A Study of Virtual Machine Placement Optimization in Data Centers

Stéphanie Challita, Fawaz Paraiso and Philippe Merle
Inria Lille - Nord Europe, France
University of Lille, CRIStAL UMR CNRS 9189, France
firstname.lastname@inria.fr

Keywords: VM placement, Data center, Multi-objective optimization, Energy-aware, Traffic-aware, Cloud computing.

Abstract: In recent years, cloud computing has shown a valuable way for accommodating and providing services over the Internet such that data centers rely increasingly on this platform to host a large amount of applications (web hosting, e-commerce, social networking, etc.). Thus, the utilization of servers in most data centers can be improved by adding virtualization and selecting the most suitable host for each Virtual Machine (VM). The problem of VM placement is an optimization problem aiming for multiple goals. It can be covered through various approaches. Each approach aims to simultaneously reduce power consumption, maximize resource utilization and avoid traffic congestion. The main goal of this literature survey is to provide a better understanding of existing approaches and algorithms that ensure better VM placement in the context of cloud computing and to identify future directions.

1 INTRODUCTION

Cloud computing is a technology that enables computing resources, software, or data to deliver on-demand services over the Internet. These resources have become cheaper, more powerful and more ubiquitously available than ever before (Mell and Grance, 2011). The cloud computing stack consists of three types of cloud service models: *Infrastructure*, *Platform* and *Software*, which are built upon the *Hardware* layer. Since provisioning Virtual Machines (VMs) is fundamental to provide infrastructure services, one can say that virtualization is the key concept of cloud computing. According to Mike Adams, director of product marketing at VMware[1], "*Virtualization software makes it possible to run multiple operating systems and multiple applications on the same server at the same time. It enables businesses to reduce IT costs while increasing the efficiency, utilization and flexibility of their existing computer hardware.* "(Angeles, 2014)

However, VMs need to be adequately placed to fulfill performance goals, to optimize network flows, and to reduce CPU, storage and energy costs. VM placement optimization processes may be traffic-aware, energy-aware, application-aware, network topology-aware, data-aware, or a combination of these.

In recent years, the problem of allocating VMs to suitable Physical Machines (PMs) has been studied for efficiency and quality purposes (Xu and Fortes, 2010), (Feller et al., 2011), (Fang et al., 2013), (Gao et al., 2013), (Vu and Hwang, 2014). On the provider side, these solutions map VMs to PMs to optimize server efficiency, allowing some servers to hibernate or shut down depending on load conditions. On the consumer side, these solutions maximize Quality of Service (QoS) and Quality of Experience (QoE). In addition to this, such solutions lead to better utilization of resources and less frequent overload situations, leading to cost savings.

Although VMs have shown considerable opportunities to the IT industry, their placement brings many challenges that need to be carefully addressed. In this paper, we present a survey of various approaches studying VM placement, highlighting their key concepts, as well as the state-of-the-art implementations. The remainder of this paper is organized as follows. In Section 2, we motivate the need of VM placement optimization. Section 3 presents the state-of-the-art that details the four approaches addressing energy, cost and data flow issues in data centers. Section 4 discusses potential comparison metrics between presented solutions. In Section 5, we discuss related surveys that allow us to situate our work. Finally, Section 6 concludes and outlines future work.

[1]VMware is an innovator in virtualization and cloud infrastructure, http://www.vmware.com/

2 MOTIVATION

As stated before, the motivation behind VM placement optimization can be traffic-aware, energy-aware, application-aware, network topology-aware, data-aware, and multi-objective. In our paper, we are interested in investigating solutions that reduce energy consumption (Ajiro and Tanaka, 2007), (Verma et al., 2008), (Chaisiri et al., 2009), resource costs (Xu and Fortes, 2010), (Fajjari et al., 2014), and properly manage data flow (Kanagavelu et al., 2014), (Aral and Ovatman, 2016) in data center architectures. Each of these objectives is detailed below.

2.1 Energy Management

Enhancing energy efficiency in data centers can be resolved by applying a suitable VM placement algorithm that minimizes the cost of powering at the hardware level. Moreover, turning off unused machines, on the basis of server consolidation[2] and energy-aware job scheduling, can also constitute a solution for the energy problem. In this context, "Green Data Centers" (Basmadjian et al., 2015) are nowadays a must to fight against huge power consumption and bills caused by inappropriate virtualization.

