electronics

MDPI

Article

Phantom: Towards Vendor-Agnostic Resource Consolidation in Cloud Environments †

Aaqif Afzaal Abbasi [1], Mohammed A. A. Al-qaness [2], Mohamed Abd Elaziz [3], Ammar Hawbani [4], Ahmed A. Ewees [5], Sameen Javed [6] and Sunghwan Kim [7],*

[1] Department of Software Engineering, Foundation University, Islamabad 44000, Pakistan; aaqif.afzaal@fui.edu.pk
[2] School of Computer Science, Wuhan University, Wuhan 430072, China; alqaness@whu.edu.cn
[3] Department of Mathematics, Faculty of Science, Zagazig University, Zagazig 44519, Egypt; abd_el_aziz_m@yahoo.com
[4] School of Computer Science and Technology, University of Science and Technology of China, Hefei 230072, China; anmande@ustc.edu.cn
[5] Department of Computer, Damietta University, Damietta 34511, Egypt; ewees@du.edu.eg
[6] Quality Assurance Department, Keeptruckin Inc., Islamabad 44000, Pakistan; sameen.javed@keeptruckin.com
[7] School of Electrical Engineering, University of Ulsan, Ulsan 44610, Korea
* Correspondence: sungkim@ulsan.ac.kr; Tel.: +82-52-259-1401

† This research is an extended version of a paper published in Proceedings of the 3rd IEEE International Conference on Data Science and Computational Intelligence (DSCI'19), Shenyang, China, 21–23 October 2019.

Received: 24 September 2019; Accepted: 15 October 2019; Published: 18 October 2019

Abstract: Mobile-oriented internet technologies such as mobile cloud computing are gaining wider popularity in the IT industry. These technologies are aimed at improving the user internet usage experience by employing state-of-the-art technologies or their combination. One of the most important parts of modern mobile-oriented future internet is cloud computing. Modern mobile devices use cloud computing technology to host, share and store data on the network. This helps mobile users to avail different internet services in a simple, cost-effective and easy way. In this paper, we shall discuss the issues in mobile cloud resource management followed by a vendor-agnostic resource consolidation approach named Phantom, to improve the resource allocation challenges in mobile cloud environments. The proposed scheme exploits software-defined networks (SDNs) to introduce vendor-agnostic concept and utilizes a graph-theoretic approach to achieve its objectives. Simulation results demonstrate the efficiency of our proposed approach in improving application service response time.

Keywords: cloud computing; management; middle box; placement; resource; SDN; vendor-agnostic; virtual machine; VM

1. Introduction

Mobile-oriented future networks [1–3] are gaining tremendous importance in the field of computing and networking industry. With the advent of wireless networking technologies, the wide-scale use of smartphone devices and the World Wide Web is being shifted rapidly from static to mobility-based solutions. For example, mobile service users will exceed two billion users [4,5]. Such drastic changes are influencing the way IT concepts used to act and behave in the past.

However, the original idea of the internet was not based on mobility-based services. In other words, it can be said that the original idea of the internet was meant for fixed hosts instead of mobile hosts. So with the emergence of mobile technology, various patch-on protocols were introduced to support mobile environments e.g., Mobile IP [6,7] and its variants. However, patch-on technology based solutions also have their limitations.

In an environment based on mobile computing, support for mobility is a vital requirement rather as an add-on feature. The legacy network protocolsmainly focus on fixed hosts. In terms of usability, legacy protocols often describe mobility as an additional functionality of a device. This behavior leads to the creation of protocols based on mobility and is related to the modified versions of TCP/IP protocols suite [8–10]. With these trends of mobility-awareness in the protocols, unexpected degradation of performance, such as overuse of proxy, triangle routing, etc., is induced.

Mobile devices, including tablet PC or smartphones, are increasingly becoming an important part of our lives as a vital and easy sources of communication tools that are not bounded by the elements of time and space [11]. Mobile users utilize multiple mobile-based services by using different kinds of mobile apps. These apps are hosted on remote servers through wireless networks. The fast growth witnessed in mobile computing is a very prominent factor in the IT industry. It also influenced the commerce industry. However, with this fast growth of mobile devices, we are also facing numerous challenges such as computing resources car city, bandwidth allocation, storage and retrieval challenges and battery life time. Therefore, it can be safely said that the limitations of computing resources greatly hinder the betterment of computing services quality.

Cloud computing has been accepted as the infrastructure of next-generation networks [12]. Cloud users can benefit through cloud infrastructure by using various services (such as storage and services hosting), platforms (operating systems, middleware and related services) and software (applications) supported by cloud-enabled services like Amazon, Salesforce or Google at low prices. Furthermore, cloud computing enables its users to broadly utilize the resources on a pay-per-use policy [13]. By using such mobile applications, users can benefit from various cloud computing functions. With the rapid growth of mobile apps and better support for cloud-oriented services, the term mobile cloud computing is introduced. Mobile cloud computing is basically an integration of cloud computing in the mobile environment. With the advent of mobile cloud computing, mobile users are taking advantage of new type of services and which facilities them in fully utilizing cloud computing services.

Mobile cloud computing [14,15] has the potential to transform the large arena of the IT industry. This will help in making software and hardware-oriented services more accessible and attractive [1]. One of the primary objectives of cloud computing is to provide computing and storage services at low and reasonable costs. This happens by sharing many resources between different users. The actual provisioning of such services at a low process depends on how efficiently resources are utilized in the cloud. A typical mobile cloud computing infrastructure is illustrated in Figure 1.

Cloud vendors can offer special hardware and particular software techniques for the provisioning of reliable services at a high price. Later, these reliable services could be sold to users by signing terms under certain clause or service level agreement. Nowadays, the cloud computing industry is using the term “no single point of failure”. But the single point of failure often occurs when a single cloud service provider is hosting all these solutions [16]. It is worth mentioning that a true vendor-agnostic solutions will not only an open source technology solution (software) but will be accepted only if it is being operated on a vendor neutral hardware (by using off-the shelf, bare-metal/SDN-enabled devices) etc.

On the other hand, software-defined clouds (SDCs) make use of SDNs in order to create a programmable and flexible network by separation of functions for control plane and data plane. The reason for choosing SDNs in data center resource management is their simplicity and control over data center infrastructure. The idea of vendor agnostics through SDNs in data centers is implemented by the Open Flow with the decomposition of traffic control authorization to different parts [17–19]. The controller element is a powerful manager of the network that is processing information related to flows. Open Flow switches include basic functions like receiving, forwarding or looking up in a data traffic table. By using OpenFlow, routing is not confined to a Media Access Control (MAC) address or IP address. This basically helps in the determination of paths with the parameters of high security, low packet loss or low delay and also helps in maintaining the fine-grained scrutiny policies for various applications.

