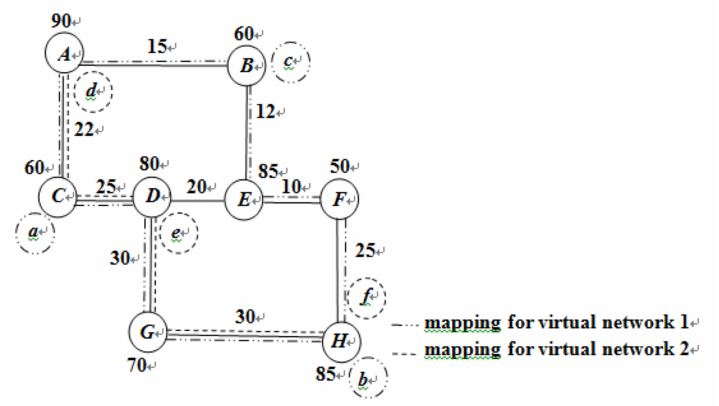


**Fig. 2.** Mapping results of Fig. 1

In the mapping problem, we need to find two solutions: (1) a node assignment solution to accommodate every virtual node with a physical node under capacity constraints and (2) a flow assignment solution to satisfy the communication constraints of the virtual requirements. The communication constraints reflect the fact that the aggregated flow passing anyone physical link cannot exceed the link capacity, i.e. the actual allocated aggregated bandwidth on the set of flows between a node pair in the physical network is larger than or equal to the virtual demand requirement between the corresponding node pair in the virtual network.

We can formulate mathematically the mapping problem. Assume that there are $x$ virtual nodes $v_v(1)$, $v_v(2)$, …, $v_v(x)$, which are mapped onto a physical node $v_p(i)$ in a node assignment solution. If this node assignment is feasible, the following equation (1) must be satisfied for any a mapped node, $v_p(i)$, in the physical network.

$$\sum_{k=1}^{x} w_v(k) \le w_p(i) \quad . \qquad \textbf{(1)}$$

Consider a undirected edge $e_p(i, j)$ in the physical network whose edge weight is $w_p(i, j)$ and there are $y$ virtual edge weight $w_{v1}$, $w_{v2}$, …, $w_{vy}$ allocated weight on them, where $w_{vy}$ represents the $y$-th logic edge weight. A flow assignment is feasible if the relations between physical link capacity and logical requirement constraints meet the following equation (2) for any a mapped edge, $e_p(i, j)$, in the physical network.

$$\sum_{k=1}^{y} w_{vk} \leq w_p(i,j) \ . \tag{2}$$

Accordingly, the mapping problem for cloud computing as follows.

*Given two weighted undirected graphs, which represent a virtual task network and a physical device network, respectively, find the feasible node and flow assignments to map the virtual network onto the physical network.*

## 3 Node Assignment Algorithm

In this section, a node assignment algorithm will be presented to generate a feasible node assignment, i.e. mapping to all nodes of the virtual network onto the nodes of physical network so that the capacity constraints are satisfied. We first present a clustering algorithm to create an augmented physical network. Next, according to the location requirement of the virtual nodes, every virtual network node is mapped onto a physical network node to satisfy capacity constraints.

### 3.1 Augmented Physical Network Construction

The objective of a clustering algorithm is to find groups of objects such that the objects have similar characteristics in the same group. In this subsection we present a clustering algorithm which produces an optimal number of clusters. A binary tree is a simple structure for representing binary relationship, and any connected components of tree is called *subtree*. Each cluster corresponds to one subtree, which does not overlap there presenting subtree of any other cluster. Clustering problem is equivalent to a problem of identifying these subtrees through solving a tree partitioning problem.

