



# Modelling Virtual Machine Workload in Heterogeneous Cloud Computing Platforms

Suliman Mohamed Fati, Prince Sultan University, Saudi Arabia

 <https://orcid.org/0000-0002-6969-2338>

Ayman Kamel Jaradat, Al Majmaah University, Saudi Arabia

Ibrahim Abunadi, Prince Sultan University, Saudi Arabia

 <https://orcid.org/0000-0002-2546-2450>

Ahmed Sameh Mohammed, Prince Sultan University, Saudi Arabia

## ABSTRACT

Cloud computing, as a trend technology, has stemmed from the concept of virtualization. Virtualization makes the resources available to the public to use without any concern for ownership or maintenance cost. In addition, the hosted applications in cloud computing platforms are highly interactive and require intensive resources. The new trend is to duplicate these applications in multiple virtual machines based on demand. Such a scheme needs an efficient resource provisioning to manage the resource assignment to multiple virtual machines properly. One of the issues in the current resource provisioning technique is assigning the resources proactively without predicting the workload of hosted applications, which cause load imbalance and resource wasting. Thus, this paper proposes a new model to predict the application workload. The experimental results show the ability of the proposed model to allocate more virtual machines and to balance the workload among the physical machines.

## KEYWORDS

Cloud Computing, Resource Allocation, Virtual Machine, Virtual Machine Status, Workload Prediction

## INTRODUCTION

Due to its unique characteristics, Cloud computing became a trend technology and gained a huge attention. Out of these distinctive characteristics are cost-effectiveness, elasticity, resource pooling, measured services, and adaptability (Armbrust et al., 2010). The importance of cloud computing comes from the concept of utility computing, which allows the people to utilizing the computing resources in the same way of utilizing the utility services (e.g. electricity, water, gas, etc.) (Buyya, Yeo, Venugopal, Broberg, & Brandic, 2009). In cloud computing, the user can rent the computing resources instead of owning it and pay based on his/her consumption.

The idea Cloud computing comes from the abundance of computing resources in the datacenters, which are surplus. Such huge resources represents the resource pooling that can be rented to the

DOI: 10.4018/JITR.20201001.oa1

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

others upon the user's demands (Pandya & Bheda, 2014). Elevating cloud computing as a prominent technology is due to many enabler technologies such as Service Oriented Architecture (SOA) and virtualization (L. Wang et al., 2010). SOA allows the user to use the resources as a service, while virtualization allows cloud computing provider to offer the single resource to multiple users based on multi-tenancy concept (AlJahdali et al., 2014). Consequently, these computing resources can be offered to the public over the internet to be used on demand basis. Cloud computing can be offered in different service models based on the offered services such as infrastructure as a service (IaaS), Platform as a service (PaaS), Software as a service (SaaS). In IaaS, the hardware is virtualized and offered to the users as virtual resources. Rackspace, Amazon Web Services (AWS), Cisco Metapod famous examples of IaaS paradigm. Platform as a service (PaaS) offered the development tools to the professional people to design their systems. Microsoft Azure and Google Compute Engine (GCE) represent this model. Besides, Google apps, Google sheets, and Google Docs are examples of SaaS, wherein the software application will be offered to the end users without any customization. The main advantage of SaaS is to avoid the burden of software licensing, batch updates, and operating systems independency. Other recent service models have been invented recently as Security as a service (SECaaS), Business Process as a service (BPaaS), Containers-as-a-Service (CaaS), and so on. Thus, the concept of anything as a service (XaaS) was introduced to represent the possibility of offering any service as cloud-based (Rimal, Choi, & Lumb, 2009).

Despite the cost-effectiveness of cloud computing, the cost reduction is still one of the service providers' worries. In service provider side, cost reduction can be achieved throughout an effective resources provisioning process (Joe, Yi, & Sohn, 2011). Resource provisioning is an essential feature in cloud computing whereby estimating the computing resources for different computation tasks based on the expected/real workload. Accordingly, resource allocation process determines the locations and the amount of resources needed for each user to minimize the cost under certain constraints. In cloud computing, many users may share resources with different needs and expectation. Each user expects to use the resources freely and with a certain level of quality. Besides, each virtual machine is characterized with variable size, interactivity, and rapidly changing effective period (Rimal et al., 2009). Such characteristics make the design of cloud computing platform challengeable. For that, these resources should be shared in a more effective manner (Patel, Ranabahu, & Sheth, 2009). However, the virtual machines hosted in cloud computing platforms overwhelmingly suffer from the workload fluctuation and/or sudden peak workloads, which may cause a degradation in quality of service (QoS) and quality of experience (QoE).

Particularly, the fluctuation in the workload (i.e. the number of incoming requests targeting the hosted virtual machines) is a serious issue threatens cloud computing (Patel et al., 2009). For instance, allocating a large amount of resources for a virtual machine to cope with the sudden workload leads to low resources utilization at non-peak hours, while ignoring the peak workload leads to user dissatisfaction during the peak hours. Therefore, pay-as-you-go scheme is introduced as an elasticity feature to solve the workload fluctuation issues by scaling up/down the allocated resources based on the user demand immediately to avoid service interruption or resource wastage. Amazon Cloud Front is a global example of this scheme. In this scheme, the user has to pay for the used resources only without any up-front, commit, or service contract.

Although its efficiency to react to the user needs by allocating the needed resources timely, pay-as-you go scheme still suffers from the resource wastage due to the lack workload prediction. Thus, predicting the status of the virtual machines based on their characteristics is a crucial factor to affiliate the issues of load imbalance and waste of resources (Vicari, 2008). Virtual machine status prediction can be defined as estimating the portion of concurrent requests that target the VM according to its characteristics. Such estimation helps a lot in handling both the load balancing and resource allocation based on the anticipated load of applications and virtual machines (Li & Wu, 2010).

According to the authors in (Fati & Sumari, 2018; Fati, Sumari, Yuhaniz, & Sjarif, n.d.; A. Gaber, Mohamed, Sumari, & Budiarto, 2012; García, Pañeda, Melendi, & García, 2009), modeling

the workload for is essential factor to improve the performance, enhance the QoS, and increase the reliability. Therefore, the idea of this paper is to build a mathematical model that formulate the virtual machine workload as a function of its characteristics. However, virtual machine behavior is an unsteady as these VMs are huge size with non-uniform patterns of both lifetime and users request access. For this reason, modeling VM status is a challenge. Many works in the literature review focus on the VM mitigation to solve load imbalance problem. However, reactive load balancing algorithms cause overhead on the datacenter. Hence, the aim of this paper is how to model the VM status in cloud computing environment. Such status modelling my help in affiliating the load imbalance and resources waste problems. To the best of our knowledge, there is no work formulate the workload status of cloud-based application and/or virtual machines. The rest of this paper is organized as follows. Section 2 reviews the background and related works of workload modelling for virtual machines. Section 3 introduces the proposed workload prediction model. Section 4 discusses the experimental results. Finally, Section 5 concludes this paper.