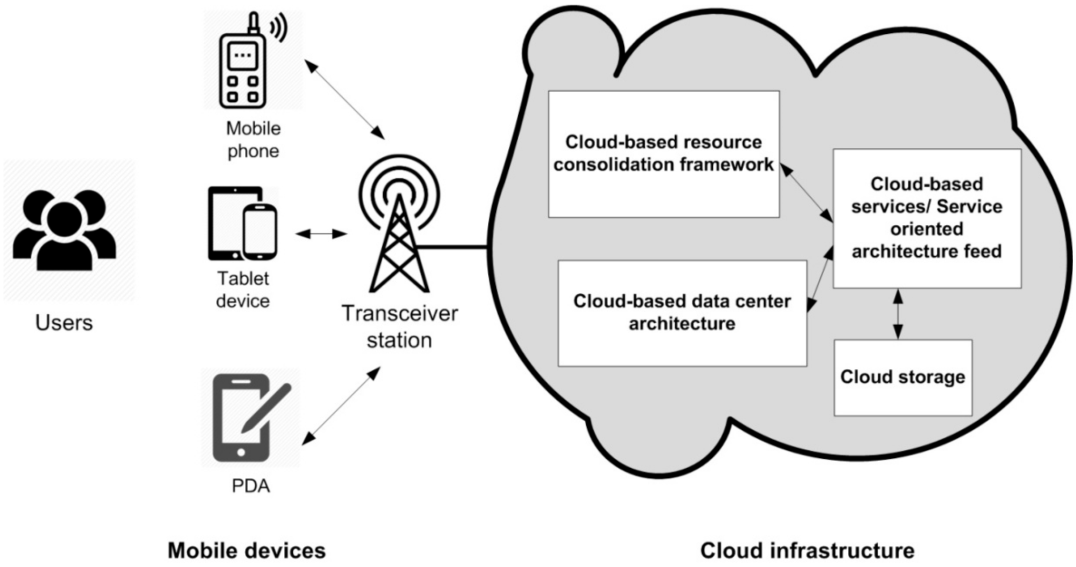

Mobile phone
Users
Tablet device
Transceiver station
PDA
Cloud-based resource consolidation framework
Cloud-based services/ Service oriented architecture feed
Cloud-based data center architecture
Cloud storage
Mobile devices
Cloud infrastructure

Figure 1. Mobile-oriented network infrastructure for cloud computing.

The main purpose of this paper is to contribute towards the field of SDCs by investigation of a major problem in cloud computing i.e. the placement of virtual machine (VM). VM placement is very important in datacenters and has been studied extensively, particularly for their use in the software-defined domain [20–22]. In a cloud environment, VMs are a key player as they provide the flexibility claimed by cloud service providers. Figure 1 presents the layout of VM in a distributed environment. When a computing service admitted into the cloud system demands for higher computational resources, then VM management plays a very important role. VM management helps in balancing the system constraints and loads [23,24]. Its main purpose is to retain user service satisfaction level. There are numerous VM placement challenges. Traditionally, the techniques for VM placement only focus on the resource allocation efficiency. Network research related to cloud resource management often focus on placement of VMs in data center environments. A vast number of VM placement techniques propose a solution based on available network resources [25]. This paper presents a relatively simpler approach for VM placement in the SDC environment. The concepts presented in this paper are related to the state-of-the-art technologies such as server and network resource utilization, software-defined networks, VM placement/ mapping and software-defined middle box networking. The paper presents a combination of these technologies for resource management in cloud environments.

The rest of the paper is further organized as follows. Section 2 discusses the related work, Section 3 presents the research allocation and mapping discussion in cloud environments, Section IV presents the mathematical modeling. In Section 4 we perform the performance evaluation. Finally, Section 5 concludes the paper.

2. Related Work

In cloud environments, resource sharing must be done in a way that a user's application requirements must not influence other user applications. Resource sharing must be done in such a way that these are secured and privately available [26]. VMs are acquired by applications on cloud infrastructure when needed. However, for cloud tenants, VM acquisition is a challenge. It is due to the limitations in cloud system's granularity and limitations in VM control and placement.

Data-intensive applications [27,28] frequently communicate with data centers. That is why the data traffic transmission of these applications is quite large. This results in network performance degradation and higher system overheads. VM placement strategies often use VM consolidation and reallocation techniques to solve vendor lock-in issues. These issues greatly influence network performance. In vendor lock-in issues, the users' traffic volumes can face delays. This ultimately leads to VM placement issues. The VM placement problem with traffic awareness [5,6] was proposed for solving these problems through network optimization based strategies.

In cloud environments, VMs follow certain patterns in accessing network resources. Research studies conducted in [29,30] involves a large number of CPU traces from different servers. It demonstrates that the demand traces are mostly in correlation and follow a periodic behavior. However, the concept of statistical multiplexing exists due to varying workloads. Data packet behavior for these applications relies primarily on the idea of exploiting possible correlations in VMs. Other approaches to vendor lock in issues include continuous monitoring of all VMs running on the network by using various VM measurement heuristics [31].

Current VM placement strategies have been extended for the inclusion of other data center infrastructure aspects such as network storage and network traffic. In a cloud infrastructure, all deployed VMs typically show a dependency on network traffic. The best optimization strategy to address their consolidation challenges is by hosting them on the nearest available physical machine [32,33]. Interestingly, network topology and data center design has a major impact on the selection of placement for traffic optimization targets [34]. Similar dependencies often occur for VMs and storage resources with different user requirements. In this situation, applications needing greater I/O performance can be moved closer to the storage locality.

Different vendors provide tools for resource-management functions. These tools include a wide range of applications. This includes system-level monitoring tools to application-level deep packet tracers and monitors. These sophisticated tools are a good choice; however, they slow down the system performance. Therefore, a vast-scale adoption of these tools will not only burden the network features, but will also influence underlying network resources (including virtual and infrastructure resources). In view of the above, a vendor-agnostic approach is used in [35,36] which proposes VM placement on a physical machine with the least data transfer time with respect to network bandwidth usage. However, within the datacenter premises, the data transmission rate is better due to wired communication. Therefore, users of these services expect high-qualityenterprise-level services rather than services offered by mobile devices with limited resources. Although not having enough tenant support for the VM migration, the cloud services provider have high control functions over all VMs locations. The manipulation of VMs can be performed by scaling in and out of physical resources.

Network support for tenant-controlledVM placement is difficult. An API-based SDN-enabled solution for these issues helps in providing a clean interface to the network administrator and is widely used in SDC environments.

SDNs [37,38] provide new possibilities for designing, operating, and securing data-intensive networks. However, the realization of these benefits largely requires the support of underlying infrastructure. In addition to handling the increased traffic loads, the network performance satisfaction opens new avenues of network services.

Mobile cloud computing based systems perform cloud computing functions with the exception that its users are mobile. Graph theory is a widely used concept in applied mathematics to structure pair wise models and relationships between objects. In this paper, we use graph-theoretic approach for resource consolidation on a vendor-agnostic hardware infrastructure which uses SDNs to administer network functions.

The proposed methodology is described by the formulation of a solution for VM placement that can be incorporated in SDCs. Currently, there is very limited support for VM placement in SDCs. For example, Amazon EC2 lacks support for co-locating its instance types. Although limited support features are present for cluster-based computational structures, high-performance features can only be afforded with premium prices [39,40]. Typically, the network resources available in close proximity are used for improved networking performance. It is believed that SDCs will be extensively used in future for flexibility and support in network applications and resource management.

3. Research Allocation and Mapping in Cloud Environments

Current mobility management schemes are based on centralized data access methodology. It is similar to the concepts used in traditional DC architectures. The main problem with this scheme is that it is difficult to manage. In terms of performance-based measures, these techniques results in routing and path optimization-related constraints which ultimately leads to performance degradation challenges.

The term vendor-agnostic refers to a concept where the products of a specific manufacturer are not tied to a particular vendor/brand etc. In distributed networks and systems theory, this term is often mingled with any off the self-solution. Vendor-agnostic solutions operate upon free, open-ended and generic solutions which involve basic mathematical optimization laws and principles not tied or related to a particular company. These solutions provide a clean interface to users for interacting with real-world problems. Our reason for highlighting vendor-agnostic behavior is based on the reason that we use a combination of open-ended hardware and software (via SDNs, graph-theory and Pareto-optimality) to achieve resource management functions.