Given a point set $S$, a Minimum Spanning Tree (MST) [18] is first constructed from the points in $S$. The weight of the edge in the tree is the Euclidean distance between the two end points. Next the average weight $\hat{W}$ of the edges and its standard deviation $\sigma$ are computed; any edge with $W > \hat{W} + \sigma$ or current longest edge is removed from the tree. This leads to a set of disjoint subtrees $S_T = \{T_1, T_2 \ldots\}$. Each of these subtrees $T_i$ is treated as a cluster. The centers of clusters or regions are identified using eccentricity of points. These points are a representative point for the each subtree $S_T$. A point $c_i$ is assigned to a cluster $i$ if $c_i \in T_i$. The group of center points is represented as $C = \{c_1, c_2, \ldots, c_k\}$. These center points $c_1, c_2, \ldots, c_k$ are connected and again minimum spanning tree is constructed. This minimum spanning tree is used for finding optimal number clusters. A Euclidean distance between pair of clusters can be represented by a corresponding weighted edge. Our algorithm is also based on the minimum spanning tree but not limited to two-dimensional points. There were two kinds of clustering problem; one that minimizes the maximum intra-cluster distance and the other maximizes the minimum inter-cluster distances. Our algorithm produces clusters with intra-cluster similarity. Next, the algorithm converts the subtree/cluster into dendrogram. This algorithm uses both divisive as well as an agglomerative approach to find Dual similarity clusters. Since the subtrees are themselves are clusters, are further, classified in order to get more informative similarity clusters.

### 3.2 Node Assignment

The key idea of the node assignment algorithm is to select a cluster of nodes that are not only lightly loaded but also likely to result in low link stresses when they are connected. This is achieved through a coarse-to-fine approach: In the algorithm, a cluster center in the network is first identified based on the stress level in its neighborhood. The nodes are then selected sequentially based on their distances.

After identifying the cluster, the next step is to select the rest of the nodes. Rather than simply selecting nodes with minimum stresses, our node selection algorithm also considers the link stress by weighting the node stress based on its distances to other selected substrate nodes. We adopt the definition of distance from the shortest-distance path algorithm [19], which can effectively balance the network load among links and efficiently utilize network resources. The set of substrate nodes with the minimum potential are then selected.

After all nodes are determined, we need to map the virtual nodes into the selected nodes such that virtual nodes with higher degrees are mapped to nodes. The intuition behind this is that virtual nodes with higher degree will set up more virtual links. All of these virtual links will go through some of the physical link in the neighborhood of the corresponding physical node. Therefore, the above matching tends to reduce the congestion and balance the link stress.

## 4 Flow Assignment Algorithm

In the multi-commodity flow (MCF) problem [4], each commodity ($s$, $t$, $d$) is composed of a source node $s$, a destination node $t$ and a demand $d$. The objective of this problem is to minimize the cost of routing a set of commodities simultaneously in $G$. One variation of the MCF problem is the maximum-concurrent flow problem. In this problem, the objective is to maximize the scaling factor $f$, such that for each commodity $i$, the $f$*$d$ amount of demand can be routed simultaneously. The solution to the maximum-concurrent flow problem, if one exists, gives the largest possible value of the scaling factor $f$ and the flow placement of each commodity on the graph. The commodity placement is a set of paths and it discloses the amount of demand for each commodity that should be placed on each graph edge in order to achieve the optimal $f$ value.

In the flow assignment problem, we are first to reserve a fixed amount of bandwidth exclusively for each pair of network edge nodes. The amount of bandwidth to be reserved and along which path(s) for a network edge pair is based on the solution to the corresponding maximum-concurrent flow problem. More precisely, suppose that $d$ is set to 1, $f$ amount of bandwidth will be reserved for a network edge pair ($s$, $t$) along the obtained commodity placement for the commodity ($s$, $t$, $d$). Next, we attempt to find a viable path for each pair of nodes using the just reserved bandwidth for that network edge pair. The path searching process is attained by applying the least-cost routing algorithm, which determines the least cost route with link cost being defined as the reciprocal of link bandwidth allocation percentage. If a feasible path cannot be found, then our scheme will attempt to run least-cost routing on the reserved resources plus the currently unallocated network resources. It is important to note that the first step is necessary only when there is a topology change or when a new edge pair is identified since the last time this step is performed. Otherwise only the second step is needed to determine a feasible assign.