## **BACKGROUND AND RELATED WORK**

In cloud computing, the resources are virtualized and managed to create virtual machines, which can accomplish jobs that are similar to the jobs executed in a host environment. Virtual machine is a software program or operating system that imitate the behavior of a computer and capable of executing jobs, like running applications and programs similar to a separate computer. Virtual machines (VMs) are becoming more commercial with the advancement of virtualization technology. In virtualization, the hypervisor or virtual machine monitor (VMM) is computer software, firmware or hardware that creates and runs virtual machines (Golden, 2011). Virtualization utilize a hypervisor to proficiently manage several VMs running on a single physical server and to efficiently utilize cloud resources (Barham et al., 2003; Bugnion, Devine, Rosenblum, Sugerman, & Wang, 2012; J. Wang & Fan, 2014; Younge et al., 2011). The process of creating and managing is referred to as resource management.

In cloud computing, resource management shares the resources pooling among multiple users based on the concept of multi-tenancy. Resource management requires complex policies to achieve the optimal usage under diverse objectives as cost reduction, quality enhancement, and power efficiency. In case of improper resource management policies, utilizing the computing resource might be inefficiently. For instance, the service providers are usually sacrifice by doing the over-provisioned to guarantee the high service availability and application quality of service (QoS) (Beloglazov & Buyya, 2013). Such an over-provisioning is not free, but it costs a lot. According to (Uddin, Shah, Alsaqour, Memon, & Saqour Rahasraha, 2013), around 30% of cloud servers, in average, use 10–15% of resource volume most of the time. Besides, virtual machine (VM) migration takes place to transfer VMs across the cloud servers and/or across the datacenters to achieve different purposes, such as server maintenance, power saving, load balancing, and fault tolerance (Kooimey, 2008).

As cloud computing is about hosting huge and interactive applications, which require intensive computing resources, co-hosting multiple VMs cut down application performance because of high resource contention (Åsberg, Forsberg, Nolte, & Kato, 2011; Habib, 2008; Hu, Zhao, Xu, Ding, & Chu, 2013; Nathan, Kulkarni, & Bellur, 2013). Therefore, in order to enhance application performance, a proper mitigation technique should be implemented to send VM to underutilized or resource rich server to cut the level of resource contention (Jeong, Kim, Kim, Lee, & Seo, 2013; Mishra & Jaiswal, 2012; Pop, Anghel, Cioara, Salomie, & Vartic, 2012; Shuja et al., 2012; Silva, Alonso, & Torres, 2009; Yao, Wu, Ren, Zhu, & Li, 2013). Moreover, VM migration techniques migrate VMs within either LAN or WAN boundaries. To optimize power consumption, VM migration technology uses server consolidation frameworks to switch off unnecessary servers (Deshpande, Kulkarni, & Gopalan, 2012).

CloudCom is a one of the platforms, which used for cloud resource management. CloudCom gives very reliable and scalable services to subscribers. Emerging technologies, including Vehicular Adhoc Network (VANET) (Whaiduzzaman, Sookhak, Gani, & Buyya, 2014), Wireless Sensor Networks

(WSN), and mobile computing applications (e.g., online games, bio-medical image processing, etc.). Khan et al. (Khan, Kiah, Khan, & Madani, 2013) used cloud-hosted services (e.g. infrastructure as a service (IaaS), platform as a service (PaaS) (Kremer, 2010; Mell & Grance, 2009). and Software as a service (SaaS) to advance and extend functionalities. An example, Vehicular cloud computing (VCC) merges VANET and CloudCom to assist vehicle drivers to minimize traffic congestion, accidents, and travel time (Huang et al., 2014; Whaiduzzaman et al., 2014). Alike, a sensor cloud merges WSN and CloudCom to develop distant healthcare, vehicular transport systems (VTS), and environmental monitoring (J. Wang & Fan, 2014; Whaiduzzaman et al., 2014) by using cloud services.

Again, such resource management/provisioning schemes are reacting to the current workload by creating virtual machines when needed without considering the available capacity and the expected prices. Moreover, assigning the resources proactively without considering the expected workload of the applications may increase the cost of application hosting due to the unplanned resource consumption. As well, the mitigation process, which happened frequently to react to the load fluctuation, is another issue in cloud computing platforms. For instance, Social networking websites are perhaps the most notable example of highly dynamic and interactive Web 2.0 applications, which gained popularity over the past few years. Their growing attractiveness has spurred demand for a highly scalable and flexible solution for hosting applications. Many larger sites are growing at 100% a year, and smaller sites are expanding at an even more rapid pace, doubling every few months (Subramanyam, Smith, van den Bogaard, & Zhang, 2009). These web applications present additional features that make them different from traditional static workloads but till now they still static and need to be hosted in dynamic cloud (Sobel et al., 2008). For example, their social networking features make each users' actions affect many other users, which makes static load partitioning unsuitable as a scaling strategy. In addition, by means of blogs, photo streams and tagging, users now publish content to one another rather than just consuming static content. Overall, these features show a new sort of workload with particular server/client communication patterns, write patterns and server load. Conversely, most available performance studies use particularly simple static file retrieval tests to evaluate Web servers, often leading to erroneous conclusions (Subramanyam et al., 2009). Authors of (Voorsluys, Broberg, Venugopal, & Buyya, 2009) have implemented a series of experimentations to evaluate the cost of live migration of virtual machines but they did not bring up a procedure describes dynamically when and where to replicate. In a scenario where a modern Internet application is hosted on a set of virtual machines. Live migration experimentations have carried out in scenarios where several levels of load have driven against the application. The results show that, in an instance of a nearly oversubscribed system (serving 600 concurrent users), live migration causes a significant downtime (up to 3 seconds).

Authors of (Rochwerger et al., 2009) proposed provision multiple instances of the same application for different tenants with different customizations but this work did not address the load fluctuations and did not show how to overcome peak hours. Amazon reports several case studies that leverage their EC2 platform, including video processing, genetic simulation and Web applications (Pratt, Howbert, Tasman, & Nilsson, 2011). In particular, such platforms are useful for multi-tier Web applications, which require intensive computing resources. The multi-tiers include web server (e.g. Apache), an application server/dynamic content generation (e.g. PHP, Java EE), and a backend database (e.g. MySQL, Oracle). Amazon EC2 platform adds extra flexibility to scale up/down such web applications, which intensive computing resources. Scaling these applications is achieved by replicating the application instances in multiple virtual machines. The resultant virtual machines are provisioned on demand without predicting the workload of these applications. Thus, provisioning VMs utilizes migration, which is costly and time consuming.