In cloud environments, the rapid interaction between network's I/O devices, data and application services affect the system's overall performance [41]. The SDN concept to decouple data from control streams eases application and network performance. Here, we want to mention that SDN itself is an enabling technology. We need to employ SDN infrastructure for developing VM placement mechanism to achieve the desired goals. Therefore, we present a VM placement scheme for a SDC environment which can improve the service response time of applications.

We consider a software-defined cloud architecture where SDN based APIs administer the cloud resource management functions. These APIs manage topology and admission control features of cloud resources. Our framework's architecture is presented in Figure 2. Beneath the APIs lies a set of network manager and cloud manager. They control various functions of cloud e.g., mapping VMs, network statistics monitoring and controlling incoming outgoing packet requests. The last layer of the design architecture consists of virtual and physical resources.

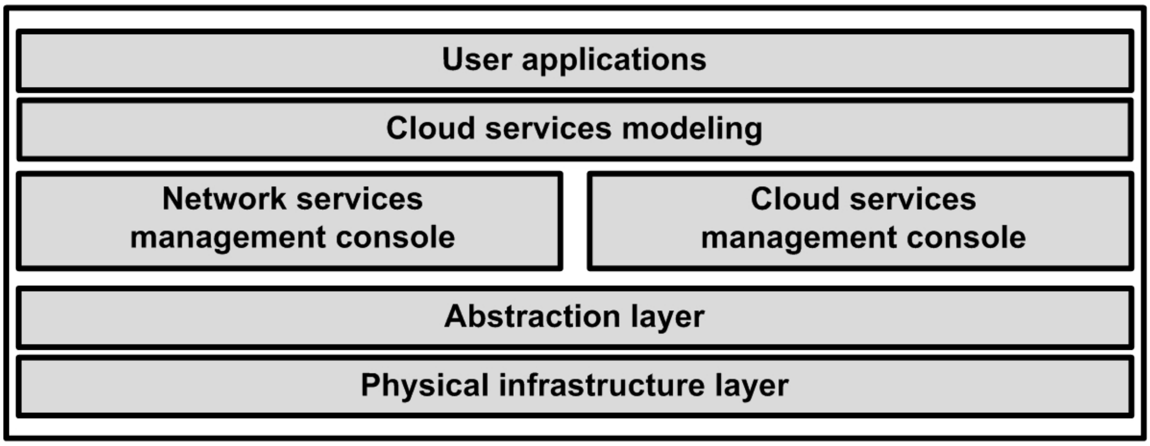

User applications
Cloud services modeling
Network services management console
Cloud services management console
Abstraction layer
Physical infrastructure layer

Figure 2. Proposed SDN cloud scenario.

SDN management APIs provide cloud resource management functions as high-level policies for the underlying network infrastructure. Such APIs help in managing and accessing an apparently infinite pool of computing resources like VMs etc. The function of the planner is to determine the location of hosting features for the received application requests in collaboration with cloud manager, modeler, and network manager. The modeler performs the comparison of received data and services from cloud planner and cloud manager. It is also used to model resource utilization features for updating network directory status. The network and cloud managers are used for managing virtual machines. The cloud and network managers on the other hand consolidate data at both physical and logical levels. The abstraction layer consists of logically-deployed physical hardware. Finally, the physical infrastructure layer consists of a list of physical resources that could be abstracted such as storage and network resources (routers or switches), servers, computing hosts, etc.

After sending off a request, the console of SDN manager makes sure that the made request is in compliance with the minimum number of SLAs. It then creates the topology of a blueprint. The topology information is later submitted to the admission controller. The admission controller validates and ensures that a connection can be established if current resources are sufficient for the proposed connection [42,43]. A simplistic approach of the performed sequential operation is illustrated in Figure 3.

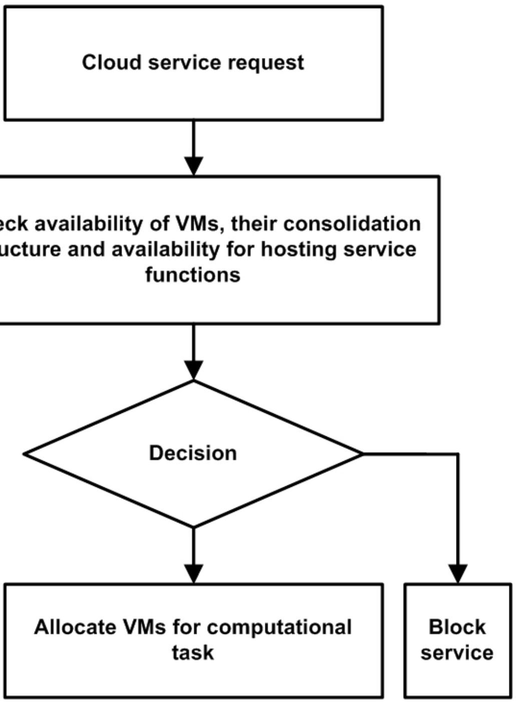

Cloud service request
Check availability of VMs, their consolidation structure and availability for hosting service functions
Decision
Allocate VMs for computational task
Block service

Figure 3. Virtual machine (VM) placement policy prototype in data center networks.

The location of hosting applications is determined by planner and modeler in consultation with cloud and network managers. Mapping of cloud resources is performed by the mapper. The proposed system performs VM placement. For ease of management, VM mapping should be controlled separately to ensure that cloud resources are managed in a clear and concise manner. The lower layer of SDC consists of different network resources. The layer for physical infrastructure contains any physical resource that could be abstracted e.g., storage and network resources (routers or switches), computing hosts, servers, etc. The abstraction layer provides abstraction information from a logical perspective. Conceptually physical layer resides beneath the abstraction layer [44,45].

In the proposed framework, by using graph theory, compute nodes are managed for allocation of VMs. In the proposed framework both virtual topologies and the physical infrastructure (switches, hosts, and links between them) are simulated for achieving dynamic routing features. In the presented scenario, all traffic patterns are supported by all the network elements. The assumptions in the proposed mechanism are mapped in a simulated environment for evaluation purpose.

The representation of the placement of VMs problem with the use of metrics (from linear algebra) is based on the fact that the cloud systems can be presented as a graph containing nodes and edges. Graph theory is the basic topic under discrete mathematics. Many advantages are present in the graph-theoretic approach.

On the basis of applied graph theory, we then manage compute nodes for VM allocation. In the proposed framework, both physical infrastructure and virtual topologies are simulated in CloudSim [46]. A detailed overview on latest trends and developments in the field of virtual resource management and network functions has been presented in [47].

4. Mathematical Modeling

We provide a mathematical representation of VM placement similar to [48] through our cloud model using a graph-theoretic approach. The entire interconnection between various entities of the proposed cloud is represented by adjacency matrices. Storage nodes (SN) represents data storage nodes. Compute nodes (CN) consist of multiple physical computational nodes, whereas data packet (DP) represents the data to be transmitted across the cloud. Our cloud infrastructure consists of 3 CNs, 2 SNs, and 3 DPs. Below we describe our model in detail. We consider a cloud system composed of m > 0 compute nodes (CN) and n > 0 storage nodes (SN). Please take note that the values of m and n are positive integers. The entire interconnection of the CNs and SNs can be depicted as a graph as shown in Figure 4.

