Security-aware Application Placement in a Mobile Micro-cloud

Shiqiang Wang*, Murtaza Zafer†, Kin K. Leung*, and Ting He†
*Department of Electrical and Electronic Engineering, Imperial College London, SW7 2AZ, United Kingdom
†IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, United States
Email: shiqiang.wang11@imperial.ac.uk, mzafer@us.ibm.com, kin.leung@imperial.ac.uk, the@us.ibm.com

***Abstract*—In military coalition operations, it is important to deliver real-time situational awareness (SA) to units on the ground, and this can be enabled through a mobile micro-cloud which utilizes a combination of computing resources at the edge and the core of a tactical hybrid network to run SA applications. One of the challenges in a mobile micro-cloud is to efficiently place application requests onto the computing resources, taking coalition security constraints into account. Towards this end, in this paper, we study security-aware placement of an application. The application is described as a graph with nodes denoting modules that have communication demands between them, and the physical network is a graph with nodes denoting servers with computing resources that are connected by communication links. We abstract the security requirements as domain constraints between nodes in application graph and nodes in physical graph that restrict the set of physical nodes each application node can be mapped to, as well as conflict constraints among nodes in the application graph that restrict the set of application nodes that can be mapped to the same physical node. Based on this abstraction, we formulate the problem as a mixed-integer linear program, and discuss possible techniques of solving it. The proposed work can be regarded as a first step towards a general framework for efficient application placement in a mobile micro-cloud.**

I. INTRODUCTION

Security-aware application placement in cloud computing environments is an interesting open issue [1]. Its importance particularly arises in the military context. In this paper, we focus on a mixed-integer linear program formulation of the application placement problem. Compared with existing work such as [2], the main contribution of our work is to incorporate security constraints into an optimization framework. For a more comprehensive introduction to the background and related work, please refer to [3].

II. PROBLEM MODEL

A. Application Graph and Physical Graph

We consider security-aware placement of applications onto the physical cloud environment. The application is modeled as a graph $G^A = (V, E)$, in which the nodes V represent the modules in the application, and the edges E represent the communication demand between nodes. Each node $v \in V$ is associated with a demand d_v which represents the computation resource demand of node v. Similarly, each edge $e \in E$ is associated with a communication bandwidth demand d_e. The physical cloud is modeled with a graph $G^P = (N, L)$, where the nodes N are network elements such as servers, routers, etc. and the edges L are communication links between nodes. Each node $n \in N$ has a total capacity c_n^t and an existing demand d_n^p (i.e. the total demand of previously assigned nodes), and each edge $l \in L$ also has a total capacity c_l^t and an existing demand d_l^p. For nodes in the physical network that do not have capability of hosting application modules (such as routers), we set $c_n^t = d_n^p = 0$. The graphs G^A and G^P can be either directed or undirected depending on practical scenarios. In this paper, we consider the case where G^A is a directed graph and G^P is an undirected graph.

B. Security Constraints

We consider two types of security constraints. The first type is a restriction on the set of physical nodes that a particular application node can be mapped to, as shown in Fig. 1(a). This resembles the case where different modules of the application must be executed by servers that belong to different organizations. We define $N_m(v) \subseteq N$ for $\forall v \in V$ as the subset of physical nodes that v can be mapped to. The second type is a conflict constraint which says that some application nodes cannot be mapped onto the same physical node, because they may pose potential security risks to each other, as shown in Fig. 1(b). We define $V_c(v) \subseteq V$ for $\forall v \in V$ as the conflict set of v, which means that application nodes in $V_c(v)$ cannot be mapped to the same physical node as v. Similar domain and conflict constraints can be defined for edges, but we omit this part in this paper due to space limitations.

C. Variables in Application Placement

The flow variables $f_{e\rightarrow(n_1,n_2)}$ for $\forall e \in E$ and $\forall n_1, n_2 \in N$ that have a communication link in between denotes the amount of data sent from n_1 to n_2, for the application edge e. The binary variables $x_{v\rightarrow n}$ for $\forall v \in V, \forall n \in N_m(v)$ indicates whether v is mapped to n. Auxiliary variables $y_{v\rightarrow n}$ for $\forall v \in V, \forall n \in N_m(v)$ indicates whether a conflicting node of v is

---

This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the author(s) and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

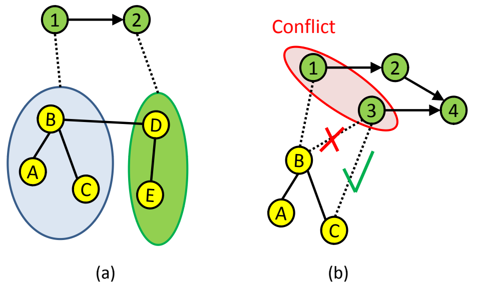

Figure 1. Security constraints: (a) domain constraint – application node 1 can only be mapped to physical nodes A, B, and C and application node 2 can only be mapped to physical nodes D and E; (b) conflict constraint – application nodes 1 and 3 cannot be mapped onto the same physical node.

mapped to n, which is used to assist the optimization problem formulation.

