Computers and Electrical Engineering 68 (2018) 44–61

Contents lists available at ScienceDirect

Computers and Electrical Engineering

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

Scalable and direct vector bin-packing heuristic based on residual resource ratios for virtual machine placement in cloud data centers☆

Saikishor Jangiti, Shankar Sriram. V.S.*

Center for Information Super Highway (CISH), School of Computing, SASTRA Deemed University, Thanjavur, Tamilnadu, India

A R T I C L E I N F O

Keywords:
Cloud computing
Infrastructure as a service
VM placement
Server consolidation
Scalability
Vector bin-packing
Resource provisioning

A B S T R A C T

Virtual Machine (VM) placement consolidates VMs into a minimum number of Physical Machines (PMs), which can be viewed as a Vector Bin-Packing (VBP) problem. Recent literature reveals the significance of first-fit-decreasing variants in solving VBP problems, however they suffer from reduced packing efficiency and delayed packing speed. This paper presents VM NeAR (VM Nearest and Available to Residual resource ratios of PM), a novel heuristic method to address the above said challenges in VBP. Further, we have developed Bulk-Bin-Packing based VM Placement (BBPVP) and Multi-Capacity Bulk VM Placement (MCBVP) as a solution for VBP. The simulation results on real-time Amazon EC2 dataset and synthetic datasets obtained from CISH, SASTRA shows that VM NeAR based MCVBP achieves about 1.6% reduction in the number of PMs and possess a packing speed which was found to be 24 times faster than exisiting state-of-the-art VBP heuristics.

1. Introduction

Rapid growth in the Internet and Information Technology (IT) services has driven the emergence of cloud computing technologies as an alternative to the traditional computing infrastructures. Infrastructure as a Service (IaaS), a cloud service delivery model provides on-demand access to remote computing resources available at various Cloud Data Centers (CDCs) across the globe. The end users hire Virtual Machine (VM) instances from popular Cloud Service Providers (CSPs) (e.g., Amazon EC2,[1] Google Compute Engine,[2] and Microsoft Azure[3]) and run applications on their own. In general, the CSPs offer reliable and robust compute and storage resources based on the users' Quality of Service (QoS) requirements for selected period. According to Gartner, "*The cloud market is growing by 40 percent, and a corporate without cloud will be rare by 2020*". Due to the rapid growth in the adoption of cloud infrastructure and services, the CDCs are expected to hold beyond 10% of the global energy utilisation [1]. Energy-efficient management of CDCs is an active research topic which aims at reducing the operational expenses, thereby making CSPs to sustain in the competitive cloud marketplace. Further, it contributes towards minimizing the carbon footprints and its adverse environmental impacts. A major component of CDC's power consumption is by Physical Machines (PMs), and even when idle, PMs consume 60% of the full load power. From the **United States data center energy usage report-2016**, it was evident that the average resource utilization in

☆ Reviews processed and recommended for publication to the Editor-in-Chief by Associate Editor Dr. L. Bittencourt.
* Corresponding author.
E-mail address: sriram@it.sastra.edu (S. Sriram. V.S.).

[1] https://aws.amazon.com/ec2
[2] https://cloud.google.com/compute
[3] https://azure.microsoft.com

https://doi.org/10.1016/j.compeleceng.2018.03.029
Received 1 May 2017; Received in revised form 19 March 2018; Accepted 19 March 2018

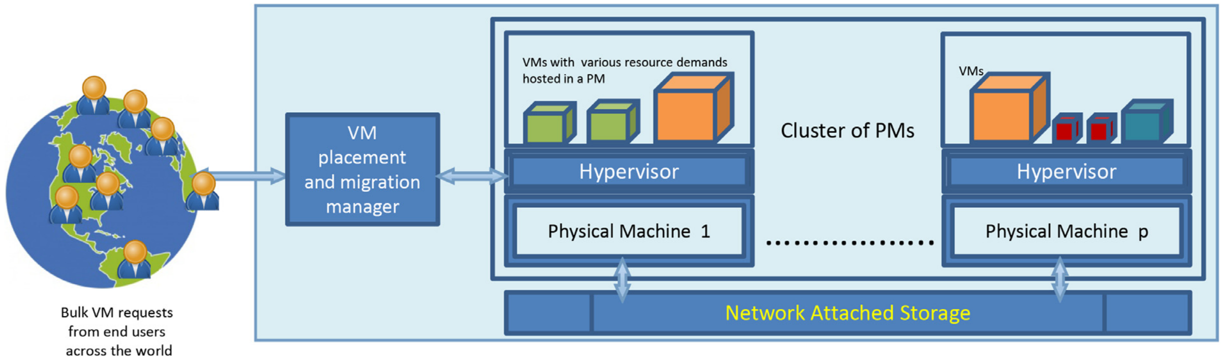

Fig. 1. Architecture model of a cloud data center.