2.2 Resource Usage Optimization

In order to maintain the application performance, isolation and security, each VM requires a certain amount of resources, such as CPU, memory and links bandwidth, etc. In order to minimize their cost, these resources should be made available to applications only as needed and not allocated statically based on the peak workload demand (Kusic et al., 2009). This is known as the "elasticity of the cloud" (Herbst et al., 2013).

2.3 Traffic Engineering

Measuring and optimizing data center traffic is important to maintain the efficiency of applications. A data center, which hosts thousands of devices like servers, switches and routers, needs an accurate planning of the network architecture. One can distinguish several architectures (Wang et al., 2014) such as Fat-tree (Al-Fares et al., 2008), VL2 (Greenberg et al., 2009) and BCube (Guo et al., 2009).

[2]Server consolidation is an approach to the efficient usage of physical resources in order to reduce the total number of servers that an organization requires. It involves gathering several VMs into a single physical server.

3 APPROACHES

In this section, we review existing methods related to the optimization of VM placement, embedded in the cloud computing domain. Placement algorithms, that collect and study information from deployed applications, can be either static or dynamic. Static algorithms, which mainly do offline calculations, take as an input the information that is formerly collected. After a primary placement, a relocation may not be considered for several months. Static allocation normally indicates initial VM placements that will be actively running in subsequent phases of the system administration. As for dynamic VM placement, it is implemented on shorter timescales, preferably shorter than periods of significant variability of the resource demand (Shankar and Bellur, 2010), (Abdelsamea et al., 2014). It does online VM placement, including VM migrations. One very important difference between static and dynamic VM placement algorithms is the fact that dynamic solutions consider potential VM live migrations and therefore require larger amount of resources than the static solutions, which can badly affect the performance of the hosted applications. Since the static solutions are primitive and outdated (Usmani and Singh, 2016), we are interested to study the dynamic VM placement optimization solutions. As shown in Figure 1, our classification is based on four main approaches: *1) Constraint Programming*, *2) Bin Packing*, *3) Stochastic Integer Programming*, and *4) Genetic Algorithm*. We detail in the following the algorthims that were implemented through these approaches.

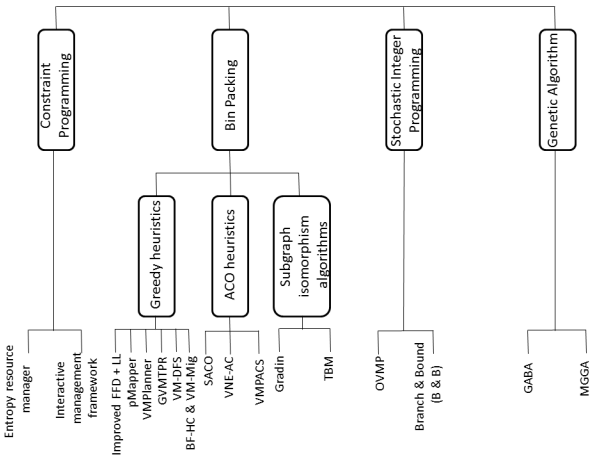

Figure 1: VM placement approaches.

3.1 Constraint Programming

Foremost Constraint Programming is used where we have already completely collected the input information, so we start the cost function calculations after we are aware of the VM requirements. Since this approach can always consider additional constraints, it can always be expandable. However, in cases where we have several constraints to take into consideration, this approach may take too much time to find the most suitable VM placement. Therefore, the main challenge consists in finding the optimal solution before any modification in terms of the constraint parameters. Although this approach formally expresses the demands of VMs in terms of constraints, it has been modestly adopted. In 2009, Hermenier et al. (Hermenier et al., 2009) introduce the Entropy resource manager, which is a constraint programming-based solution allowing for a dynamic server consolidation. It tackles both problems of finding the available servers, and migrating the VMs to these servers. This solution is only applied in homogeneous data centers. In 2010, Van et al. (Van et al., 2010) propose an interactive resource management framework that ensures both dynamic VM provisioning and placement, by treating them as two constraint satisfaction problems.

3.2 Bin Packing

Bin Packing is used for dynamic VM placement when all servers have the same amount of resources (CPU, memory, storage, etc.), but the requirements of VMs are variable in time. In order to minimize the number of PMs, this approach may host two dependent VMs on one PM. The Bin Packing problem is an NP-hard[3] problem that can be solved using **i) Greedy heuristics** like First Fit Decreasing (FFD) algorithm (Ajiro and Tanaka, 2007), (Tang et al., 2014), (Verma et al., 2008), Best Fit (BF) algorithm (Jiankang et al., 2015), Least Loaded (LL) algorithm (Ajiro and Tanaka, 2007), etc., **ii) Ant Colony Optimization (ACO) heuristics** (Dorigo and Gambardella, 1997), (Dorigo et al., 1996), (Stützle and Hoos, 2000), **iii) Subgraph isomorphism algorithms** (Ullmann, 1976).

