The Journal of Systems and Software 120 (2016) 89–104

Contents lists available at ScienceDirect

The Journal of Systems and Software

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

Network-aware embedding of virtual machine clusters onto federated cloud infrastructure

Atakan Aral*, Tolga Ovatman

Department of Computer Engineering, Istanbul Technical University, Istanbul, Turkey

ARTICLE INFO

Article history:
Received 18 January 2016
Revised 4 June 2016
Accepted 6 July 2016
Available online 9 July 2016

Keywords:
Cloud computing
Federated cloud
Infrastructure as a service
VM cluster embedding
Resource allocation
Subgraph isomorphism

ABSTRACT

Federated clouds are continuously developing as the demands of cloud users get more complicated. Contemporary cloud management technologies like Open-Stack (Sefraoui et al., 2012) and OpenNebula (Milojičić et al., 2011) allow users to define network topologies among virtual machines that are requested. Therefore, federated clouds currently face the challenge of network topology mapping in addition to conventional resource allocation problems. In this paper, topology based mapping of virtual machine clusters onto the federated cloud infrastructures is studied. A novel algorithm is presented to perform the mapping operation that work towards minimizing network latency and optimizing bandwidth utilization. To realize and evaluate the algorithm, a widely used cloud simulation environment, CloudSim (Calheiros et al., 2011), is extended to support several additional capabilities in network and cost modeling. Evaluation is performed by comparing the proposed algorithm to a number of conventional heuristics such as least latency first and round-robin. Results under different request characteristics indicate that the proposed algorithm performs significantly better than the compared conventional approaches regarding various QoS parameters such as inter-cloud latency and throughput.

1. Introduction

Magnitude of the digital data being generated and the speed at which it is aggregating in cloud is enormous. In the not so distant future, even the largest Infrastructure as a Service (IaaS) providers may run into a difficulty in scalability as a result of this enormous increase in cloud service usage. Moreover, cloud users access the data from all around the world which makes it increasingly hard to provide a globally consistent Quality of Service (QoS). Federated cloud (Rochwerger et al., 2009; Buyya et al., 2010) is motivated by these dangers and defined as the mechanisms, policies and technologies to coordinate and unite cloud data centers even if they are managed by different vendors. As distinct from multi-cloud where multiple independent clouds are utilized by an application uninformedly, cloud providers voluntarily collaborate in the federated cloud scenario (Grozev and Buyya, 2014).

Federated clouds allow vendors to dispatch Virtual Machine (VM) requests to the other members of the federation, delivering the infinite scalability promise of cloud computing (Rochwerger et al., 2009). This improves the QoS by giving cloud vendors the ability to cope with demand peaks as well as to provide complete geographical coverage. Additionally, such an interoperability at the infrastructure level sets cloud users free of vendor lock-in and allows private data center owners to easily hybridize. Finally and more importantly for our work, with federated cloud it is possible to scale VM clusters across multiple vendor clouds (Buyya et al., 2010). Here, a VM cluster is a group of collaborating VMs that constitute a cloud service. It is a common practice to isolate different components of a service (e.g. storage, application logic, user interface) using distinct VMs that communicate among themselves.

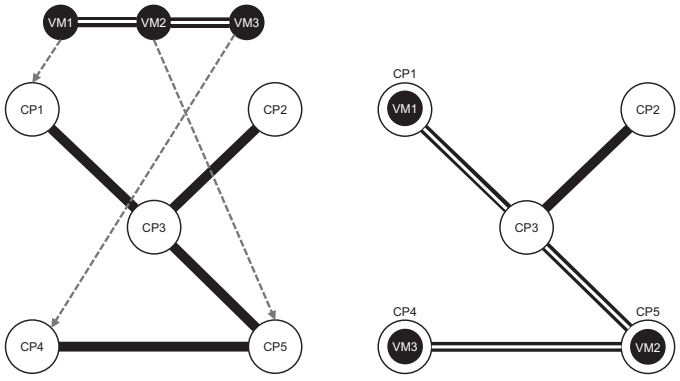
From the point of a cloud based service provider (and an IaaS user), deployment of cooperating VMs on different clouds paves the way for the following advantages.

- **Availability and disaster recovery.** The effect of a failure or low QoS in a cloud vendor can be easily compensated with minimal damage to the overall service.
- **Geographical coverage.** Geographically distributed user base of the service can be covered with a high QoS.
- **No vendor lock-in.** VMs can be migrated easily and quickly between vendors in case of any dissatisfaction.
- **Cost reduction.** Different pricing policies of the vendors can be exploited to reduce infrastructure cost.

However, distributed placement of VMs onto a federated cloud infrastructure also presents new problems that need to be addressed. One of the most significant of these problems is finding

* Corresponding author.
E-mail addresses: aralat@itu.edu.tr (A. Aral), ovatman@itu.edu.tr (T. Ovatman).

http://dx.doi.org/10.1016/j.jss.2016.07.007
0164-1212/



(a) Mapping (b) Deployment

Fig. 1. VM cluster embedding (Aral and Ovatman, 2015).

an efficient mapping between the physical topology and the user requests in the form of virtual topologies (Pittaras et al., 2015). Here, virtual topology defines the bandwidth requirements for data flows between VMs in the same cluster. When a service provider requests a VM cluster, it characterizes VM capacities as well as the amount of data that will be transferred between each VM pair in terms of bandwidth. Such a virtual topology can be defined via contemporary cloud management technologies like OpenStack (Sefraoui et al., 2012) and OpenNebula (Milojičić et al., 2011). On the other hand, physical topology defines the available dedicated network connections between cloud providers with their bandwidth capacities and latencies. Not all cloud provider pairs in the federation may have direct dedicated connections and not all VM pairs in a cluster need to communicate, thus neither of the topologies are complete graphs in general. When adjacent VMs are mapped to nonadjacent clouds, the connection has to be multi-hop.

