*I.J. Information Technology and Computer Science,* **2019, 5, 11-22**
Published Online May 2019 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2019.05.02

# Autonomous Virtual Machine Sizing and Resource Usage Prediction for Efficient Resource Utilization in Multi-Tenant Public Cloud

**Derdus M. Kenga and Vincent O. Omwenga**
Faculty of Information Technology, Strathmore University, Kenya
E-mail: {derduskenga, vincentoteke}@gmail.com

**Patrick J. Ogao**
Faculty of Engineering Science and Technology, Technical University of Kenya, Kenya
E-mail: ogaopj@gmail.com

Received: 16 January 2019; Accepted: 17 February 2019; Published: 08 May 2019

***Abstract*—**In recent years, the use of cloud computing has increased exponentially to satisfy computing needs in both big and small organizations. However, the high amounts of power consumed by cloud data centres have raised concern. A major cause of power wastage in cloud computing is inefficient utilization of computing resources. In Infrastructure as a Service, the inefficiency is caused when users request for more resources for virtual machines than is required. In this paper, we propose a technique for automatic virtual machine sizing and resource usage prediction using neural networks, for multi-tenant Infrastructure as a Service cloud service model. The proposed technique aims at reducing energy wastage in data centres by efficient resource utilization. An evaluation of our technique on CloudSim Plus cloud simulator and WEKA shows that effective VM sizing not only achieves energy savings but also reduces the cost of using cloud services from a customer perspective.

***Index Terms*—**Cloud computing, virtual machine sizing, IaaS cloud, multi-tenant public cloud, energy efficiency, CloudSim plus, neural networks.

## I. Introduction

The growth of appetite to cloud computing by small and big enterprises has resulted in setting up of many data centres. Unfortunately, data centres consume a lot of energy and this a concern. According to [1] and [2], power bills have been the largest commodity service expenditure in CSPs. Moreover, data centre electricity usage was about 3% by 2012 and now it is expected to triple by 2020 [1]. In addition, high power usage has a negative environmental implication, which is the release of $CO_2$ to the environment. In 2013 alone, the US data centres consumed 91 billion kWh of electricity [3]. This amount of electricity is enough to power New York City's households for a full 2 years and is expected to rise to 140 billion kWh by 2020, which can in turn release 150 million tons of $CO_2$.

Some of the energy consumed in data centres does not perform useful processing and thus is wasted [4]. The major causes of data centre power wastage are low server utilization and idle power wastage, which is caused by inefficient resource utilization [4,5]. An example of inefficient resource utilization is where excessive resources are provisioned than is required. This means that many PMs are used to run workloads, which would actually be executed by less PMs. Idle power wastage is caused by non-proportional computing, where energy consumed by data centre servers is high even at low server load. A survey involving 5000 servers revealed that although servers are generally not idle, their utilization never reaches 100% [6]. An analysis by [7] on Google Cluster Trace (GCT), 65 % of CPU and 45 % of memory go to waste.

Virtualization technology is poised to address the problem of resource wastage in cloud computing [8]. Virtualization provides a way of independently hosting applications that share resources in the same PM and thus improves the energy efficiency of data centres through consolidation. Consolidation allows packing many VMs in the one PM so that other PMs can be shut down thus achieving energy savings. However, this technique may not useful in some circumstances. For instance, in Infrastructure as a service (IaaS), which is the most promising cloud model among small organizations [9], [10], customers are allowed to pick VM sizes from CSPs' list of available VM types without the knowledge of the actual amount of resources their applications need [11]. A VM size or type is the amount of each computing resources assigned to a VM such as memory, CPU, hard disk and network bandwidth [12]. More often, the resources are over-provisioned and thus goes to waste. From this viewpoint, resources not performing useful work consume energy and the customer has to pay for them.