3.2.1 Greedy heuristics

FFD algorithm is a greedy heuristic. It places each VM into "the first bin in which it will fit". This approach is more efficient when first sorting the list of elements into a decreasing order. For example, Ajiro et al. (Ajiro and Tanaka, 2007) present an Improved FFD + LL heuristic. Both FFD and LL heuristics pack VMs with certain resources into destination servers. However, other resources may interfere and increase the number of destination servers. The proposed algorithm aims to address this issue and reduce the number of destination servers.

The pMapper system (Verma et al., 2008) packs the VMs in a small number of physical servers in order to minimize the migration costs, but always under a fixed constraint of performance. The pMapper system is a subset of the FFD heuristic and it aims for a compromise between performance and cost.

In 2013, VMPlanner (Fang et al., 2013) is proposed as another greedy Bin Packing algorithm for reducing power costs of network elements in data centers. VMPlanner takes advantage of the flexibility provided by dynamic VM migration and programmable flow-based routing, available in modern data centers, to optimize network power consumption while satisfying network traffic demands. In order to evaluate VMPlanner, the authors implement it in a simulated environment running real data center traffic workloads, with a VL2 architecture (Greenberg et al., 2009).

Kanagavelu et al. (Kanagavelu et al., 2014) develop a fast heuristic algorithm called Greedy Virtual Machine Placement with Two Path Routing (GVMTPR) that is based on some constraints. Its goal is to reduce the maximum load on any link, by reducing the network traffic load, followed by the number of used servers and the server resource utilization, while guaranteeing the specified protection grade requirements. It uses a greedy approach to assign the hypervisor for each VM, in order to satisfy the computing and memory resource requirements. This will be done due to the VM management and routing decisions. In addition, instead of a single-path routing, it chooses a two-path routing to ensure load balancing on the two least-congested paths. Traffic splitting helps to reduce congestion, which is largely influenced by the VM placement, and ensures traffic protection (minimum bandwidth) in the event of failures.

Likewise, in 2014, Tang et al. (Tang et al., 2014) present VM-DFS, an algorithm based on a dynamic forecast scheduling method and that is a sort of Bin Packing problem which can also be solved by FFD. VM-DFS analyzes the historical memory consumption of each VM, to select the most appropriate PM to place the running VM according to the future consumption prediction. The purpose of this method is to reduce the number of active servers, such that every VM can conform to the required memory consump-

[3] A problem is NP-hard if the algorithm for solving it can be translated into one for solving any NP-problem (nondeterministic polynomial time) problem. NP-hard therefore means "at least as hard as any NP-problem," but it can eventually be harder!

tion SLA[4].

In (Jiankang et al., 2015), the authors propose BF-HC and VM-Mig, two algorithms based on the BF heuristic. The difference from FFD heuristic is that the VMs are placed in the bin that can hold them with minimum empty space. The objective of this two-stage heuristic algorithm is to minimize the energy consumption and to optimize the traffic management, by respectively reducing the number of activated PMs and network elements and maximizing the link utilization. The evalutaion is performed by simulation experiments in Amazon EC2 data center using Fat-tree topology (Al-Fares et al., 2008).

3.2.2 ACO heuristics

Broadly speaking, ACO is a probabilistic technique for solving computational problems, which can be reduced to finding good paths through graphs. The collective behaviour of social insects is an inspiration source for researchers. When searching for food, ants tend to choose paths marked by strong pheromone concentrations. As soon as an ant finds a food source, it studies the quantity and the quality of the food and takes some of it back to the nest. During the return trip, the quantity of pheromones that an ant leaves on the ground may depend on the quantity and quality of the food. The pheromone trails will guide other ants to the food source and enables them to find the shortest paths between their nest and food sources (Blum, 2005). Briefly, ants mark the best solutions and take into account the previous markings to optimize their search (Yu et al., 2008). This behaviour, aiming for the shortest paths, can be used for the VM migration optimization between PMs. Some extensions of ACO algorithms are presented in the literature such as Ant System (AS) (Dorigo et al., 1996), Ant Colony System (ACS) (Dorigo and Gambardella, 1997) and MAX-MIN Ant System (MMAS) (Stützle and Hoos, 2000). Let us note that AS is the progenitor of all the research efforts with Ant algorithms. It is widely used to solve the Traveling Salesman Problem (TSP[5]) (Hoffman et al., 2013).