Authors in (Varia, 2011) propose a mechanism for dynamic VM provisioning in IaaS data centers based on clustering. In such an effort, it is essential not only to decide the number of virtualized application instances but also their types. In this research, type of instance is not part of the problem. Hence, use deployed instances, which can all the time, to serve requests. Authors in (Zhu & Agrawal, 2010) proposed a dynamic mechanism for VM provisioning based on control theory considering

user budget. However, such a work considers reconfiguration of available virtual instances (increase or decrease their capacity) and not increasing/decreasing number of instances for a customer, conversely to the proposed approach that applies the latter approach for VM provisioning. (Bi, Zhu, Tian, & Wang, 2010) proposed a model for provisioning multitier applications in Cloud data centers based on queueing networks. However, such a model does not perform recalculation of number of required VMs based on expected load and monitored performance as does our approach. (Chieu, Mohindra, Karve, & Segal, 2009) proposed a reactive algorithm for dynamic VM provisioning of PaaS and SaaS applications, whereas our approach is proactive in the sense that number of instances is changed based on the expected arrival rate of requests. (Lee, Wang, Zomaya, & Zhou, 2010) proposed a queueing network to model SaaS mashup applications. The goal is to maximize profit or reduce costs of the SaaS provider by finding an optimal number of instances for the application, but they did not estimate the upcoming load. (Rodero-Merino et al., 2010) proposed a system called Claudia, where provisioning is based on performance indicators and elasticity rules defined by users. In both approaches number of instances vary reactively to incoming request rate, whereas the proposed work proactively applies adaptive provisioning to the expected load. (Jung, Hiltunen, Joshi, Schlichting, & Pu, 2010) proposed the Mistral system, which performs management at the host level in datacenter to manage power consumption of resources and performance of applications. However, this approach requires access to the physical infrastructure, which typical IaaS providers do not provide to consumers. Therefore, Mistral is suitable in places where the same provider offers both the infrastructure and the application, while the proposed approach can both applied in the same case or in cases where IaaS and PaaS/SaaS providers are different organizations. Calheiros et al. (2011) introduced automatic workload adaptation model that automatically fit to the change in application workload. Alheiros et al. modeled the application behavior based on the analytical indicators and QoS criteria of application performance. The simulation-based experimental results proves the effect of arrival patterns and resource demands on the workload change of the application. Froushha and Reza (2018) proposed a workload modelling for cloud-based application for both stream and batch processing data-intensive systems. Their proposed model was based on statistical measures about the application performance during the run-time.

## VIRTUAL MACHINE STATUS MODELLING

As mentioned above, the workload of any virtual machine can be estimated by considering the virtual machine characteristics. For example, the virtual machine workload means the number of active requests targeting that virtual machine at the same time during the peak busy period. In the case of estimating the workload for individual virtual machine, there is a population of cloud users  $H$  issuing different requests targeting diverse virtual machines at the peak busy period. Each cloud user can issue a number of normal requests  $\lambda$  and interactive request  $\lambda_{ucr}$  with holding times  $t_i$  and  $t_{ucr}$  for normal and interactive requests, respectively. For instance, the cloud user can request a web page in web server, in addition to, playing an interactive game online in an interactive online gaming platform like Pokémon Go. Thus, the workload for a cloud-based application  $i$  can be estimated as the portion of concurrent normal and interactive requests targeting the cloud datacenter by the cloud users to access/interact with that cloud-based application during the peak busy period. This can be interpreted mathematically as in Equation 1:

$$L_i = \frac{p_i * H}{T_{peak}} \left( (\lambda * t_i) + (\lambda_{ucr} * t_{ucr}) \right) \quad (1)$$

where the terms  $L_i$ ,  $p_i$  and  $t_i$  denote the expected workload, the popularity and the running time of the particular cloud-based application  $i$ , respectively. The popularity takes a value between 0 and 1, which follow the applications popularity scheme on the internet (Hameed et al., 2016). The term  $H$  denotes the number of cloud users, and  $T_{peak}$  denotes the peak busy period of that application in minutes. In addition, the terms  $\lambda$  and  $\lambda_{vcr}$  denote the request arrival rate and VCR commands request arrival rate, respectively. Both arrival rates follow Poisson distribution. In Equation (1), the term  $H \left( (\lambda * t_i) + (\lambda_{vcr} * t_{vcr}) \right)$  represents the total sum of all requests issued by the cloud users within the peak busy period. By multiplying this term by the cloud-based application popularity value  $p_i$ , the number of requests targeting this cloud-based application is obtained. Finally, the concurrent requests per a unit time are estimated by dividing on the term  $T_{peak}$ .

After cloud-based application workload estimation, the replication degree (i.e. the number of virtual machines that are required to handle the expected load of cloud-based application) is calculated. The replication degree must be controlled by the cloud-based application popularity. For instance, cloud-based application  $i$  can be allocated on all physical machines if its popularity is extremely high. This is to ensure that the expected load can be distributed on as a large number of physical machines as possible to minimize the request rejection rate. On the other hand, there is no need to replicate the remarkably low popular cloud-based applications so that one copy is enough to catch its low expected load. Based on the above explanation, the replication degree (i.e. number of copies) for a cloud-based application  $i$  can be formulated as a function of the number of physical machines and the normalized popularity as shown in Equation (2). According to (Nafaa, Murphy, & Murphy, 2008), normalizing the popularity distribution improve the overall performance by building a strong relationship between the content popularity and the replication degree:

$$r_i = \left\lceil S_a * p_i^n \right\rceil \quad (2)$$

where the term  $r_i$  denotes the number of replicas for cloud-based application  $i$ ,  $S_a$  represents the number of physical machines in cloud datacenter  $a$ , and finally the term  $p_i^n$  represented the normalized value of  $p_i$  that should be within [0,1]. The normalized popularity value has been obtained using Min-Max Normalization Law. Min-max normalization performs a linear transformation on the original data (Fati et al., n.d.). Min-max normalization maps  $p_i$  to  $p_i^n$  value in the range  $[\text{new\_min}, \text{new\_max}]$ . In this case, the  $\text{new\_max}$  value equals one where the highest popularity value in the dataset will be scaled to equal the value (1) and the other values will be scaled successively. The operator  $\lceil \cdot \rceil$  is a ceiling function operator to take the largest integer nearest to the calculated term.