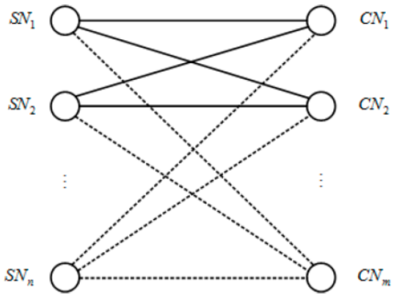


Figure 4. A graph-theoretic representation of the interconnection between the compute nodes (CN) and storage nodes (SN) forming a bipartite.

In discrete mathematics terminology (especially in graph theory), the graph shown in Figure 5 is known as a bipartite. A bipartite is a group of two sets of nodes where each member of each set is able to "communicate" with each and every member of the other set. The edges connecting the CN and SN may represent any relationship between these nodes. In order to limit and scale down the performance of our simulation, we assume that these edges could represent either bandwidth in MBps or time constant in secs/MB (which is just the reciprocal of the bandwidth). For example, the edge connecting SN1 to CN1 could represent the bandwidth value of 3.2 MBps or time constant of 0.3125 secs/MB (i.e., 1/3.2 MBps).

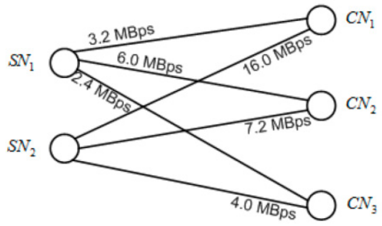


Figure 5. A graph-theoretic representation of a 2-SN, 3-CN cloud system given the values of the networks bandwidths between each combination of CN and SN nodes. This is an example of a 2×3 bipartite B3.2.

In applied graph theory, an adjacency matrix is a matrix that represents the values of all edges connected to the nodes in the graph. Consider the $n \times m$ adjacency matrix

$$\boldsymbol{B} = \left[b_{ij}\right]_{n\times m} = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1m} \\ b_{21} & b_{22} & \cdots & b_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nm} \end{bmatrix}, \tag{1}$$

where the matrix elements b$_{ij}$ for $1 \le i \le n$ and $1 \le j \le m$ are values representing the edges connecting node i to node j. As we consider a 2-CN, 3-SN cloud system as depicted in Figure 6, the network bandwidth between the nodes are as follows:

- SN1—CN1: 3.2 MBps;
- SN1—CN2: 6.0 MBps;
- SN1—CN3: 2.4 MBps;
- SN2—CN1: 16.0 MBps;
- SN2—CN2: 7.2 MBps; and
- SN2—CN3: 4.0 MBps;

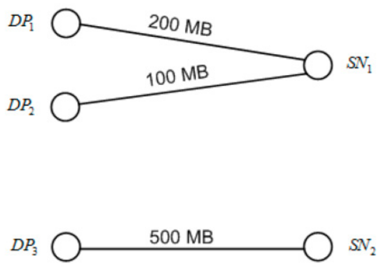


Figure 6. A graph representation of the relationship between the data pieces and the storage nodes. The first sub-graph is a 2 × 1 bipartite while the other sub-graph is a 1 × 1 bipartite (or simply a connection between two nodes).

Then the corresponding graph-theoretic representation will result in Figure 5 having the adjacency matrix with row i representing the SN number and column j representing the CN number.

$$\mathbf{B}_{3.2}\begin{bmatrix} 3.2 & 16.0 \\ 6.0 & 7.2 \\ 2.4 & 4.0 \end{bmatrix}\text{MBps}. \tag{2}$$

Similar to the approach in the previous section, a graph-theoretic approach can also be used to represent the relationship between the data pieces and SNs. In order to consider an environment two data pieces, DP1 = 200 MB and DP2 = 100 MB are both stored at storage node SN1, while another data piece, DP3 = 500 MB, is stored in SN2. The resulting graph shall be composed of two sub-graphs: one graph representing the relationship between DP1 and DP2 to SN1, and the relationship between DP3 to SN2. Please take note that each sub-graph is also a bipartite as shown in Figure 6. Since each data piece is stored only in a dedicated SN, it will be assumed in this architecture that the data piece is not shared between other SNs. Therefore, each sub-graph will only have one SN but can have multiple DPs.

The entire interconnection between the DP and SN can also be represented by an adjacency matrix where the column q shall represent the SN number. Since there are 2 SNs, then the matrix will have n = 2 columns. The number of rows of the adjacency matrix shall be equal to the maximum number of data pieces in any SN. In this particular example, since SN1 has two data pieces, namely DP1 and DP2, the number of rows shall be equal to p = 2. The resulting adjacency matrix becomes

$$\mathbf{D}_{2.2} = [\mathrm{d}_{\mathrm{ki}}]_{2\times 2} = \begin{bmatrix} \mathrm{DP}_1 & \mathrm{DP}_3 \\ \mathrm{DP}_2 & 0 \end{bmatrix} = \begin{bmatrix} 200 & 500 \\ 100 & 0 \end{bmatrix} \mathrm{MB}. \tag{3}$$

Consider for example a network bandwidth of b MBps. If data of size d MB will be transmitted into the network, then the response time can be obtained through

$$t = \frac{d}{b}[sec], \tag{4}$$

where simply the data and bandwidth are being adjusted with respect to time.

Similarly, if the network bandwidth b is inverted resulting in the time constant τ = 1/b (secs./MB), then the response time can be calculated using

$$t = d(MB).\ \tau\left(\frac{sec}{MB}\right)[sec], \tag{5}$$

where time and bandwidth are relating to resource requirements

However, this expression is only valid for scalar quantities, i.e., if there is only one data piece being processed by one CN through one SN. In fact, the notation "×" can be used here to represent scalar multiplication.

First, consider the graphical representation of the merger between $\mathrm{B}_{\mathrm{n,m}}$ and $\mathrm{D}_{\mathrm{p,n}}$ as shown in Figure 7. From the graphical representation, data flows between the SN and CN and DPs storage location can be seen. Using graph theory [16], it is possible to graphically represent networks using nodes and edges even if their quantities are different. The matrix can be defined as

$$\Upsilon_{n\times m} = \left[\tau_{ij}\right]_{n\times m} \tag{6}$$

where the values τ_{ij} represent the time constants between the SN and CN. Let us call this the time constant matrix. Basically, the values in this matrix are just the reciprocals of the bandwidths, therefore the following mathematical expression

$$\left[\tau_{ij}\right]_{n\times m} = \left[\frac{1}{b_{ij}}\right]_{n\times m} \tag{7}$$

where cross matrix multiplication ensures SN to CN matrix mapping shall apply for $1 \le i \le n$ and $1 \le j \le m$. Given the data set matrix **D**p,n we can now get the response times for each data piece in various CNs.

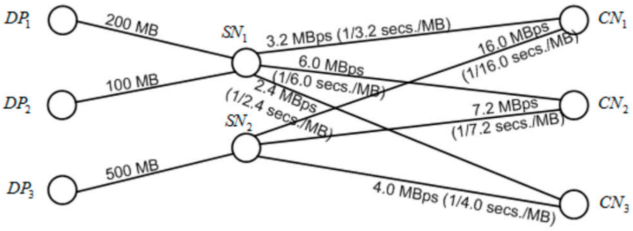

DP_1
DP_2
DP_3
200 MB
100 MB
500 MB
SN_1
SN_2
3.2 MBps (1/3.2 secs./MB)
6.0 MBps (1/6.0 secs./MB)
2.4 MBps (1/2.4 secs./MB)
16.0 MBps (1/16.0 secs./MB)
7.2 MBps (1/7.2 secs./MB)
4.0 MBps (1/4.0 secs./MB)
CN_1
CN_2
CN_3

Figure 7. Graph representing entire cloud mapping.