MMAS (Stützle and Hoos, 2000) is a variant of ACO that aims to exploit the best solutions found during the search in order to reach an optimal solution. MMAS updates the pheromone according to the best solution found from the beginning of the process until the current iteration. Feller et al. (Feller et al., 2011) propose Single-objective ACO (SACO), a MMAS metaheuristic-based algorithm that is focused on reducing the number of physical servers, which are needed to handle the workload. Moreover, Fajjari et al. (Fajjari et al., 2014) propose a new, efficient and scalable, strategy named VNE-AC. It is based on a MMAS metaheuristic too. Its main aim is to reduce the amount of allocated resources for each virtual network request. This will help to minimize the reject rate and maximize the cloud provider[6]'s revenue. Based on extensive simulations, this proposal makes the VM placement more effective and achieves better resource utilization even when assuming a high arrival rate of VM requests. Its fundamental stages are: *i)* formation of solution components, *ii)* localization of potential candidates, *iii)* stochastic selection of the candidate, *iv)* selection of the best solution, and *v)* updating the pheromone trail. Moreover, VNE-AC goes beyond related strategies investigated in existing literature, such as VNE-Greedy (Yu et al., 2008), VNE-Cluster (Zhu and Ammar, 2006), VNE-Subdividing (Zhu and Ammar, 2006), and VNE-Least (Zhu and Ammar, 2006).

Most of the solutions that attempt to optimize the placement of VMs are shifted towards processing only one criterion. However, similar to any other problem, it is usually useful to simultaneously consider several criteria. For this, a new research direction aims to realize multi-objective optimized allocation of VMs. Therefore, Gao et al. (Gao et al., 2013) propose a solution that tackles at a time the problem of resource waste and energy utilization. Knowing that ACS is built upon the previous AS, a modified version of ACS algorithm is proposed and designed to properly deal with the potential large solution space for large-scale data centers. Its name is VMPACS. A performance benchmark is carried out between the proposed Ant algorithm and other algorithms, such as Multi-objective Grouping Genetic Algorithm (MGGA[7]) (Xu and Fortes, 2010), SACO[8] algorithm (Feller et al., 2011) and the Improved FFD + LL algorithm (Ajiro and Tanaka, 2007). To summarize the results, we provide in Table 1 the energy utilization and resource consumption comparison between the algorithms under consideration.

As depicted in Table 1, it can be noticed that VMPACS goes beyond MGGA. Indeed, the choice of

[4]A Service-Level Agreement (SLA) is a contract between a network service provider and a customer that specifies, in measurable terms, what services the network service provider will furnish.

[5]Given a list of cities and the distances between each pair of cities, TSP looks for the shortest possible route that visits each city exactly once and returns to the origin city.

[6]Such as Google, Microsoft and Amazon.

[7]Further information about MGGA are provided in Section 3.4

[8]SACO is a modified MMAS algorithm for VM placement.

Table 1: Comparison of VMPACS and other algorithms.

	Energy Utilization	Resource Consumption
MGGA (Xu and Fortes, 2010)	✓	✓
Improved FFD + LL (Ajiro and Tanaka, 2007)	✓✓✓✓	✓✓
VMPACS (Gao et al., 2013)	✓✓✓	✓✓✓
SACO (Feller et al., 2011)	✓✓	✓✓✓✓

placement given by VMPACS consists of: *i)* the data being collected at the moment and *ii)* the traces kept by a non-optimal solution. VMPACS also updates, in a continuous way, the pheromones, which allows a much more relevant VM placement. Evenly, VMPACS goes beyond Improved FFD + LL heuristic and SACO. These results are mainly related to the number of servers employed and to the resource utilization of each algorithm.

3.2.3 Subgraph isomorphism algorithms

The subgraph isomorphism problem is a NP-hard computational task in which two graphs are given as input, and one must determine whether the first graph contains a subgraph that is isomorphic to the second graph. Recently, several algorithms have used subgraph isomorphism to formulate the problem of VM placement, i.e., to model data center topologies and VM clusters. (Zong et al., 2014) proposes Gradin, a new graph index framework that accelerates subgraph matching on dynamic graphs of numerical labels. Gradin efficiently calculates frequent index updates and eliminates unpromising matches to minimize the cost. Gradin's performance is evaluated over BCube topology (Guo et al., 2009), on query processing, index update and scalability. Results show that Gradin outperforms competitive approaches such as VF2 (Cordella et al., 2004) up to 10 times.