Fig. 1 visualizes the mapping and deployment of a single VM cluster of three VMs onto a federation of five cloud providers (CP). Here, physical topology is represented with white circles (clouds) and thick lines (inter-cloud network connections) while virtual topology is represented with black circles (VMs) and double lines (data flows). According to the requested virtual topology, data will flow between pairs VM1 – VM2 and VM2 – VM3 but not VM1 – VM3. Fig. 1(a) demonstrates an example mapping between VMs and clouds. Different mappings can be generated via optimization algorithms with different objective functions, however the mapping relation must satisfy the function property (each VM must be mapped to exactly one output). In Fig. 1(b) VMs are dispatched to clouds according to the mapping in Fig. 1(a) and deployed there. During the execution, data transfer between VM2 and VM3 will be direct, while it will be through CP3 for VM1 and VM2. In a real world scenario with non-trivial number of VM clusters, multiple VMs belonging to different clusters would be hosted at each cloud.

VM cluster embedding (VMCE) problem deals with finding a mapping between inter-connected VMs and clouds, as exemplified by Fig. 1(a). The problem is not trivial due to multiple constraints and objectives present (Papagianni et al., 2013). First of all, clouds have limited and heterogeneous capacities in terms of CPU, memory and storage. Similarly, network connections have varied latencies and bandwidth capacities. VMs of different sizes should be placed on clouds respecting such limits and making an efficient use of the resources to increase utilization. A similar problem is referred as Virtual Network Embedding (VNE) in the literature (Fischer et al., 2013; Pittaras et al., 2015; Zhang et al., 2015). Our definition of VMCE diverges from VNE as it also involves constraints and requirements for nodes (cloud/VM) in addition to the edges (network). A mathematical definition of the VMCE problem is provided in Section 3.1.2.

Network plays a key role in the performance of distributed cloud services. Hence, communicating VMs should be placed on clouds that have low latency inter se for high QoS. Another factor is the latency between the user base and selected clouds. In the case of provisional applications such as scientific calculations or MapReduce (Dean and Ghemawat, 2008) jobs, high latency also extends the execution time and accordingly increases resource costs. Better latency optimization is vital for distributed, soft real-time services and applications (e.g. video streaming, online gaming) to run on federated cloud. Cloud computing may find a new area of application in real-time software provided that the network related challenges are overcome (García-Valls et al., 2014).

Major contributions of this paper can be listed as follows.

- A novel VMCE algorithm for federated cloud, Topology Based Mapping (TBM), is proposed. TBM employs a graph theoretical approach in combination with greedy heuristics. Main objectives of the TBM are to reduce network latency and optimize bandwidth utilization.
- RalloCloud, a framework for the realistic simulation of resource allocation in federated cloud as an extension to CloudSim (Calheiros et al., 2011), is presented. It includes topology, network and cost modeling as well as several performance criteria.
- Evaluation of the TBM algorithm as well as baseline heuristics in terms of latency, execution time, throughput, cost, rejection rate, etc. is performed.

TBM algorithm mainly focuses on the bandwidth and latency that is (1) within the VM cluster, and (2) between the VM cluster and the intermediate cloud user who submits it (e.g. a cloud-based service provider or a scientist running a high performance job). Optimization of the latency and bandwidth between the VM cluster and the geographically distributed end users is beyond the scope of this work. Reader may refer to works on replica placement (e.g. Smaragdakis et al. (2014); Xu and Li (2013)) for this kind of optimization.

A preliminary version of the TBM algorithm is published as a work-in-progress paper (Aral and Ovatman, 2015). Here, we present the complete version of the algorithm and extended evaluation results. Improvements to the TBM algorithm includes validity consideration (see Section 3.1.2) while the extended evaluation includes new evaluation parameters and heterogeneous infrastructure capacities (see Section 5.1). Consequently, TBM significantly outperforms its preliminary version in Aral and Ovatman (2015). Moreover, RalloCloud framework is introduced for the first time in this article.

The rest of this paper is organized as follows. Section 2 summarizes the relevant literature. Respectively in Sections 3 and 4, suggested framework (RalloCloud) and algorithm (TBM) are introduced. Section 5 presents evaluation results and their discussion. We conclude the paper in Section 6.

2. Related work

2.1. Graph and subgraph equivalence

Two graphs are called isomorphic if a bijective function that pairs vertices of one graph to the vertices of the other graph with edge preserving property can be defined. Edge preserving property states that two adjacent vertices of one graph can be paired to two vertices in the other if and only if they are adjacent as well. Instead, if a subdivision of a graph is isomorphic to a subdivision of another graph, then these graphs are called homeomorphic. A subdivision of a graph can be generated by replacing an edge with a