The series of manual computations above are easily done due to the small dimensions of the cloud system. Since real cloud systems have hundreds of thousands of SN and CN, it will be impossible for us to have all the combinations and compute them manually. It is now important to have everything done with a computer through linear algebra. In order to obtain the total response times tCN,1, tCN,2 and tCN,3 we need the following step:

(1) Step 1: get the time constant matrix from the bandwidth matrix: Let

$$\mathrm{B}_{3.2} = \begin{bmatrix} 3.2 & 16.0 \\ 6.0 & 7.2 \\ 2.4 & 4.0 \end{bmatrix} \text{MBps} \tag{8}$$

be the bandwidth matrix of the cloud system. The time constants for each element in the matrix can be obtained by simply getting the reciprocals of each element. The resulting matrix becomes

$$\mathrm{B}_{3.2} = \begin{bmatrix} 1/3.2 & 1/16.0 \\ 1/6.0 & 1/7.2 \\ 1/2.4 & 1/4.0 \end{bmatrix} = \begin{bmatrix} 0.3125 & 0.6250 \\ 0.1667 & 0.1389 \\ 0.4167 & 0.2500 \end{bmatrix} \sec/\text{MB}. \tag{9}$$

(2) Step 2: get the transpose of the time constant matrix: Given the time constant matrix $\Upsilon_{3.2}$, its transpose can be obtained as

$$\Upsilon_{3.2}^{T} = \begin{bmatrix} 0.3125 & 0.6250 \\ 0.1667 & 0.1389 \\ 0.4167 & 0.2500 \end{bmatrix} = \begin{bmatrix} 0.3125 & 0.1667 & 0.4167 \\ 0.6250 & 0.1389 & 0.2500 \end{bmatrix} \sec/\text{MB}^{T}. \tag{10}$$

(3) Step 3: multiply the data set matrix with the transposed time constant matrix: Given the data set matrix

$$\mathbf{D}_{2.2} = \begin{bmatrix} 200 & 500 \\ 100 & 0 \end{bmatrix} \text{MB}. \tag{11}$$

The response time matrix can then be obtained as follows

$$T_R = \mathbf{D}_{2.2}\Upsilon_{3.2}^{T} = \begin{bmatrix} DP_1 & DP_3 \\ DP_2 & 0 \end{bmatrix} \begin{bmatrix} \tau_{11} & \tau_{12} \\ \tau_{21} & \tau_{22} \\ \tau_{31} & \tau_{32} \end{bmatrix} = \begin{bmatrix} DP_1 & DP_3 \\ DP_2 & 0 \end{bmatrix} \begin{bmatrix} \tau_{11} & \tau_{21} & \tau_{31} \\ \tau_{12} & \tau_{22} & \tau_{32} \end{bmatrix}^{T} \tag{12}$$

$$T_R = \begin{bmatrix} 200 & 500 \\ 100 & 0 \end{bmatrix} \begin{bmatrix} 0.3125 & 0.1667 & 0.4167 \\ 0.6250 & 0.1389 & 0.2500 \end{bmatrix} \tag{13}$$

$$T_R = \begin{bmatrix} (t_{11} + t_{31}) & (t_{12} + t_{32}) & (t_{13} + t_{33}) \\ (t_{21}) & (t_{22}) & (t_{23}) \end{bmatrix} = \begin{bmatrix} t_{R,11} & t_{R,12} & t_{R,13} \\ t_{R,21} & t_{R,22} & t_{R,23} \end{bmatrix}. \tag{14}$$

One key characteristic of the network response time matrix $\mathrm{T_R}$ is that if you get the sum of all elements per column, you actually obtain the total response time for each and every CN. Each column of $\mathrm{T_R}$ represents each CN. Since, in this example, there are 3 CNs in the cloud system, $\mathrm{T_R}$ results in a matrix having 3 columns as well. If the matrix is denoted by

$$\mathbf{T}_{\mathrm{R}} = \begin{bmatrix} t_{\mathrm{R},11} & t_{\mathrm{R},12} & t_{\mathrm{R},13} \\ t_{\mathrm{R},21} & t_{\mathrm{R},22} & t_{\mathrm{R},23} \end{bmatrix}, \tag{15}$$

The total response time per CN can be obtained using the expression

$$\mathrm{t}_{\mathrm{CN},i} = \sum\nolimits_{j=1}^{n} t_{\mathrm{R},ji}\,, \tag{16}$$

Therefore,

$$\mathrm{t}_{\mathrm{CN},1} = \sum\nolimits_{j=1}^{n=2} t_{\mathrm{R,j1}} = t_{\mathrm{R},11} + t_{\mathrm{R},21} \tag{17}$$

$$\mathrm{t}_{\mathrm{CN},2} = \sum\nolimits_{j=1}^{n=2} t_{\mathrm{R,j2}} = t_{\mathrm{R},12} + t_{\mathrm{R},22} \tag{18}$$

$$\mathrm{t}_{\mathrm{CN},3} = \sum_{j=1}^{n=2} t_{\mathrm{R,j3}} = t_{\mathrm{R,13}} + t_{\mathrm{R,23}} \tag{19}$$

Real-world optimization issues can be implemented on cloud-based systems to involve multiple conflicting objectives. Therefore, a vector-optimization problem in a standardized manner can be represented as a standardized vector $X = (x_1, x_1, \ldots, x_n)$. A Pareto-optimal solution [49,50] for resource existence (when no other solution exists) is represented in Figure 8. This helps in ensuring that one objective (resource allocation) can be improved without affecting the other objective.

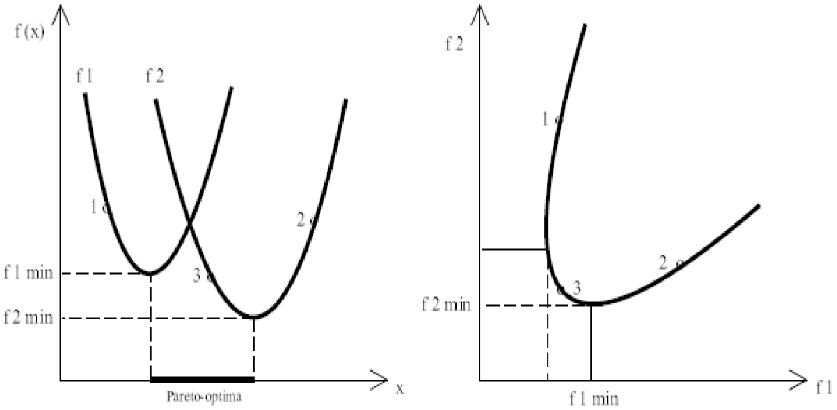


Figure 8. The Pareto-optimal solution where one objective can be improved without the expense of others.

5. Performance Evaluation

In this section, we explain the simulations and experiments carried to evaluate the proposed resource consolidation approach. We used a channel model approach similar to presented in [51,52], which is widely used for mobile cloud and cellular networks. The testbed simulation consisted of two storage nodes, three data sets, and three mobile hosts. The mobile devices map their resources by using a graph-theoretic model as explained in previous sections and use SDN-based infrastructure for controlling data and traffic behavior functions. In this regard, we implemented our algorithm on CloudSim v 3.0. The Cloudsim is often used as an extensible simulation toolkit for simulation purposes.