(Aral and Ovatman, 2016) provides Topology Based Mapping (TBM) algorithm that uses a subgraph search to locate federated clouds with a topology isomorphic to the VM cluster. Additionally, it presents RalloCloud framework for modeling and simulating distributed VM allocation. Using RalloCloud for TBM evaluation, results indicated that TBM outperforms greedy heuristics in latency, throughput, cost and acceptance rate.

3.3 Stochastic Integer Programming

Stochastic Integer Programming is introduced for situations where future demands and prices of resources are unknown, but their expected distributions are either known or can be computed. This is the best technique to be used in the case where we have two or more uncertain parameters on which the cost depends. It is helpful in estimating the variation in demands and costs. Thereby, frequent recomputations are not needed, but if there is an error in the estimation, users might end up paying more. For example, the authors in (Chaisiri et al., 2009) define Optimal Virtual Machine Placement (OVMP), an algorithm for optimally placing the VMs on the suitable PMs. The goal is to minimize the cost of deploying and running VMs in a cloud provider environment, where future demands and prices are not necessarly stable. This objective is reached by reducing the number of used nodes. The algorithm they define is particularly based on linear and quadratic programming. In (Speitkamp and Bichler, 2010), the authors translate the server consolidation problem into a linear programming formulation. Their approach, Branch & Bound (B & B), aims at limiting the number of VMs in each PM. As a result, it is certain that VMs are not hosted in a single physical server. Some design constraints were added in order to achieve this approach, very useful to ensure fault-tolerancy in case of outages.

3.4 Genetic Algorithm

Finally, Genetic Algorithm is particularly convenient in cases where we need to operate on groups and for problems where objective functions[9] dynamically change. It considers additional constraints while optimizing the cost function, so it solves the VM interference problem encountered in the Bin Packing approach, but it requires more computing time and higher computing resources as compared to Bin Packing. Mi et al. (Mi et al., 2010) propose a genetic algorithm-based approach, namely GABA, to adaptively self-reconfigure the VMs in cloud data centers consisting of heterogeneous nodes. GABA can efficiently decide the optimal physical placement of VMs according to application requirements and dynamic environmental conditions, that may vary over time. Moreover in the same year, the VM placement problem is formulated in (Xu and Fortes, 2010) as

[9] An objective function is a function to maximize or minimize.

a multi-objective optimization problem of simultaneously minimizing total resource wastage, power consumption and thermal dissipation costs. MGGA with fuzzy multi-objective evaluation is investigated for efficiently searching the large solution space and conveniently combining possibly conflicting objectives.

4 DISCUSSION

A proper VM placement is essential for easier failure recovery, better geographical coverage and vendor lock-in avoidance in the cloud computing. The existing VM placement techniques consider various approaches, system assumptions, features of data centers like the network topology, as well as different evaluation approaches. Moreover, VM placement and migration is a broad research area with various optimization and objectives. Some of the techniques are only focused on a single objective, such as (Ajiro and Tanaka, 2007), (Chaisiri et al., 2009) and (Feller et al., 2011). Other techniques try to incorporate multiple objectives while making VM placement decisions, such as (Xu and Fortes, 2010) and (Gao et al., 2013). We can state that Bin Packing is the most employed approach. It always generates a good solution in a correct amount of time.

It is crucial to choose a VM placement technique that suits the needs of both the cloud user and cloud provider. However, due to the presence of several parameters, comparative analysis in a uniform fashion of such techniques becomes quite tricky. In fact, each of the VM placement algorithm works well under certain specific conditions/objectives. Thereby, in order to compare the efficiency of the previous algorithms, we propose that future empirical studies will be based on the three following metrics to measure and evaluate the algorithms performance. Firstly, one should take into account the *energy amount* consumed by data center resources, due to the application workloads. The second metric to be considered is the *SLA violation percentage*, which expresses the level by which performance requirements defined between the resource provider and consumers are violated. The SLA violation can happen when VMs sharing the same PM need a CPU performance that cannot be provided because of energy-aware resource management and consolidation. The provider pays a penalty to the client in case of SLA violation. The third metric is the *number of VM migrations* during the adaptation of the VM placement (Beloglazov et al., 2012). VM migrations consume time, energy and network bandwidth. Thus, it is important to minimize the number of VM migrations.