## 5 Experimental Results

In this section, we first describe the evaluation environment, and then present our evaluation results. We have implemented a discrete event simulator to evaluate the performance of our algorithms which is freely available at [20]. The physical network topologies in our experiments are randomly generated with 50 nodes using the GT-ITM tool [21] in grids. Each pair of physical nodes is randomly connected with probability 0.5. The CPU and bandwidth resources of the physical nodes and links are real numbers uniformly distributed between 50 and 100. In each virtual network request, the number of virtual nodes is randomly determined by a uniform distribution between 2 and 10 following similar setups to previous works [22]. The CPU requirements of the virtual nodes are real numbers uniformly distributed between 0 to 20 and the bandwidth requirements of the virtual links are uniformly distributed between 0 and 50.

We use several performance metrics for evaluation purposes in our experiments. We measure the average acceptance ratio, revenue, and provisioning costs for virtual network requests over time. We also measure the average node utilization and average link utilization of the substrate network. In all these cases we plot the performance metrics against time to show how each of the G-MCF [21] and D-ViNE-SP [22] algorithms actually perform in the long run.

Fig. 3 and Fig. 4 depict the average utilization of physical nodes and physical links for different algorithms. It can be seen that the proposed algorithm has the highest node and link utilization. We believe that the reason behind this is the distributed nature of the proposed algorithm.

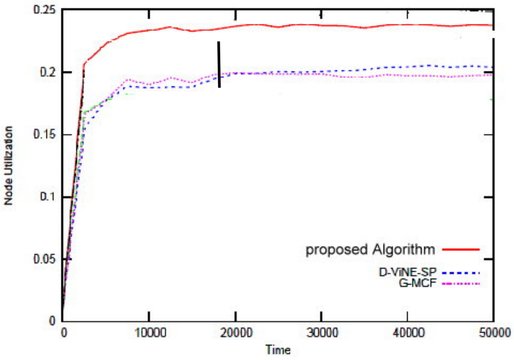


**Fig. 3.** Average node utilization

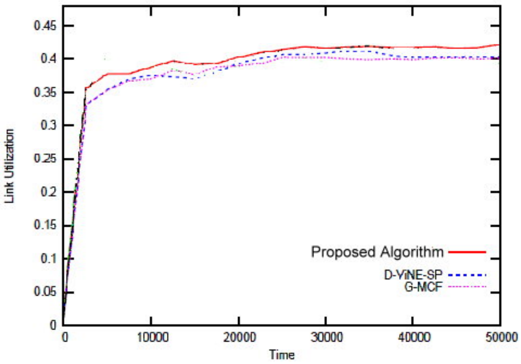


**Fig. 4.** Average link utilization

## 6 Conclusions

In this paper, we study the problem of mapping a virtual network on to a physical network in cloud computing. We set a set of nodes with edges that denote the communication bandwidth requirement between the machines in the virtual network and a set of physical nodes with edges that represent the available physical resources in a physical network. The goal is to map the virtual nodes of the physical nodes and find a physical resource allocation to meet the virtual network demands subject to physical network constraints. An efficient approach is proposed to achieve a feasible virtual to physical mapping.

## Acknowledgements

This work was supported in part by the National Science Council, R.O.C., under Grant NSC 101-2221-E-024-020.

## References

[1] B. Han, V. Gopalakrishnan, L. Ji, S. Lee, Network function virtualization: Challenges and opportunities for innovations, IEEE Commun. Mag. 53(2)(2015) 90-97.

[2] Q. Duan, Y. Yan, A.V. Vasilakos, A survey on service-oriented network virtualization toward convergence of networking and cloud computing, IEEE Trans. Netw. Serv. Manag. 9(4)(2012) pp. 373-392.