After computing the expected number of virtual machines for a cloud-based application  $i$ , the expected load for each virtual machine can be calculated by dividing the expected load for that video on its number of replicas as follows:

$$L_{ri} = L_i / r_i \quad (3)$$

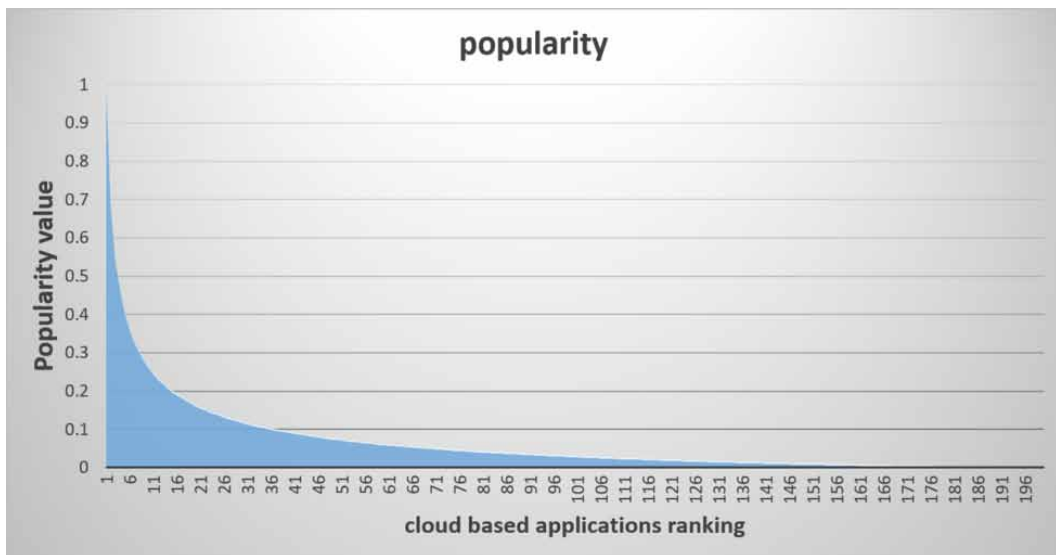
where  $L_{ri}$  represents the load of one virtual machine for cloud-based application  $i$ ,  $L_i$  denotes the load of cloud-based application  $i$ , and  $r_i$  represents the replication degree for cloud-based application  $i$ , which obtained from Equation (2).

According to the proposed virtual machine status model, the content of high popularity value, which has a highly expected load, is replicated more to handle the increasing demands. On the other hand, the contents of low popularity value can be replicated as less as possible for serving the low expected load. Furthermore, allocating the virtual machines and distributing the incoming requests according to this proposed virtual machine status model can be useful in maintaining the load as balanced as possible within the datacenter.

## EXPERIMENTAL RESULTS

To investigate the performance of our proposed virtual machine status model, we have tested the model on an empirical data set that is sampled according to Zipf's like popularity distribution, which widely used in web world. There are a set of assumptions taken into account during the experimental study of the proposed model as following: the population of subscribers  $H=1000$ , each subscriber can issue a number of normal requests  $\lambda=2$  request/user/minutes and interactive request  $\lambda_{vcr}=2$  request/user/minutes with holding times  $t_{vcr}=10$  seconds for interactive requests. The popularity distribution of cloud-based application is being affected with users' preference skewness according to the users' habits and/or behaviors, as depicted in Figure 1. In this figure, we can notice that only some cloud-based applications are popular during the running time.

Figure 1. Popularity distribution for cloud-based applications based on users' preferences skewness



The applications popularity is computed according to Zipf's Law (Adamic & Huberman, 2002) as in Equation 4 where  $i$ ,  $N$ , and  $\theta$  represent the content rank, the total number of contents, and the skewness degree respectively. The application rank refers to the position of application in the sorted list according to the popularity values. In this list, the first video is the highest popular video, and so on:

$$P(i, \theta, N) = \frac{1/i^\theta}{\sum_{n=1}^N (1/n^\theta)} \quad (4)$$

These popular cloud-based applications will be accessed frequently and will utilize more resources than the other applications. Figure 2 depicts the resources consumption distribution for the cloud-based applications.

Then, the workload is computed using our proposed virtual machine status model. Note that the expected load of server  $j$  can be expressed by summing the load of all applications stored in this server. Figure 3 depicts the expected load for the cloud-based applications, which is obtained from the proposed virtual machine status model.