Algorithm 1 VM Placement

```
$CN$ denotes the Computes of cloud
$t_{CN,i}$ denotes the response time value of $CNi$
$least$ denotes least response time value of $CNs$
$j$ denotes CN having least response time value
For calculating $t_R$ for each $CN$, we have
$t_{CN,i} = \sum_{j=1}^{n} t_{R,\,ji}$
$i \leftarrow 0$
$j \leftarrow 0$
$least \leftarrow t_{CN,0}$
while $i < n$ do
    if $least > t_{CN,i}$:then
        $least \leftarrow t_{CN,i}$
        $j \leftarrow i$
    end if
    $i \leftarrow i+1$
end while
Choose VM location at CN $j$
Exit
```

We developed an algorithm (Algorithm 1) for virtual machine placement on a particular cloud node. It works by calculating service response time of individual compute nodes and then selecting VM with least response time. By using the proposed Algorithm 1, we compute the service response time T_R of individual *CN* and select a *CN* having least response time to host VM. We then calculated the response time of these data loads using vmallocationpolicysimple algorithm [53,54]. We selected the vmallocationpolicysimple algorithm because of two major reasons. Firstly, it does not implement dynamic consolidation of VMs, and only places new VMs on hosts; fulfilling our scenario's demand. Secondly, it is the default VM placement strategy in CloudSim. Finally, we compared the service response time of algorithm 1 with that of vmallocationpolicysimple algorithm for given workloads. The evaluation setting is similar to emulate the environment presented in Figure 7.

Simulation results in Figure 9 show the service response time for tasks requesting DP1 data load. In Figures 10 and 11, we illustrate the service response time for tasks requesting DP2 and DP3 respectively. The graphics illustrations reveal that our proposed scheme demonstrates improved service response time to requests as compared to vmallocationpolicysimple algorithm. It is because the presented algorithm clearly chooses a CN with reduced response time and shorter data access route for VM allocation.

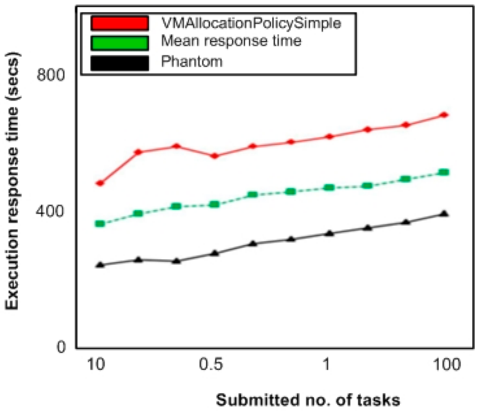


Figure 9. Service response time (DP1 requesting tasks).

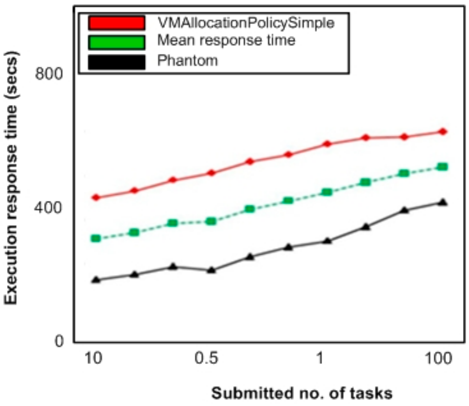


Figure 10. Service response time (DP2 requesting tasks).

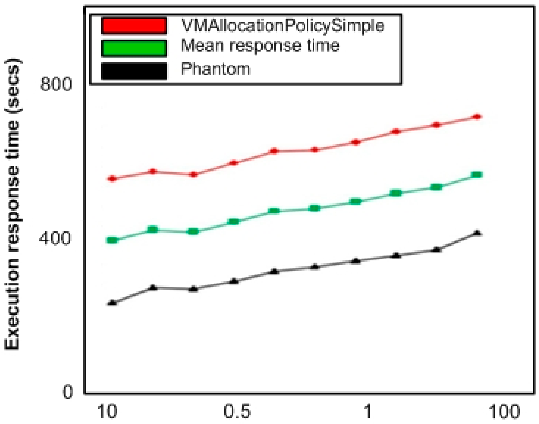
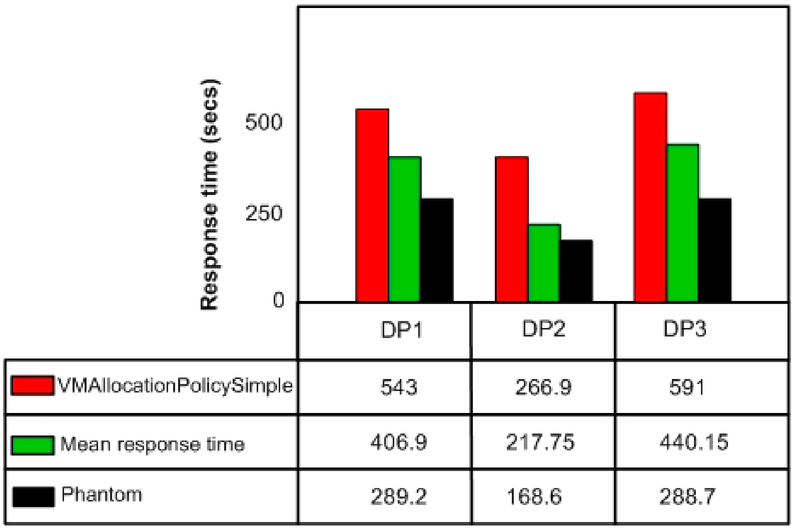


Figure 11. Service response (DP3 requesting tasks).

By observing the comparative results in Figure 12, it can be observed that the traffic intensity for DP1, DP2, and DP3 exhibit the same behavior of variance with respect to mean response time. However, the difference in values with respect to the vmallocationpolicysimple is different. This also resulted in increased variance rates of bandwidth consumption. In order to retain the job queue waiting time, we can manage the waiting-time window slot timing. If the received task arrives which can be completed in a relatively long period of time, the mapping scheme can adjust itself according to window time and accommodates more tasks as compared to the vmallocationpolicysimple strategy. The same concept can be improved with the predictive analysis however we didn't consider it due to overhead costs incurred on VM loads.



	DP1	DP2	DP3
🟥 VMAllocationPolicySimple	543	266.9	591
🟩 Mean response time	406.9	217.75	440.15
⬛ Phantom	289.2	168.6	288.7

Figure 12. Cumulative comparison of service response time for 100 requests.

6. Conclusions and Future Work

Future mobile-oriented networks are taking the computing industry by storm. The recent developments witnessed in the enhanced computational capability of mobile equipment led to the concepts of mobile cloud computing. Taking this paradigm to another step, in this paper we presented a case where cloud data centers are managed in a mobile cloud environment. We begin the paper by explaining the importance of mobile future network architectures followed by the concepts of resource management in mobile clouds using a vendor-agnostic approach (through SDNs).

To sum up the concept of the paper, we argue that the cloud computing concept involves the availability of computing resources for data storage and processing. Due to increasing number of network applications, number of users and their requirements, there is a dire need to develop tools to improve cloud computing performance. On the other hand, software-defined networking concept allows cloud data center administers to manage cloud resource allocation function according to their own needs via bypassing proprietary network peripherals. As SDN concepts discourage excessive use of proprietary equipment, it is often referred as a bare-metal solution, off-the-shelf solution and vendor-agnostic solution).

The paper relates SDN-based mobile cloud environment to propose a VM mapping policy using a graph-theoretic approach. The reason for calling this technique vendor-agnostic is the use of a vendor-agnostic platform (SDN-enabled H/W) for evaluation purpose.