5 RELATED SURVEYS

Three recent surveys also discuss the state-of-the-art of VM placement techniques in data centers. (Pires and Barán, 2015) covers several axes such as *objective functions* (i.e., energy consumption minimization, network traffic minimization, economical costs optimization, and performance maximization), *solution techniques* (i.e., algorithms and meta-heuristics), *cloud architectures* (i.e., single cloud, multi-clouds, and federated clouds), and *experiment types* (i.e. simulation and applications). However, (Pires and Barán, 2015) remains mainly statistics-oriented. Our work provides though better details especially regarding the solution techniques. It is also focused on energy-aware and traffic-aware solutions.

Also, we consider (Jennings and Stadler, 2015) as a decent and comprehensive survey. It details the different kinds of cloud resources and discusses approaches based on eight subdomains: *global scheduling, local scheduling, demand profiling, utilization estimation, pricing, application scaling, workload management, cloud management, and measurement studies*. Being focused on resource management for cloud environments in a general way, (Jennings and Stadler, 2015) misses quite a number of approaches in VM placement field. The authors refer to the latter as *"global scheduling"* subdomain. In contrast, our taxonomy focuses on VM placement problem and covers all the approaches that have been investigated in this field.

Finally, while (Usmani and Singh, 2016) offers a similar classification to ours, its motivation is mainly focused on server consolidation. Our survey addresses the topic of VM placement with a broader motivation and takes into account the different types of system architecture. Moreover, we explain the optimal case for the use of each of the four discussed approaches. We certainly fill a void in the literature by providing this up-to-date survey, which is useful for the cloud computing domain.

6 CONCLUSION AND FUTURE DIRECTIONS

In this paper, we survey the state-of-the-art of VM placement optimization, covering its essential approaches and key existing solutions that reduce network power cost and prevent congestion of data flow. As the cost and the performance of cloud data centers become a practical concern and attract attention of both service provider and consumer, our survey provides a better comprehension of the existing VM

placement algorithms that deal with power cost and handle traffic in data centers.

However, many future directions and perspectives have not been explored yet and can be contemplated for the future. For example, data security remains unresolved in cloud infrastructures. Since the migration of VMs require a secure connection between both source and target servers, we pave the way for further work to address this issue and we mark the urge of defining reliable protocols for establishing and managing a protected communication. Moreover, in order to optimally benefit from the features of the four approaches, designing an hybrid solution combining several approaches represents a future challenge. Finally, nowadays there are several resource managers that are mostly doing the placement of VMs, like *vMotion* (Murphy, 2011), the commercial product of VMware and *OpenStack* (Sefraoui et al., 2012), the open-source cloud manager. Other resource managers like *Kubernetes* (Brewer, 2015), *Swarm* (Thapatsuwan et al., 2009), and *Mesos* (Hindman et al., 2011), to cite a few, are responsible for the placement of containers. Therefore, it will be interesting to conduct an exhaustive study of the existing static and dynamic resource managers, and to map between them and the placement algorithm(s) they use.

ACKNOWLEDGEMENTS

This work is supported by the OCCIware[10] research and development project funded by French Programme d'Investissements d'Avenir (PIA). Likewise, this work is partially funded by Nord-Pas de Calais Regional Council.

REFERENCES

Abdelsamea, A., Hemayed, E. E., Eldeeb, H., and Elazhary, H. (2014). Virtual Machine Consolidation Challenges: A Review. *International Journal of Innovation and Applied Studies*, 8(4):1504.

Ajiro, Y. and Tanaka, A. (2007). Improving Packing Algorithms for Server Consolidation. In *Int. CMG Conference*, pages 399–406.

Al-Fares, M., Loukissas, A., and Vahdat, A. (2008). A Scalable, Commodity Data Center Network Architecture. *ACM SIGCOMM Computer Communication Review*, 38(4):63–74.

Angeles, S. (2014). Virtualization vs Cloud Computing: What's the Difference? *BusinessNewsDaily, January 20*.

[10]http://www.occiware.org/

Aral, A. and Ovatman, T. (2016). Network-Aware Embedding of Virtual Machine Clusters onto Federated Cloud Infrastructure. *Journal of Systems and Software*, 120:89–104.

Basmadjian, R., Bouvry, P., Da Costa, G., Gyarmati, L., Kliazovich, D., Lafond, S., Lefevre, L., De, H., Meer, J.-M. P., Pries, R., et al. (2015). Green Data Centers. *Large-scale Distributed Systems and Energy Efficiency: A Holistic View. John Wiley & Sons. P*, pages 159–196.