Figure 2. Resource consumption for cloud-based applications

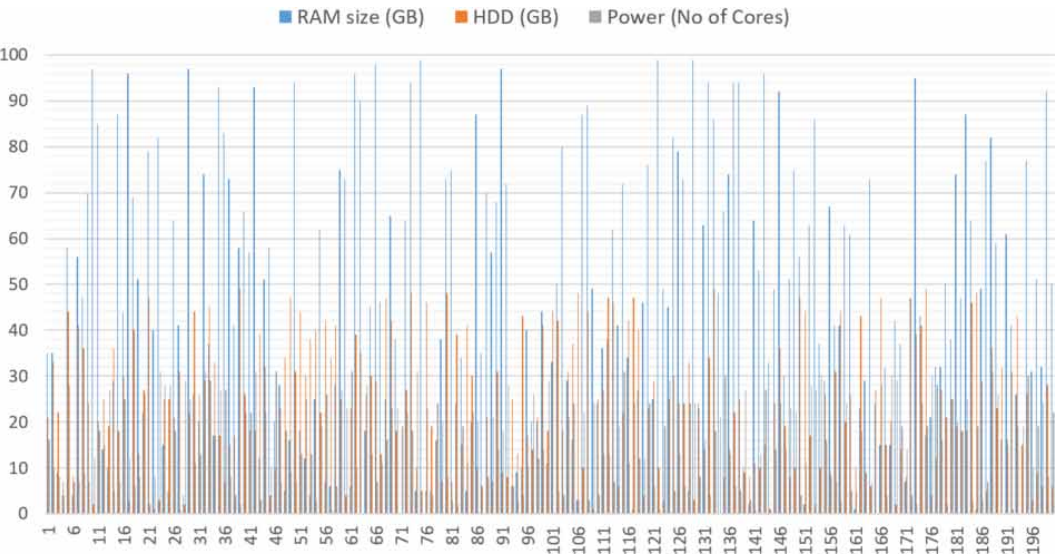


Figure 3. The distribution of expected content load

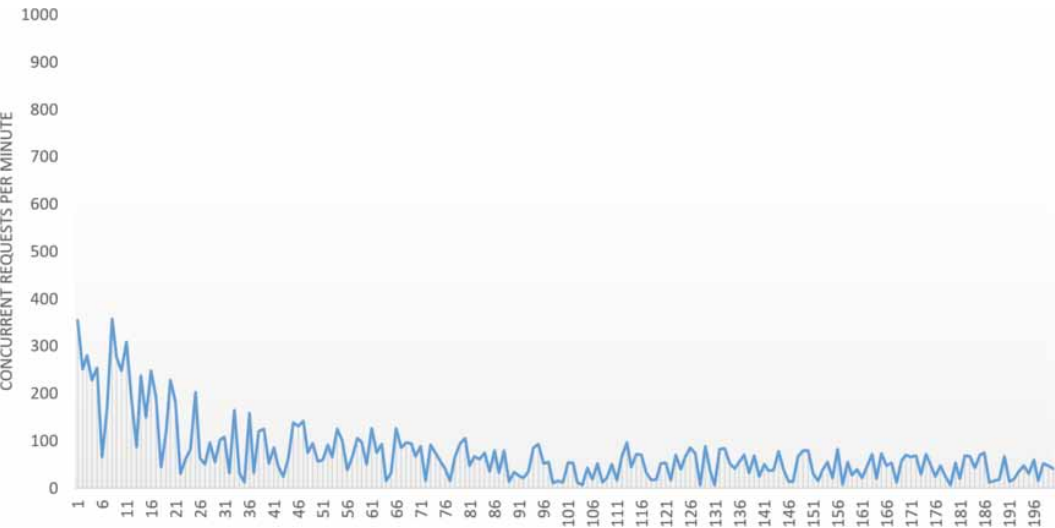
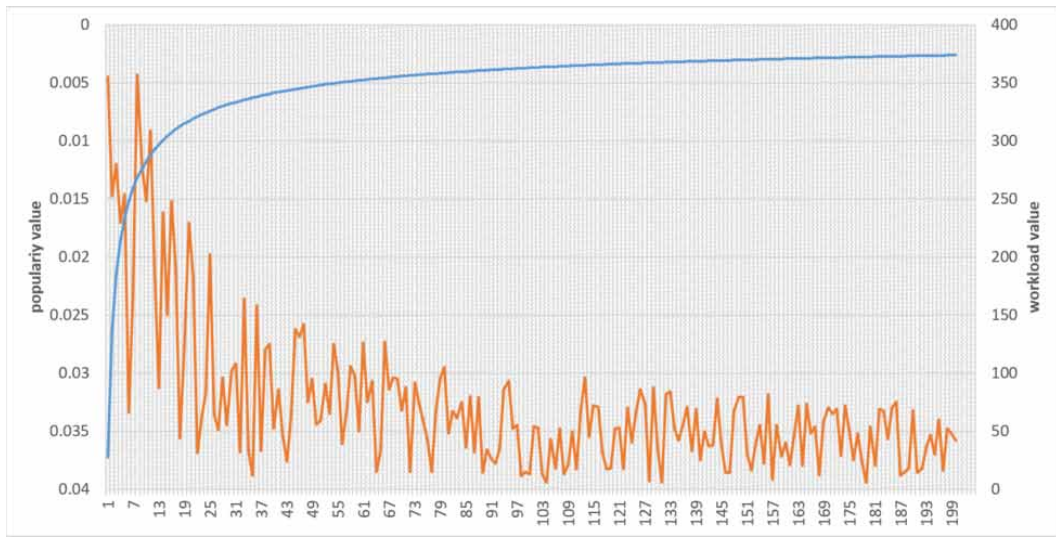


Figure 4 plots the relationship between the expected workload of cloud-based applications and the popularity distribution. What is apparent is the fact that the expected load of applications follows the popularity distribution, but with a slight difference. This difference comes out the other factors that contribute in the workload estimation like the running time and interactivity.

Figure 4. The relation between expected load and popularity



Another experiment is conducted on a datacenter with 100 blade servers whereby each application will be replicated among a certain number of virtual machines according to two factors: application popularity and expected application workload. The rationale behind this experiment is to estimate how many virtual machines needed for each application, and the workload distribution among these virtual machines. In this case, the workload of each application will be distributed among the virtual machines to ensure serving all the incoming requests without delay or interruption. Figure 5 depicts the number of virtual machines and estimated average load distribution. In this figure, the number of virtual machines for each application is following the popularity distribution. This means that the popular application with high demand will be replicated in more virtual machines than those applications having less demand or not requested frequently. In the figure, we can notice that there are some applications with high workload, but with less popularity. These kinds of applications will be replicated in a less number of virtual machines, and the average workload for the resultant virtual machines will be a bit high, as shown in the applications between the numbers 20 and 40 in the x-axis.

After that, we deployed a request distribution algorithm, which proposed by (S. M. A. Gaber & Sumari, 2014) wherein the incoming requests will be assigned to the virtual machines based on the expected and current load for both virtual machines and the physical machines hosting that virtual machines. This algorithm, which named CARDA, has been tailored to run on the top of our proposed virtual machine status model. The findings from CARDA has been compared with the widely used request distribution algorithm, called Round Robin (RR) algorithm. The experiment showed that tailored CARDA algorithm outperforms RR algorithm in terms of physical machines workload balance, as well as, the amount of served requests. Figure 6 depicts the distribution of workload among the physical machines. Moreover, it is interesting to say that the popularity distribution has a significant effect on expecting the workload for applications and virtual machines, which helps in planning the