In this paper, the simple representation of VM resource allocation and representation helped in clearly determine the network management by the use of matrices. A vendor-agnostic-based approach, therefore, offers several advantages over conventional approaches. These advantages can be seen particularly in distributed systems like cloud computing environments. Therefore, we implemented this approach to consolidate VM resources in a simplistic and well-organized way. We believe that graphs can be represented using adjacency matrices where each element of the matrix denotes values that show relationships between any two nodes. Therefore, we used a graph-theoretic approach to achieve our consolidation approach. We developed a framework and compared its performance with the vmallocationpolicysimple technique. Results demonstrate that our proposed framework can limit the cloud topology scaling issues of VM placement in a more clearer and concise manner. We strongly believe that a vendor-agnostic approach in data centers can be considered as the next step towards the evolution of virtualization, mobile cloud computing, and future mobile-oriented networks.

Author Contributions: Conceptualization, A.A.A.; data curation, A.A.A. and M.A.A.A.-q.; formal analysis, M.A.A.A.-q. and A.H.; investigation, A.A.A. and A.H.; methodology, M.A.E. and A.A.E.; resources, M.A.E. and S.J.; software, A.A.E.; supervision, S.K.; writing—original draft, A.A.A.; writing—review and editing, A.A.A. and S.J. All authors have read and approved the manuscript.

Funding: Following are results of a study on the "Leaders in INdustry-university Cooperation+" project, supported by the Ministry of Education and National Research Foundation of Korea.

Acknowledgments: The authors thank the anonymous reviewers for their thorough feedback and suggestions, which were crucial to improving the contents and presentation of the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kim, J.I.; Choi, N.J.; You, T.W.; Jung, H.; Kwon, Y.W.; Koh, S.J. Mobile-Oriented Future Internet: Implementation and Experimentations over EU–Korea Testbed. *Electronics* **2019**, *8*, 338. [CrossRef]
2. Kim, J.I.; Jung, H.; Koh, S.J. Mobile oriented future internet (MOFI): Architectural design and implementations. *ETRI J.* **2013**, *35*, 666–676. [CrossRef]
3. Jung, H.Y.; Koh, S.J. *Mobile-Oriented Future Internet (MOFI): Architecture and Protocols*; ETRI: Daejeon, Korea, 2010.
4. Boukerche, A.; Guan, S.; Grande, R.E.D. Sustainable Offloading in Mobile Cloud Computing: Algorithmic Design and Implementation. *ACM Comput. Surv.* **2019**, *52*, 11. [CrossRef]
5. Chaudhry, S.A.; Kim, I.L.; Rho, S.; Farash, M.S.; Shon, T. An improved anonymous authentication scheme for distributed mobile cloud computing services. *Clust. Comput.* **2019**, *22*, 1595–1609. [CrossRef]
6. Ahmed, A.A.; Alzahrani, A.A. A comprehensive survey on handover management for vehicular ad hoc network based on 5G mobile networks technology. *Trans. Emerg. Telecommun. Technol.* **2019**, *30*, e3546. [CrossRef]
7. Kushwah, R.; Tapaswi, S.; Kumar, A. A detailed study on Internet connectivity schemes for mobile ad hoc network. *Wirel. Pers. Commun.* **2019**, *104*, 1433–1471. [CrossRef]
8. Garcia, A.J.; Toril, M.; Oliver, P.; Luna-Ramirez, S.; Garcia, R. Big Data Analytics for Automated QoE Management in Mobile Networks. *IEEE Commun. Mag.* **2019**, *57*, 91–97. [CrossRef]
9. Liu, K.; Zha, Z.; Wan, W.; Aggarwal, V.; Fu, B.; Chen, M. Optimizing TCP Loss Recovery Performance Over Mobile Data Networks. *IEEE Trans. Mob. Comput.* **2019**. [CrossRef]
10. Aceto, G.; Ciuonzo, D.; Montieri, A.; Pescapé, A. Mobile encrypted traffic classification using deep learning: Experimental evaluation, lessons learned, and challenges. *IEEE Trans. Netw. Serv. Manag.* **2019**. [CrossRef]

11. Misra, S.; Wolfinger, B.E.; Achuthananda, M.P.; Chakraborty, T.; Das, S.N.; Das, S. Auction-Based Optimal Task Offloading in Mobile Cloud Computing. *IEEE Syst. J.* **2019**. [CrossRef]
12. Moreno-Vozmediano, R.; Huedo, E.; Montero, R.S.; Llorente, I.M. A Disaggregated Cloud Architecture for Edge Computing. *IEEE Internet Comput.* **2019**, *23*, 31–36. [CrossRef]
13. Chen, D.; Zhang, X.; Wang, L.L.; Han, Z. Prediction of Cloud Resources Demand Based on Hierarchical Pythagorean Fuzzy Deep Neural Network. *IEEE Trans. Serv. Comput.* **2019**. [CrossRef]
14. Ahmed, E.; Naveed, A.; Gani, A.; Ab Hamid, S.H.; Imran, M.; Guizani, M. Process state synchronization-based application execution management for mobile edge/cloud computing. *Future Gener. Comput. Syst.* **2019**, *91*, 579–589. [CrossRef]
15. Agrawal, N.; Shashikala, T. A trustworthy agent-based encrypted access control method for mobile cloud computing environment. *Perv. Mob. Comput.* **2019**, *52*, 13–28. [CrossRef]
16. Sharma, Y.; Weisheng, S.; Daniel, S.; Bahman, J. Failure-aware energy-efficient VM consolidation in cloud computing systems. *Future Gener. Comput. Syst.* **2019**, *94*, 620–633. [CrossRef]
17. Minh, Q.T.; Dang, T.K.; Nam, T.; Kitahara, T. Flow aggregation for SDN-based delay-insensitive traffic control in mobile core networks. *IET Commun.* **2019**, *13*, 1051–1060. [CrossRef]
18. Wu, W.; Liu, J.; Huang, T. Decoupled delay and bandwidth centralized queue-based QoS scheme in OpenFlow networks. *China Commun.* **2019**, *16*, 70–82.
19. Shamshirband, S.; Hossein, S. LAAPS: An efficient file-based search in unstructured peer-to-peer networks using reinforcement algorithm. *Int. J. Comput. Appl.* **2018**. [CrossRef]
20. Abbasi, A.A.; Al-qaness, M.A.; Elaziz, M.A.; Khalil, H.A.; Kim, S. Bouncer: A Resource-Aware Admission Control Scheme for Cloud Services. *Electronics* **2019**, *8*, 928. [CrossRef]
21. Abbasi, A.A.; Abbasi, A.; Shamshirband, S.; Chronopoulos, A.T.; Persico, V.; Pescapè, A. Software-Defined Cloud Computing: A Systematic Review on Latest Trends and Developments. *IEEE Access* **2019**, *7*, 93294–93314. [CrossRef]
22. Jin, H.; Abbasi, A.A.; Wu, S. Pathfinder: Application-aware distributed path computation in clouds. *Int. J. Parallel Program.* **2017**, *45*, 1273–1284. [CrossRef]
23. Priya, B.; Gnanasekaran, T. To optimize load of hybrid P2P cloud data-center using efficient load optimization and resource minimization algorithm. *Pee Peer Netw. Appl.* **2019**. [CrossRef]
24. Curino, C.; Subru, K.; Konstantinos, K.; Sriram, R.; Giovanni, M.F.; Botong, H.; Kishore, C.; Arun, S.; Chen, Y.; Heddaya, S.; et al. Hydra: A federated resource manager for data-center scale analytics. In Proceedings of the 16th USENIX Symposium on Networked Systems Design and Implementation (NSDI '19), Boston, MA, USA, 26–28 February 2019; NetApp: Sunnyvale, CA, USA, 2019; pp. 177–192.
25. Asghari, S.; Navimipour, N.J. Resource discovery in the peer to peer networks using an inverted ant colony optimization algorithm. *Peer Peer Netw. Appl.* **2019**, *12*, 129–142. [CrossRef]
26. Shamshirband, S.; Chronopoulos, A.T. A new malware detection system using a high performance-ELM method. In Proceedings of the 23rd International Database Applications & Engineering Symposium, ACM, Athens, Greece, 10–12 June 2019; p. 33.
27. Shuib, L.; Shamshirband, S.; Ismail, M.H. A review of mobile pervasive learning: Applications and issues. *Comput. Hum. Behav.* **2015**, *46*, 239–244. [CrossRef]
28. Stan, R.G.; Catalin, N.; Florin, P. Cloudwave: Content gathering network with flying clouds. *Future Gener. Comput. Syst.* **2019**, *98*, 474–486. [CrossRef]
29. Lin, L.; Liu, X.; Ma, R.; Li, J.; Wang, D.; Guan, H. vSimilar: A high-adaptive VM scheduler based on the CPU pool mechanism. *J. Syst. Archit.* **2019**. [CrossRef]
30. Kalogirou, C.; Koutsovasilis, P.; Antonopoulos, C.D.; Bellas, N.; Lalis, S.; Venugopal, S.; Pinto, C. Exploiting CPU Voltage Margins to Increase the Profit of Cloud Infrastructure Providers. In Proceedings of the 2019 19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), Larnaca, Cyprus, 14–17 May 2019; pp. 302–311.
31. Moges, F.F.; Abebe, S.L. Energy-aware VM placement algorithms for the OpenStack Neat consolidation framework. *J. Cloud Comput.* **2019**, *8*, 2. [CrossRef]
32. Le, F.; Nahum, E.M. Experiences Implementing Live VM Migration over the WAN with Multi-Path TCP. In Proceedings of the IEEE Infocom 2019 IEEE Conference on Computer Communications, Paris, France, 29 April–2 May 2019; pp. 1090–1098.