[3] J. Botero, X. Hesselbach, A. Fischer, H. de Meer, Optimal mapping of virtual networks with hidden hops, Telecommunication Systems 51(4)(2011) pp. 1-10.

[4] K. Pentikousis, C. Meirosu, D.R. Lopez, S. Denazis, K. Shiomoto, F.J. Westphal, Network and service virtualization, IEEE

Commun. Mag. 53(4)(2015) 88-89.

[5] F. Paganelli, M. Ulema, B. Martini, Context-aware service composition and delivery in NGSONs over SDN, IEEE Commun. Mag. 52(8)(2014) 97-105.

[6] ETSI. Network Functions Virtualisation (NFV), Management and Orchestration, GS NFV-MAN 001 V1.1.1; ETSI: Sophia Antipolis Cedex, France, 2014.

[7] R.V. Rosa, M.A. Silva Santos, C.E. Rothenberg, MD2-NFV: The case for multi-domain distributed network functions virtualization, in: Proc. the IEEE 2015 International Conference and Workshops on Networked Systems (NetSys), 2015.

[8] V. Lopez, O.G. de Dios, B. Fuentes, M. Yannuzzi, J.P. Fernández-Palacios, D. Lopez, Towards a network operating system, in: Proc. the 2014 Optical Fiber Communication Conference, 2014.

[9] Y. Azar, A.Z. Broder, A.R. Karlin, On-line load balancing, in: Proc. IEEE Symposium Foundations of Computer Science, 1992.

[10] J. Aspnes, Y. Azar, A. Fiat, S. Plotkin, O. Waarts, On-line load balancing with applications to machine scheduling and virtual circuit routing, in: Proc. ACM symposium on Theory of computing, 1993.

[11] Y. Azar, B. Kalyanasundaram, S.A. Plotkin, K. Pruhs, O. Waarts, Online load balancing of temporary tasks, in: Proc. Workshop on Algorithms and Data Structures, 1993.

[12] D. Bertsekas, R. Gallager, Data Networks, 2nd ed., Prentice-Hall, Upper Saddle River, NJ, 1992.

[13] M. Sayal, Y. Breitbart, P. Scheuermann, R. Vingralek, Selection algorithms for replicated web servers, SIGMETRICS Perform. Eval. Rev. 26(3)(1998) 44-50.

[14] P. Krishnan, D. Raz, Y. Shavitt, The cache location problem, IEEE/ACM Trans. Networking 8(5)(2000) 568-582.

[15] S. Shi, J. Turner, Placing servers in overlay networks, in: Proc. International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPETS), 2002.

[16] B. Naudts, W. Tavernier, S. Verbrugge, D. Colle, M. Pickavet, Deploying SDN and NFV at the speed of innovation: toward a new bond between standards development organizations, industry fora, and open-source software projects, IEEE Commun. Mag. 54(3)(2016) 46-53.

[17] S. John Peter, S.P. Victor, Constraint-free optimal dual similarity validity clusters using dynamic minimum spanning tree, International Journal of Computer Science and Network Security 10(7)(2010) 253-261.

[18] Q. Ma, P. Steenkiste, On path selection for traffic with bandwidth guarantees, in: Proc. IEEE ICNP, 1997.

[19] ViNE-Yard. <http://www.mosharaf.com/ViNE-Yard.tar.gz>.

[20] E. Zegura, K. Calvert, S. Bhattacharjee, How to model an Internetwork, in: Proc. IEEE INFOCOM, 1996.

[21] M. Yu, Y. Yi, J. Rexford, M. Chiang, Rethinking virtual network embedding: Substrate support for path splitting and migration, ACM SIGCOMM Computer Communication Review 38(2)(2008) 17-29.

[22] M. Chowdhury, M.R. Rahman, R. Boutaba, ViNEYard: virtual network embedding algorithms with coordinated node and link mapping, IEEE/ACM Trans. Network 20(1)(2012) 206-219.