Beloglazov, A., Abawajy, J., and Buyya, R. (2012). Energy-Aware Resource Allocation Heuristics for Efficient Management of Data Centers for Cloud Computing. *Future Generation Computer Systems*, 28(5):755–768.

Blum, C. (2005). Ant colony optimization: Introduction and recent trends. *Physics of Life reviews*, 2(4):353–373.

Brewer, E. A. (2015). Kubernetes and the path to cloud native. In *Proceedings of the Sixth ACM Symposium on Cloud Computing*, pages 167–167. ACM.

Chaisiri, S., Lee, B.-S., and Niyato, D. (2009). Optimal Virtual Machine Placement across Multiple Cloud Providers. In *IEEE Asia-Pacific Services Computing Conference, APSCC 2009*, pages 103–110. IEEE.

Cordella, L. P., Foggia, P., Sansone, C., and Vento, M. (2004). A (sub) Graph Isomorphism Algorithm for Matching Large Graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10):1367–1372.

Dorigo, M. and Gambardella, L. M. (1997). Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation*, 1(1):53–66.

Dorigo, M., Maniezzo, V., and Colorni, A. (1996). Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 26(1):29–41.

Fajjari, I., Aitsaadi, N., Pióro, M., and Pujolle, G. (2014). A New Virtual Network Static Embedding Strategy within the Clouds Private Backbone Network. *Computer Networks*, 62:69–88.

Fang, W., Liang, X., Li, S., Chiaraviglio, L., and Xiong, N. (2013). VMPlanner: Optimizing Virtual Machine Placement and Traffic Flow Routing to Reduce Network Power Costs in Cloud Data Centers. *Computer Networks*, 57(1):179–196.

Feller, E., Rilling, L., and Morin, C. (2011). Energy-Aware Ant Colony Based Workload Placement in Clouds. In *Proceedings of the 2011 IEEE/ACM 12th International Conference on Grid Computing*, pages 26–33. IEEE Computer Society.

Gao, Y., Guan, H., Qi, Z., Hou, Y., and Liu, L. (2013). A Multi-Objective Ant Colony System Algorithm for Virtual Machine Placement in Cloud Computing. *Journal of Computer and System Sciences*, 79(8):1230–1242.

Greenberg, A., Hamilton, J. R., Jain, N., Kandula, S., Kim, C., Lahiri, P., Maltz, D. A., Patel, P., and Sengupta, S. (2009). Vl2: A Scalable and Flexible Data Center

Network. In *ACM SIGCOMM Computer Communication Review*, volume 39, pages 51–62. ACM.

Guo, C., Lu, G., Li, D., Wu, H., Zhang, X., Shi, Y., Tian, C., Zhang, Y., and Lu, S. (2009). BCube: A High Performance, Server-centric Network Architecture for Modular Data Centers. *ACM SIGCOMM Computer Communication Review*, 39(4):63–74.

Herbst, N. R., Kounev, S., and Reussner, R. H. (2013). Elasticity in Cloud Computing: What It Is, and What It Is Not. In *ICAC*, pages 23–27.

Hermenier, F., Lorca, X., Menaud, J.-M., Muller, G., and Lawall, J. (2009). Entropy: A Consolidation Manager for Clusters. In *Proceedings of the 2009 ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*, pages 41–50. ACM.

Hindman, B., Konwinski, A., Zaharia, M., Ghodsi, A., Joseph, A. D., Katz, R. H., Shenker, S., and Stoica, I. (2011). Mesos: A Platform for Fine-Grained Resource Sharing in the Data Center. In *NSDI*, volume 11, pages 22–22.

Hoffman, K. L., Padberg, M., and Rinaldi, G. (2013). Traveling Salesman Problem. In *Encyclopedia of Operations Research and Management Science*, pages 1573–1578. Springer.

Jennings, B. and Stadler, R. (2015). Resource Management in Clouds: Survey and Research Challenges. *Journal of Network and Systems Management*, 23(3):567–619.

Jiankang, D., Hongbo, W., and Shiduan, C. (2015). Energy-Performance Tradeoffs in IaaS Cloud with Virtual Machine Scheduling. *Communications, China*, 12(2):155–166.

Kanagavelu, R., Lee, B.-S., Le, N. T. D., Mingjie, L. N., and Aung, K. M. M. (2014). Virtual Machine Placement with Two-path Traffic Routing for Reduced Congestion in Data Center Networks. *Computer Communications*, 53:1–12.

