Computer Networks 91 (2015) 508–527

Contents lists available at ScienceDirect

Computer Networks

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

Network-aware virtual machine placement in cloud data centers with multiple traffic-intensive components

Amir Rahimzadeh Ilkhechi [a,*], Ibrahim Korpeoglu [b], Özgür Ulusoy [c]

[a] Computer Engineering Department, Duke University, 1505-Duke University Road, Apt #4k, Durham, North Carolina, ZIP code: 27701, United States
[b] Computer Engineering Department, Bilkent University, office at Engineering EA-401, Ankara, Turkey
[c] Computer Engineering Department, Bilkent University, office at Engineering EA-402, Ankara, Turkey

ARTICLE INFO

Article history:
Received 27 November 2014
Revised 14 August 2015
Accepted 27 August 2015
Available online 11 September 2015

Keywords:
Cloud computing
Virtual machine placement
Sink node
Predictable flow
Network congestion

ABSTRACT

Following a shift from computing as a purchasable product to computing as a deliverable service to consumers over the Internet, cloud computing has emerged as a novel paradigm with an unprecedented success in turning utility computing into a reality. Like any emerging technology, with its advent, it also brought new challenges to be addressed. This work studies network and traffic aware virtual machine (VM) placement in a special cloud computing scenario from a provider's perspective, where certain infrastructure components have a predisposition to be the endpoints of a large number of intensive flows whose other endpoints are VMs located in physical machines (PMs). In the scenarios of interest, the performance of any VM is strictly dependent on the infrastructure's ability to meet their intensive traffic demands. We first introduce and attempt to maximize the total value of a metric named "satisfaction" that reflects the performance of a VM when placed on a particular PM. The problem of finding a perfect assignment for a set of given VMs is NP-hard and there is no polynomial time algorithm that can yield optimal solutions for large problems. Therefore, we introduce several off-line heuristic-based algorithms that yield nearly optimal solutions given the communication pattern and flow demand profiles of subject VMs. With extensive simulation experiments we evaluate and compare the effectiveness of our proposed algorithms against each other and also against naïve approaches.

1. Introduction

The problem of appropriately placing a set of Virtual Machines (VMs) into a set of Physical Machines (PMs) in distributed environments has been an important topic of interest for researchers in the area of cloud computing. The proposed approaches often focus on various problem domains with different objectives: initial placement [1–3], throughput maximization [4], consolidation [9,10], Service Level Agreement (SLA) satisfaction versus provider operating costs minimization [11], etc. [5]. Mathematical models are often used to formally define the problems of that category. Then, they are normally fed into solvers operating based on different approaches including but not limited to greedy, heuristic-based or approximation algorithms. There are also well-known optimization tools such as CPLEX [12], Gurobi [15] and GLPK [17] that are predominantly utilized in order to solve placement problems of small size.

There is also another way of classifying the works related to VM placement based on the number of cloud environments: *Single-cloud* environments and *Multi-cloud* environments.

* Corresponding author. Tel.: +1 9196997984.
E-mail addresses: ilkhechi@cs.duke.edu (A.R. Ilkhechi), korpe@cs.bilkent.edu.tr (I. Korpeoglu), oulusoy@cs.bilkent.edu.tr (Ö. Ulusoy).

http://dx.doi.org/10.1016/j.comnet.2015.08.042

The first category is mostly concerned with service to PM assignment problems which are often NP-hard in complexity. That is, given a set of PMs and a set of services that are encapsulated within VMs with fluctuating demands, design an online placement controller that decides how many instances should run for each service and also where the services are assigned to and executed in, taking into account the resource constraints. Several approximation approaches have been introduced for that purpose including the algorithm proposed by Tang et al. in [16].

The second category, namely the VM placement in multiple cloud environments, deals with placing VMs in numerous cloud infrastructures provided by different Infrastructure Providers (IPs). Usually, the only initial data that is available for the Service Provider (SP) is the provision-related information such as types of VM instances, price schemes, etc. Without any information about the number of physical machines, the load distribution, and other such critical factors inside the IP side mostly working on VM placement across multi-cloud environments are related to cost minimization problems. As an example of research in that area, Chaisiri et al. [18] propose an algorithm to be used in such scenarios to minimize the cost spent in each placement plan for hosting VMs in a multiple cloud provider environment.