Figure 5. The number of virtual machine and the average workload

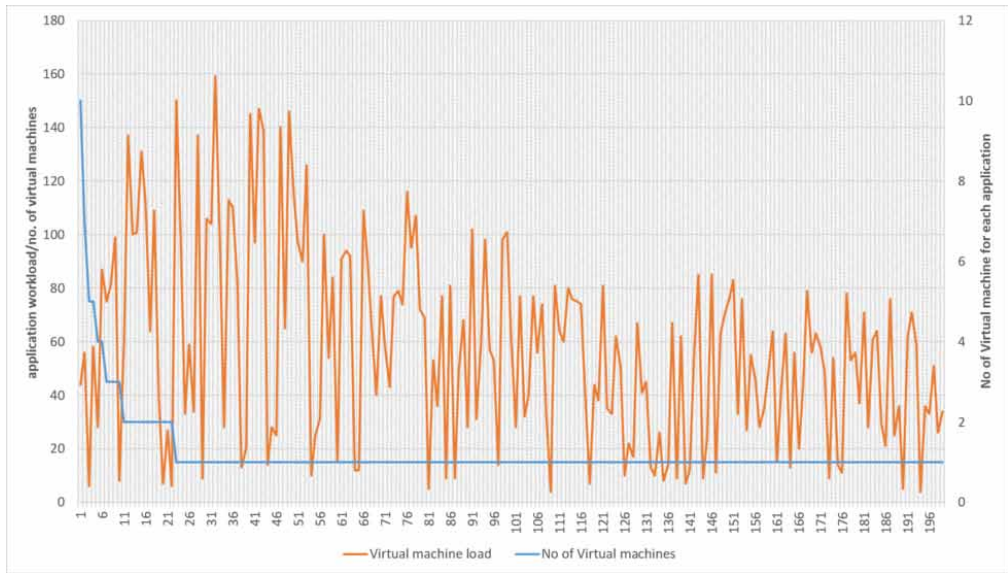
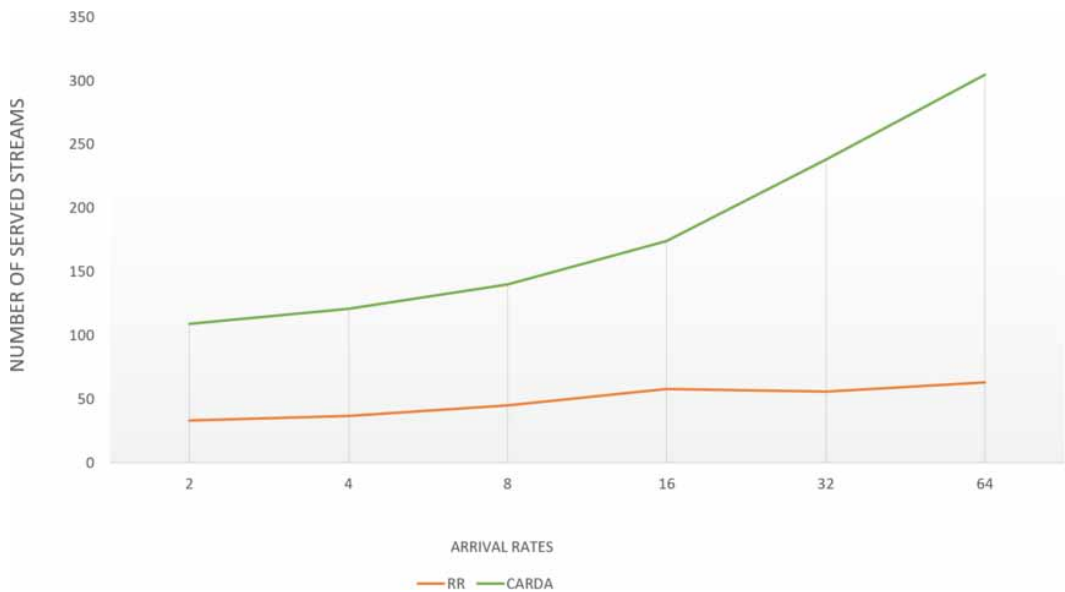


Figure 6. The distribution of workload among the physical machines



virtual machines allocation; however, it has no effect on the incoming requests distribution. The influential factor in request distribution is the request arrival rate.

## CONCLUSION

In cloud computing platforms, huge applications with diverse characteristics are hosted and replicated among multiple virtual machines to absorb the incoming workload. Many resource-provisioning

techniques are proposed to deal with load fluctuation proactively. The main aim of these techniques is to improve the user satisfaction and enhance the quality of service. However, such resource provisioning techniques suffer from the intensive overhead of mitigation and sometime load imbalance. To the best of authors' knowledge, there is no estimation/prediction for the cloud-based application according to its characteristics. Thus, this paper introduces the concept of workload estimation as a mathematical formula that consider the different characteristics as interactivity, popularity, request arrival rate (e.g. normal requests and interactive requests), and size. Estimating the workload of each application. The proposed model estimates the maximum workload for each application. The load is estimated as the number of active concurrent requests that access the application at the peak busy period based on application popularity. The experimental results showed promising performance for proposed model against the traditional models used currently.

## REFERENCES

- Adamic, L. A., & Huberman, B. A. (2002). Zipf's law and the Internet. *Glottometrics*, 3(1), 143–150.
- AlJahdali, H., Albatli, A., Garraghan, P., Townend, P., Lau, L., & Xu, J. (2014). *Multi-tenancy in cloud computing*. Paper presented at the Service Oriented System Engineering (SOSE), 2014 IEEE 8th International Symposium on. doi:10.1109/SOSE.2014.50
- Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., & Stoica, I. et al. (2010). A view of cloud computing. *Communications of the ACM*, 53(4), 50–58. doi:10.1145/1721654.1721672
- Åsberg, M., Forsberg, N., Nolte, T., & Kato, S. (2011). *Towards real-time scheduling of virtual machines without kernel modifications*. Paper presented at the Emerging Technologies & Factory Automation (ETFA), 2011 IEEE 16th Conference on. doi:10.1109/ETFA.2011.6059185
- Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., . . . Warfield, A. (2003). *Xen and the art of virtualization*. Paper presented at the ACM SIGOPS operating systems review. doi:10.1145/945445.945462
- Beloglazov, A., & Buyya, R. (2013). Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints. *IEEE Transactions on Parallel and Distributed Systems*, 24(7), 1366–1379. doi:10.1109/TPDS.2012.240
- Bi, J., Zhu, Z., Tian, R., & Wang, Q. (2010). *Dynamic provisioning modeling for virtualized multi-tier applications in cloud data center*. Paper presented at the Cloud Computing (CLOUD), 2010 IEEE 3rd international conference on. doi:10.1109/CLOUD.2010.53
- Bugnion, E., Devine, S., Rosenblum, M., Sugerman, J., & Wang, E. Y. (2012). Bringing virtualization to the x86 architecture with the original vmware workstation. *ACM Transactions on Computer Systems*, 30(4), 12. doi:10.1145/2382553.2382554
- Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, 25(6), 599–616. doi:10.1016/j.future.2008.12.001
- Calheiros, R. N., Ranjan, R., & Buyya, R. (2011, September). Virtual machine provisioning based on analytical performance and QoS in cloud computing environments. In *2011 International Conference on Parallel Processing* (pp. 295–304). IEEE. doi:10.1109/ICPP.2011.17
- Chieu, T. C., Mohindra, A., Karve, A. A., & Segal, A. (2009). *Dynamic scaling of web applications in a virtualized cloud computing environment*. Paper presented at the E-Business Engineering, 2009. ICEBE'09. IEEE International Conference on. doi:10.1109/ICEBE.2009.45
- Deshpande, U., Kulkarni, U., & Gopalan, K. (2012). Inter-rack live migration of multiple virtual machines. *Proceedings of the 6th international workshop on Virtualization Technologies in Distributed Computing Date*. doi:10.1145/2287056.2287062
- Fati, S. M., & Sumari, P. (2018). Content Awareness in IPTV Delivery Networks. *IPTV Delivery Networks: Next Generation Architectures for Live and Video-on-Demand Services*, 93.
- Fati, S. M., Sumari, P., Yuhaniz, S. S., & Sjarif, N. N. B. A. (n.d.). *Modelling contents status for IPTV delivery networks*. Academic Press.
- Foroushha, K., & Reza, A. (2018). *Workload Modelling and Elasticity Management of Data-Intensive Systems*. Academic Press.
- Gaber, A., Mohamed, S., Sumari, P., & Budiarto, R. (2012). Balanced content allocation scheme for peerservice area CDN architecture for IPTV SERVICES. *Journal of Information & Communication Technology*, 11.
- Gaber, S. M. A., & Sumari, P. (2014). Predictive and content-aware load balancing algorithm for peer-service area based IPTV networks. *Multimedia Tools and Applications*, 70(3), 1987–2010. doi:10.1007/s11042-012-1209-7
- García, R., Pañeda, X. G., Melendi, D., & García, V. (2009). Probabilistic analysis and interdependence discovery in the user interactions of a video news on demand service. *Computer Networks*, 53(12), 2038–2049. doi:10.1016/j.comnet.2009.03.011