Kusic, D., Kephart, J. O., Hanson, J. E., Kandasamy, N., and Jiang, G. (2009). Power and Performance Management of Virtualized Computing Environments via Lookahead Control. *Cluster computing*, 12(1):1–15.

Mell, P. and Grance, T. (2011). The NIST Definition of Cloud Computing.

Mi, H., Wang, H., Yin, G., Zhou, Y., Shi, D., and Yuan, L. (2010). Online Self-Reconfiguration with Performance Guarantee for Energy-Efficient Large-Scale Cloud Computing Data Centers. In *2010 IEEE International Conference on Services Computing (SCC)*, pages 514–521. IEEE.

Murphy, A. (2011). Enabling Long Distance Live Migration with F5 and VMware vMotion.

Pires, F. L. and Barán, B. (2015). A Virtual Machine Placement Taxonomy. In *15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGrid 2015, Shenzhen, China, May 4-7, 2015*, pages 159–168.

Sefraoui, O., Aissaoui, M., and Eleuldj, M. (2012). OpenStack: Toward an Open-Source Solution for Cloud Computing. *International Journal of Computer Applications*, 55(3).

Shankar, A. and Bellur, U. (2010). Virtual Machine Placement in Computing Clouds. *CoRR, vol. abs/1011.5064*.

Speitkamp, B. and Bichler, M. (2010). A Mathematical Programming Approach for Server Consolidation Problems in Virtualized Data Centers. *IEEE Transactions on Services Computing*, 3(4):266–278.

Stützle, T. and Hoos, H. H. (2000). Max-Min Ant System. *Future Generation Computer Systems*, 16(8):889–914.

Tang, Z., Mo, Y., Li, K., and Li, K. (2014). Dynamic Forecast Scheduling Algorithm for Virtual Machine Placement in Cloud Computing Environment. *The Journal of Supercomputing*, 70(3):1279–1296.

Thapatsuwan, P., Sepsirisuk, J., Chainate, W., and Pongcharoen, P. (2009). Modifying Particle Swarm Optimisation and Genetic Algorithm for Solving Multiple Container Packing Problems. In *ICCAE'09. International Conference on Computer and Automation Engineering, 2009.*, pages 137–141. IEEE.

Ullmann, J. R. (1976). An Algorithm for Subgraph Isomorphism. *Journal of the ACM (JACM)*, 23(1):31–42.

Usmani, Z. and Singh, S. (2016). A Survey of Virtual Machine Placement Techniques in a Cloud Data Center. *Procedia Computer Science*, 78:491–498.

Van, H. N., Tran, F. D., and Menaud, J.-M. (2010). Performance and Power Management for Cloud Infrastructures. In *2010 IEEE 3rd International Conference on Cloud Computing (CLOUD)*, pages 329–336. IEEE.

Verma, A., Ahuja, P., and Neogi, A. (2008). pMapper: Power and Migration Cost Aware Application Placement in Virtualized Systems. In *Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware*, pages 243–264. Springer.

Vu, H. T. and Hwang, S. (2014). A Traffic and Power-Aware Algorithm for Virtual Machine Placement in Cloud Data Center. *International Journal of Grid & Distributed Computing*, 7(1):350–355.

Wang, T., Su, Z., Xia, Y., and Hamdi, M. (2014). Rethinking the Data Center Networking: Architecture, Network Protocols, and Resource Sharing. *IEEE access*, 2:1481–1496.

Xu, J. and Fortes, J. A. (2010). Multi-Objective Virtual Machine Placement in Virtualized Data Center Environments. In *Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on & Int'l Conference on Cyber, Physical and Social Computing (CPSCom)*, pages 179–188. IEEE.

Yu, M., Yi, Y., Rexford, J., and Chiang, M. (2008). Rethinking Virtual Network Embedding: Substrate Support for Path Splitting and Migration. *ACM SIGCOMM Computer Communication Review*, 38(2):17–29.

Zhu, Y. and Ammar, M. H. (2006). Algorithms for Assigning Substrate Network Resources to Virtual Network Components. *INFOCOM*, 1200(2006):1–12.

Zong, B., Raghavendra, R., Srivatsa, M., Yan, X., Singh, A. K., and Lee, K.-W. (2014). Cloud Service Placement via Subgraph Matching. In *2014 IEEE 30th International Conference on Data Engineering*, pages 832–843. IEEE.