To begin with, our work falls into the first category that pertains to single cloud environments. Based on this assumption, we can take the availability of detailed information about VMs and their profiles, PMs and their capacities, the underlying interconnecting network infrastructure and all related for granted. Moreover, we concentrate on network rather than data center/server constraints associated with VM placement problem.

This paper introduces nearly optimal placement algorithms that map a set of virtual machines (VMs) into a set of physical machines (PMs) with the objective of maximizing a particular metric (named *satisfaction*) which is defined for VMs in a special scenario. The details of the metric and the scenario are explained in Section 3 while also a brief explanation is provided below. The placement algorithms are offline and assume that the communication patterns and flow demand profiles of the VMs are given. The algorithms consider network topology and network conditions in making placement decisions.

Imagine a network of physical machines in which there are certain nodes (physical machines or connection points) that virtual machines are highly interested in communicating. We call these special nodes “sinks”, and call the remaining nodes “Physical Machines (PMs)”. Despite the fact that sink usually is a receiver node in networks, we assume that flows between VMs and sinks are bidirectional.

As illustrated in Fig. 1, assuming a general unstructured network topology, some small number of nodes (shown as cylinder-shaped components) are functionally different than the rest. With a high probability, any VM to be placed in the ordinary PMs will be somehow dependent on at least one of the sink nodes shown in the figure. By dependence, we mean the tendency to require massive end-to-end traffic between a given VM and a sink that the VM is dependent on. With that definition, the intenser the requirement is, the more dependent the VM is said to be.

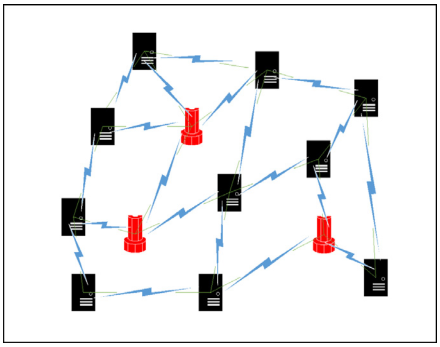

Fig. 1. Interconnected physical machines and sink nodes in an unstructured network topology.

The network connecting the nodes can be represented as a general graph $G(V, E)$ where E is the set of links, V is the set of nodes and S is the set of sinks (note that $S \in V$). On the other hand, the number of normal PMs is much larger than the number of sinks (i.e. $|S| \ll |V - S|$).

Each link consisting of end nodes u_i and u_j is associated with a capacity c_{ij} that is the maximum flow that can be transmitted through the link.

Assume that the intensity of communication between physical machines is negligible compared to the intensity of communication between physical machines and sinks. In such a scenario, the quality of communication (in terms of delay, flow, etc.) between VMs and the sinks is the most important factor that we should focus on. That is, placing the VMs on PMs that offer a better quality according to the demands of the VMs is a reasonable decision. Before advancing further, we suppose that the following a priori information is given about any VM:

- *Total Flow*: the total flow intensity that the VM will demand in order to achieve perfect performance (for sending to and/or receiving data from sinks).
- *Demand Weight*: for a particular VM (vm_i), the weights of the demands for the sinks are given as a demand vector $V_i = (v_{i1}, v_{i2}, \dots, v_{i|S|})$ with elements between 0 and 1 whose sum is equal to 1. (v_{ik} is the weight of demand for sink k in vm_i).

Moreover, suppose that each PM-sink pair is associated with a numerical cost. It is clearly not a good idea to place a VM with intensive demand for sink x in a PM that has a high cost associated with that sink.

Based on these assumptions, we define a metric named satisfaction that shows how “satisfied” a given virtual machine v is, when placed on a physical machine p.

By maximizing the overall *satisfaction* of the VMs, we can claim that both the service provider and the service consumer sides will be in a win-win situation. From consumer’s point of view, the VMs will experience a better quality of service which is a catch for users. Similarly, on the provider side, the links will be less likely to be saturated which enables serving more VMs.

The placement problem in our scenario is the complement of the famous Quadratic Assignment Problem (QAP) [19] which is NP-hard. On account of the dynamic nature of