CDCs was 14% and through VM consolidation, CDCs could save 520 billion kWh by 2020, thereby insists the importance of virtualization technologies in improving energy efficiency. In general, virtualization technologies enables sharing of a physical resource by multiple software components to form secure and isolated IT resources in the form of VMs. Fig. 1 represents the generic architecture of a cloud data center. The CDC receives bulk VM requests of different resource requirements from the customers across the globe. The autonomic cloud manager assigns VM requests into fewer PMs to minimize the operational expenses, which is a classical application of multi-capacity or Vector Bin-Packing (VBP).

This work is one another attempt to solve the VM Placement as a vector bin packing problem, since the economic and environmental benefits of this study places it as an active research challenge. The efficiency of the VBP heuristics is determined by two factors, namely packing efficiency and speed to decide whether to use a bin-packing heuristic or not. The state-of-the-art meta-heuristics and greedy heuristics for solving VBP problem either focus on the packing efficiency or speed. Further, the speed and packing efficiency of the existing approaches have a negative correlation between them, scalability is another prime quality metric to be considered along with the speed and packing efficiency for VM placement. In this way, we propose the first direct VBP heuristic, multi-capacity bulk VM placement with improved speed, better packing efficiency and high scalability for energy efficient VM placement.

OpenStack[4] and Eucalyptus[5] are the popular tools used to build public, private and hybrid clouds. The automatic management of these tools use bin-packing heuristics for VM initial placement. From the literature, it was evident that greedy heuristics outperform the meta-heuristic approaches like genetic algorithm and local search in 79.08% instances [2–4]. Variants of First Fit Decreasing (FFD), a one-dimensional bin-packing greedy heuristic are widely used to solve VBP due to the lack of direct VBP heuristic. Panigrahy et al. [5] studied the FFD variants like FFDProd, FFDSum, FFDAvgSum, and FFDExpSum & designed new geometric heuristics, namely Norm-Based Greedy (NBG) and Dot Product (DP) for VM Placement which dominates the other heuristics in terms of solution quality. Microsoft cloud platform system center (virtual machine manager) also uses DP and NBG as an internal function for VM consolidation [6]. Wei Zhu et al. [7] recently proposed a DP based vector bin packing algorithm named Heuristic Virtual Resource Allocation Algorithm (HVRAA). To summarize, we have identified two significant limitations in DP and NBG i.e., *delayed packing speed* and *reduced packing efficiency* due to *continuous sorting* and *collision in the vector-to-scalar conversions* respectively. *Continuous sorting* improves the solution quality; DP and NBG update the scalar weights of items according to the changes in PM residual capacities after placing each VM. This update is followed by a sort to determine the largest item. These repeated sortings will slow down the placement process and also raise the scalability issues when the number of items is large. *Collision in the vector-to-scalar conversions* is that the conversion of two or more completely different vectors may result in the same scalar value. This collision leads to a wrong choice of VMs for placement and has a high impact on the solution quality. Hence, this paper presents a novel VM choosing heuristic called (VM NeAR) which employs a new bin and item representation vector called Resource Ratio Vector (RRV) to address the above-said limitations.

The CSPs offer some standard VM instances, and there will be repeated (more than one) requests of the same VM instance type from customers across the globe. It is observed that in case of repeated requests, more than one PM end up with the same placement. Bulk-Bin-Packing (BBP) is proposed from this observation and is used to speed up the placement process by identifying how many more PMs can be filled with the same kind of VMs.

In addition to VM NeAR, we have also developed:

(i) Bulk-Bin-Packing based VM Placement (BBPVP), a rapid and scalable bin-packing of repeated VM requests with same resource ratios
(ii) Multi-Capacity Bulk VM Placement (MCBVP), the first direct VBP heuristic

VM placement problem has two subproblems: **VM initial placement** - performed once, the first time when a VM request comes.

[4] https://www.openstack.org/
[5] http://www.eucalyptus.com/