III. OPTIMIZATION OF APPLICATION PLACEMENT

To jointly consider the load balancing of servers and communication links, as well as minimizing the number of hops for communications, we consider the following objective function:

$$\begin{aligned}\min \Bigg\{ & \max_{n\in N} \frac{\alpha_n}{c_n^t}\left(d_n^p + \sum_{v\in V} d_v x_{v\to n}\right)\\ & + \max_{l\in L}\frac{\beta_l}{c_l^t}\left(d_l^p + \sum_{e\in E}\left(f_{e\to(n_1,n_2)} + f_{e\to(n_2,n_1)}\right)\right)\\ & + \sum_{l\in L}\beta'_l \sum_{e\in E}\left(f_{e\to(n_1,n_2)}+f_{e\to(n_2,n_1)}\right)\Bigg\},\end{aligned} \tag{1}$$

where the edge l connects n_1 and n_2, and α_n, β_l, and β'_l are weighting factors. This objective function can be rewritten as a linear objective function, by adding two additional constraints for the maximum operations.

The node and edge capacity constraints are the following:

$$\sum_{v\in V} d_v x_{v\to n} \le c_n^t - d_n^p, \forall n \in N, \tag{2}$$

$$\sum_{e\in E}\left(f_{e\to(n_1,n_2)} + f_{e\to(n_2,n_1)}\right) \le c_l^t - d_l^p, \forall l \in L, \tag{3}$$

where n_1 and n_2 are connected via l. The flow conservation constraint is:

$$\begin{aligned}&\sum_{n_2:(n_1,n_2)\in L} f_{e\to(n_1,n_2)} - \sum_{n_2:(n_1,n_2)\in L} f_{e\to(n_2,n_1)}\\ &= d_e\left(x_{v_1\to n_1} I_{n_1\in N_m(v_1)} - x_{v_2\to n_1} I_{n_1\in N_m(v_2)}\right),\\ &\forall n_1 \in N, \forall e \in E,\end{aligned} \tag{4}$$

where e is the directed application edge from v_1 to v_2, the indicator I indicates whether the condition in its subscript is satisfied. The following constraint guarantees that each application node is mapped to one physical node:

$$\sum_{n\in N_m(v)} x_{v\to n} = 1, \forall v \in V. \tag{5}$$

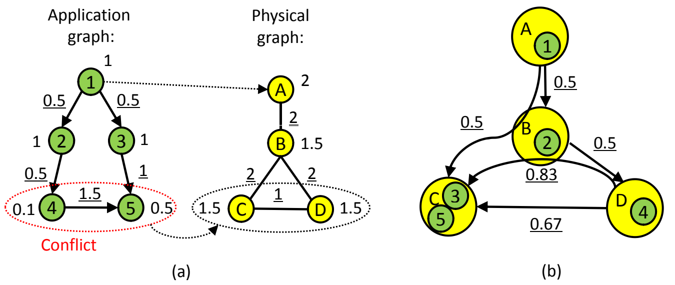

Figure 2. Example of application placement: (a) problem setting, (b) mapping result. In (a), the numbers besides nodes and edges in the application graph are resource demands, and the numbers in the physical graph are capacity values, the underlined numbers correspond to edge values. In (b), the underlined numbers are the data flow amount on the corresponding communication links.

The node conflict constraints are:

$$y_{v\to n} \le \sum_{v_1\in V_c(v)} x_{v_1\to n}, \forall v \in V, \forall n \in N, \tag{6}$$

$$y_{v\to n} \ge \frac{\sum_{v_1\in V_c(v)} x_{v_1\to n}}{|V|}, \forall v \in V, \forall n \in N, \tag{7}$$

$$x_{v\to n} + y_{v\to n} \le 1, \forall v \in V, \forall n \in N. \tag{8}$$

We also require that $f_{e\to(n_1,n_2)} \ge 0$, $x_{v\to n} = \{0,1\}$, and $y_{v\to n} = \{0,1\}$.

IV. SOLUTION

Our optimization problem is a mixed-integer linear program, which can be solved with IBM CPLEX [4], or OPTI Toolbox [5], etc. Fig. 2 shows an example scenario and its mapping result which has been obtained with OPTI Toolbox. In the example, application node 1 can only be mapped to physical node A, and application nodes 4 and 5 can only be mapped to physical nodes C and D but not on the same node. We set $\alpha_n = \beta_l = \beta'_l = 1$ and $d_n^p = d_l^p = 0$, the demands and capacities are indicated in Fig. 2(a) and the mapping result is shown in Fig. 2(b).

V. CONCLUSION

In this paper, we have proposed an optimization framework for application placement in cloud environments with security constraints. The problem is a mixed-integer linear program which can be solved with general optimization tools. Future work will focus on developing approximation algorithms for more efficient solution in large scale networks.

REFERENCES

[1] A. Fischer, J. Botero, M. Beck, H. De Meer, and X. Hesselbach, “Virtual network embedding: A survey,” *IEEE Commun. Surveys Tuts.*, Feb. 2013, accepted.

[2] M. Chowdhury, M. Rahman, and R. Boutaba, “Vineyard: Virtual network embedding algorithms with coordinated node and link mapping,” *IEEE/ACM Trans. Netw.*, vol. 20, no. 1, pp. 206–219, 2012.

[3] S. Wang, G.-H. Tu, R. Ganti, T. He, K. Leung, H. Tripp, K. Warr, and M. Zafer, “Mobile micro-cloud: Application classification, mapping, and deployment,” in *Proc. of Annual Fall Meeting of ITA (AMITA) 2013*, Oct. 2013.

[4] IBM CPLEX Optimizer. [Online]. Available: http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/

[5] OPTI Toolbox. [Online]. Available: http://www.i2c2.aut.ac.nz/Wiki/OPTI/