- Golden, B. (2011). *Virtualization for dummies*. John Wiley & Sons.
- Habib, I. (2008). Virtualization with kvm. *Linux Journal*, 2008(166), 8.
- Hameed, A., Khoshkbarforoushha, A., Ranjan, R., Jayaraman, P. P., Kolodziej, J., Balaji, P., & Vishnu, A. et al. (2016). A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems. *Computing*, 98(7), 751–774. doi:10.1007/s00607-014-0407-8
- Hu, L., Zhao, J., Xu, G., Ding, Y., & Chu, J. (2013). HMDC: Live virtual machine migration based on hybrid memory copy and delta compression. *Applications of Mathematics*, 7(2L), 639–646.
- Huang, J., Du, D., Duan, Q., Zhang, Y., Zhao, Y., Luo, H., Mai, Z., & Liu, Q. (2014). Modeling and analysis on congestion control for data transmission in sensor clouds. *International Journal of Distributed Sensor Networks*, 10(3), 453983. doi:10.1155/2014/453983
- Jeong, J., Kim, S.-H., Kim, H., Lee, J., & Seo, E. (2013). Analysis of virtual machine live-migration as a method for power-capping. *The Journal of Supercomputing*, 66(3), 1629–1655. doi:10.1007/s11227-013-0956-1
- Joe, I., Yi, J. H., & Sohn, K.-S. (2011). A content-based caching algorithm for streaming media cache servers in cdn. In *Multimedia, Computer Graphics and Broadcasting* (pp. 28-36). Springer. doi:10.1007/978-3-642-27204-2\_4
- Jung, G., Hiltunen, M. A., Joshi, K. R., Schlichting, R. D., & Pu, C. (2010). *Mistral: Dynamically managing power, performance, and adaptation cost in cloud infrastructures*. Paper presented at the 2010 International Conference on Distributed Computing Systems. doi:10.1109/ICDCS.2010.88
- Khan, A. N., Kiah, M. M., Khan, S. U., & Madani, S. A. (2013). Towards secure mobile cloud computing: A survey. *Future Generation Computer Systems*, 29(5), 1278–1299. doi:10.1016/j.future.2012.08.003
- Koomey, J. G. (2008). Worldwide electricity used in data centers. *Environmental Research Letters*, 3(3), 034008. doi:10.1088/1748-9326/3/3/034008
- Kremer, J. (2010). *Cloud Computing and Virtualization*. White paper on virtualization.
- Lee, Y. C., Wang, C., Zomaya, A. Y., & Zhou, B. B. (2010). *Profit-driven service request scheduling in clouds*. Paper presented at the Cluster, Cloud and Grid Computing (CCGrid), 2010 10th IEEE/ACM International Conference on. doi:10.1109/CCGRID.2010.83
- Li, M., & Wu, C.-H. (2010). A cost-effective resource allocation and management scheme for content networks supporting IPTV services. *Computer Communications*, 33(1), 83–91. doi:10.1016/j.comcom.2009.08.003
- Mell, P., & Grance, T. (2009). The NIST definition of cloud computing. *National Institute of Standards and Technology*, 53(6), 50.
- Mishra, R., & Jaiswal, A. (2012). Ant colony optimization: A solution of load balancing in cloud. *International Journal of Web & Semantic Technology*, 3(2), 33–50. doi:10.5121/ijwest.2012.3203
- Nafaa, A., Murphy, S., & Murphy, L. (2008). Analysis of a large-scale VOD architecture for broadband operators: A P2P-based solution. *IEEE Communications Magazine*, 46(12), 47–55. doi:10.1109/MCOM.2008.4689207
- Nathan, S., Kulkarni, P., & Bellur, U. (2013). Resource availability based performance benchmarking of virtual machine migrations. *Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering*. doi:10.1145/2479871.2479932
- Pandya, P. P., & Bheda, H. A. (2014). Dynamic resource allocation techniques in cloud computing. *International Journal of Advance Research in Computer Science and Management Studies*, 2(1).
- Patel, P., Ranabahu, A. H., & Sheth, A. P. (2009). *Service level agreement in cloud computing*. Academic Press.
- Pop, C. B., Anghel, I., Cioara, T., Salomie, I., & Vartic, I. (2012). A swarm-inspired data center consolidation methodology. *Proceedings of the 2nd International Conference on Web Intelligence, Mining and Semantics*.
- Pratt, B., Howbert, J. J., Tasman, N. I., & Nilsson, E. J. (2011). MR-tandem: Parallel X! tandem using hadoop MapReduce on amazon Web services. *Bioinformatics (Oxford, England)*, 28(1), 136–137. doi:10.1093/bioinformatics/btr615 PMID:22072385