33. Mohiuddin, I.; Ahmad, A. Workload aware VM consolidation method in edge/cloud computing for IoT applications. *J. Parallel Distrib. Comput.* **2019**, *123*, 204–214. [CrossRef]
34. Zhao, H.; Han, G.; Niu, X. The Signal Control Optimization of Road Intersections with Slow Traffic Based on Improved PSO. *Mob. Netw. Appl.* **2019**. [CrossRef]
35. Guerrero, C.; Isaac, L.; Carlos, J. A lightweight decentralized service placement policy for performance optimization in fog computing. *J. Ambient Intell. Humaniz. Comput.* **2019**, *10*, 2435–2452. [CrossRef]
36. Badawy, M.; Hisham, K.; Hesham, A. New approach to enhancing the performance of cloud-based vision system of mobile robots. *Comput. Electr. Eng.* **2019**, *74*, 1–21. [CrossRef]
37. Lin, F.P.C.; Tsai, Z. Hierarchical Edge-Cloud SDN Controller System with Optimal Adaptive Resource Allocation for Load-Balancing. *IEEE Syst. J.* **2019**. [CrossRef]
38. Chirivella-Perez, E.; Marco-Alaez, R.; Hita, A.; Serrano, A.; Alcaraz Calero, J.M.; Wang, Q.; Neves, P.M.; Bernini, G.; Koutsopoulos, K.; Martínez Pérez, G.; et al. SELFNET 5G mobile edge computing infrastructure: Design and prototyping. *Softw. Pract. Exp.* **2019**. [CrossRef]
39. Ma, X.; Wang, S.; Zhang, S.; Yang, P.; Lin, C.; Shen, X.S. Cost-Efficient Resource Provisioning for Dynamic Requests in Cloud Assisted Mobile Edge Computing. *IEEE Trans. Cloud Comput.* **2019**. [CrossRef]
40. Tasiopoulos, A.; Ascigil, O.; Psaras, I.; Toumpis, S.; Pavlou, G. FogSpot: Spot Pricing for Application Provisioning in Edge/Fog Computing. *IEEE Trans. Serv. Comput.* **2019**. [CrossRef]
41. Rehman, A.; Hussain, S.S.; ur Rehman, Z.; Zia, S.; Shamshirband, S. Multi-objective approach of energy efficient workflow scheduling in cloud environments. *Concurr. Comput. Pract. Exp.* **2019**, *31*, e4949. [CrossRef]
42. Al-Dulaimi, A.; Mumtaz, S.; Al-Rubaye, S.; Zhang, S.; Chih-Lin, I. A Framework of Network Connectivity Management in Multi-Clouds Infrastructure. *IEEE Wirel. Commun.* **2019**. [CrossRef]
43. Park, S.T.; Oh, M.R. An empirical study on the influential factors affecting continuous usage of mobile cloud service. *Clust. Comput.* **2019**, *22*, 1873–1887. [CrossRef]
44. Zhou, Y.; Tian, L.; Liu, L.; Qi, Y. Fog computing enabled future mobile communication networks: A convergence of communication and computing. *IEEE Commun. Mag.* **2019**, *57*, 20–27. [CrossRef]
45. Elhabbash, A.; Faiza, S.; James, H.; Yehia, E. Cloud brokerage: A systematic survey. *ACM Comput. Surv.* **2019**, *51*, 119. [CrossRef]
46. Barbierato, E.; Gribaudo, M.; Iacono, M.; Jakóbik, A. Exploiting CloudSim in a multiformalism modeling approach for cloud based systems. *Simul. Model. Pract. Theory* **2019**, *93*, 133–147. [CrossRef]
47. Laghrissi, A.; Taleb, T. A survey on the placement of virtual resources and virtual network functions. *IEEE Commun. Surv. Tutor.* **2018**, *21*, 1409–1434. [CrossRef]
48. Piao, J.T.; Yan, J. A network-aware virtual machine placement and migration approach in cloud computing. In Proceedings of the 2010 Ninth International Conference on Grid and Cloud Computing, Nanjing, China, 1–5 November 2010; pp. 87–92.
49. Blasco, X.; Reynoso-Meza, G.; Sánchez-Pérez, E.A.; Sánchez-Pérez, J.V. Computing optimal distances to pareto sets of multi-objective optimization problems in asymmetric normed lattices. *Acta Appl. Math.* **2019**, *159*, 75–93. [CrossRef]
50. Wang, S.; Zhao, Y.; Xu, J.; Yuan, J.; Hsu, C.H. Edge server placement in mobile edge computing. *J. Parallel Distrib. Comput.* **2019**, *127*, 160–168. [CrossRef]
51. Hong, S.T.; Kim, H. QoE-aware Computation Offloading to Capture Energy-Latency-Pricing Tradeoff in Mobile Clouds. *IEEE Trans. Mob. Comput.* **2018**. [CrossRef]
52. Lei, L.; Xu, H.; Xiong, X.; Zheng, K.; Xiang, W. Joint Computation Offloading and Multi-User Scheduling using Approximate Dynamic Programming in NB-IoT Edge Computing System. *IEEE Internet Things J.* **2019**. [CrossRef]
53. CloudSim. Available online: http://www.cloudbus.org/cloudsim/doc/api/org/cloudbus/cloudsim/power/PowerVmAllocationPolicySimple.html (accessed on 7 October 2019).
54. Jammal, M.; Hawilo, H.; Kanso, A.; Shami, A. Generic input template for cloud simulators: A case study of CloudSim. *Softw. Pract. Exp.* **2019**, *49*, 720–747. [CrossRef]