- Rimal, B. P., Choi, E., & Lumb, I. (2009). *A taxonomy and survey of cloud computing systems*. Paper presented at the INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference on. doi:10.1109/NCM.2009.218
- Rochwerger, B., Breitgand, D., Levy, E., Galis, A., Nagin, K., Llorente, I. M., . . . Caceres, J. (2009). The reservoir model and architecture for open federated cloud computing. *IBM Journal of Research and Development*, 53(4).
- Rodero-Merino, L., Vaquero, L. M., Gil, V., Galán, F., Fontán, J., Montero, R. S., & Llorente, I. M. (2010). From infrastructure delivery to service management in clouds. *Future Generation Computer Systems*, 26(8), 1226–1240. doi:10.1016/j.future.2010.02.013
- Shuja, J., Madani, S. A., Bilal, K., Hayat, K., Khan, S. U., & Sarwar, S. (2012). Energy-efficient data centers. *Computing*, 94(12), 973–994. doi:10.1007/s00607-012-0211-2
- Silva, L. M., Alonso, J., & Torres, J. (2009). Using virtualization to improve software rejuvenation. *IEEE Transactions on Computers*, 58(11), 1525–1538. doi:10.1109/TC.2009.119
- Sobel, W., Subramanyam, S., Sucharitakul, A., Nguyen, J., Wong, H., Klepchukov, A., & Patterson, D. et al. (2008). Cloudstone: Multi-platform, multi-language benchmark and measurement tools for web 2.0. *Proc. of CCA*.
- Subramanyam, S., Smith, R., van den Bogaard, P., & Zhang, A. (2009). Deploying web 2.0 applications on sun servers and the opensolaris operating system. *Sun BluePrints*, 820-7729.
- Uddin, M., Shah, A., Alsaqour, R., & Memon, J., & Saqour Rahasraha, M. J. (2013). Measuring efficiency of tier level data centers to implement green energy efficient data centers. *Middle East Journal of Scientific Research*, 15(2), 200–207.
- Varia, J. (2011). Best practices in architecting cloud applications in the AWS cloud. *Cloud Computing: Principles and Paradigms*, 457-490.
- Vicari, C. (2008). *Distributed Dynamic Replica Placement and Request Redirection in Content Delivery Networks*. Academic Press.
- Voorsluys, W., Broberg, J., Venugopal, S., & Buyya, R. (2009). *Cost of virtual machine live migration in clouds: A performance evaluation*. Paper presented at the IEEE International Conference on Cloud Computing. doi:10.1007/978-3-642-10665-1\_23
- Wang, J., & Fan, Z. H. (2014). *Family health telemonitoring system based on WSN*. Paper presented at the Advanced Materials Research.
- Wang, L., Von Laszewski, G., Younge, A., He, X., Kunze, M., Tao, J., & Fu, C. (2010). Cloud computing: A perspective study. *New Generation Computing*, 28(2), 137–146. doi:10.1007/s00354-008-0081-5
- Whaiduzzaman, M., Sookhak, M., Gani, A., & Buyya, R. (2014). A survey on vehicular cloud computing. *Journal of Network and Computer Applications*, 40, 325–344. doi:10.1016/j.jnca.2013.08.004
- Yao, L., Wu, G., Ren, J., Zhu, Y., & Li, Y. (2013). Guaranteeing fault-tolerant requirement load balancing scheme based on VM migration. *The Computer Journal*, 57(2), 225–232. doi:10.1093/comjnl/bxt012
- Younge, A. J., Henschel, R., Brown, J. T., Von Laszewski, G., Qiu, J., & Fox, G. C. (2011). *Analysis of virtualization technologies for high performance computing environments*. Paper presented at the Cloud Computing (CLOUD), 2011 IEEE International Conference on. doi:10.1109/CLOUD.2011.29
- Zhu, Q., & Agrawal, G. (2010). Resource provisioning with budget constraints for adaptive applications in cloud environments. *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*. doi:10.1145/1851476.1851516

*Suliman Mohamed Fati obtained his BSc (2002), MSc (2009), and PhD (2014) from Ain Shams University, Egypt; Cairo University, Egypt; and Universiti Sains Malaysia (USM), Malaysia, respectively. He is an assistant Professor in College of Computers and Information Sciences, Prince Sultan University, Saudi Arabia. His research interests focus on Internet of Things, Machine Learning, Social Media Mining, Cloud Computing, Cloud Computing Security, and Information Security. He has authored over 20 ISI and Scopus journal/conference papers, books, and book chapters. He is the lead editor for an edited book shortly be published by Wiley entitled A Comprehensive Guide to IPTV Delivery Networks. In addition, he is a member of different professional bodies such as IEEE, IACSIT, IAENG, and Institute of Research Engineers and Doctors, USA. He is a reviewer in many international impact-factor journals, and also technical committee program member in a number of international conferences.*

*Ayman Kamel Jaradat has obtained his PhD from Universiti Teknologi PETRONAS in 2013, MSc from University Sains Malaysia in 2007, and BSc from Yarmouk University Jordan in 1989. Jaradat is specialized in Computer Science and his research interest includes high-performance computing, grid computing, cloud computing, genetic algorithm, distributed algorithms and applications. Jaradat was the Dean of Faculty of Computer and Information Technology at Al-Madinah International University and currently is assistant professor at Al Majmaah University.*

*Ibrahim Abunadi is an Assistant Professor in the College of Computer and Information Sciences and fellow of the British Higher Education Academy. He received his Ph.D. in Information Systems from the School Information Communication Technology at Griffith University in Australia. Dr. Abunadi taught many courses including Human Computer Interaction, Business Process Management, Enterprise Architecture, Technology Innovations, Business Analysis, Computer Databases, and Computer Applications for Business. His research focuses on data mining, technology adoption, e-government, and human-computer interaction. He has numerous publications in journals such as IEEE Access, Journal of Universal Computer Science, Information Technology and Libraries, Journal of Organizational and End User Computing, and others.*

*Ahmed Sameh Mohammed is Professor of Computer Science and Information Systems at Prince Sultan University since 2009. He is also a Professor of Computer Science and Engineering at the American University in Cairo since 1991. He is currently the Chair of Information Systems at PSU since 2010. His current research interests include Neural Networks, Artificial Intelligence, Parallel Computing, Mobile Computing, and Hardware Design. He directs the post-graduate program at the Computer Science and Information System department at PSU. He has supervised 30 Ph.D. and Master theses at both The American University in Cairo and Cairo University. His publication record includes 12 book chapters, 45 journal papers, and 122 conference papers. He was a faculty member in the Department of Computer Science at Kuwait University, George Washington University, and Lewis & Clark College, Portland, Oregon. He is a member of ACM, IEEE, ICS, CIPS, and ISCA. He received the Google Research Award in 2005 for his life-long contribution to the field of computing through his 30 years in the field, and 180 publications of book chapters, journals, and refereed conferences. Born in Alexandria, Egypt, Dr. Ahmed Sameh earned his B.Sc. in Computer Science from Alexandria University, Egypt in 1979. He earned his M.Sc. and Ph.D. in Computer Science from University of Alberta, Canada in 1985 and 1989, respectively. Dr. Ahmed Sameh held visiting research positions at George Washington University, University of Iowa, and Queens University.*