In order to determine the actual amount of resources used by a VM, data about a particular VM has to be collected and analyzed and this has to be done by the CSP. From a CSPs' point of view, applications running in a particular customer VM are a black box host in a VM [13]. Fortunately, the CSP has access to the virtualization layer, where they can monitor resource usage for each hosted VM. From this viewpoint, there are many attempts that have made by the large organization to address the problem of VM sizing, which include ParkMyCloud [14] for Windows Azure Cloud service customers, VM sizing recommendation service for Google customers [15] and Amazon's CloudWatch [16]. All the methods provided by Azure, Google and AWS cloud services have to be manually completed by customers and seems to fit customers who already have knowledge in cloud computing.

To address the problem of non-proportional computing, Dynamic Voltage and Frequency Scaling (DVFS) has been used. DVFS is an energy saving technique in computer architecture that is used to save energy when the server load is low [4]. In this technique, the frequency and voltage of the CPU are scaled dynamically to relate with the amount of server load. The power, *P*, of the CPU is computed as shown in equation (1) [17] where *V* is the voltage, *F* is the frequency and *C* is the capacitive load on the system. It is observed that if voltage and frequency are lowered, there will be a significant reduction in power consumption. However, DVFS is hardware-based technique and works well only on CPU bound tasks because dynamic power ranges for other components (memory, disk and network) are much narrower [6].

$$P = V^2 * \mathrm{F} * \mathrm{C}, \tag{1}$$

Another technique for managing dynamic resource management is dynamic memory allocation known as *Memory Ballooning* [18, 19, 20, 21]. This technique allows the hypervisor to reclaim unused memory from one VM to share it with another.

VM trace logs collected from VMs can be analyzed or characterized using statistical techniques such as mean, quartiles and correlations [22]. This analysis can be used as insights to address the problems of VM sizing for efficient resource utilization. Moreover, other considerations such as industry standards can affect how VM sizing is achieved. For instance, the Data Center Maturity Model (DCMM) is a best practice reference model used for evaluating data centres. According to DCMM, the highest level, otherwise known as *Visionary*, is achieved if the average monthly CPU utilization is above 60% [23]. Another reference model is a threshold setting for physical CPU and memory known as VMware Knowledge Base (VMware KB) [21]. According to VMware KB, 80% CPU utilization is considered a ceiling and a warning if CPU utilization is 90% for 5 minutes. Thresholds set by such reference models can be used for scaling [24]. All these techniques can be completed by using automatic resource allocation, resource provisioning and resource monitoring by designing Autonomous Resource Management System (ARMS) [25].

In this paper, we propose an architecture for VM sizing in IaaS multi-tenant public cloud based on historical resource usage. Because of the dynamic nature of cloud workload, static resources assigned to VM may not be accurate. Thus the fixed thresholds resource values can trigger unnecessary migrations, which will, in turn, increase energy consumption [26]. Therefore, we extend our architecture to predict resource usage so that, at VM peak resource usage, resources from host reserves can be provisioned.

In order to apply our technique using real workload traces, we have utilized GWA-T-13 Materna dataset, which contains information about VMs hosted in a data centre that supports business-critical workloads in Germany. The dataset structure and format is explained in [27].

## II. Related Work

Efficient management of power usage in data centres through prediction and VM sizing is on the rise [11,28,29]. However, the approaches used have not attempted to combine VM sizing and resource usage prediction. Moreover, there are gaps that can be exploited when it comes to multi-tenant IaaS cloud models.

The work in [30] in one of the earliest attempts to introduce VM sizing. The authors' attempts to give a VM its size (resource demand) based on its contribution to the aggregate resource of the host server. An overall outcome is a function of the VMs own resource demand and that of its co-located VMs along the time. Based on this effective size, a VM placement algorithm is developed, which reduces chances of VM migration.

In [11] the authors have described an architecture for customizing VMs to match workload task characteristics in container-based virtualization. Their approach clusters job tasks based on resource usage and other characteristics such as task length, priority and submission rate, which are them mapped to appropriate VM types (VM sizes). They have used GCT to evaluate their technique, which they conclude achieves their goal - to reduce energy consumption in data centres by efficient utilization of resources.

In [31] the authors have proposed a VM resizing strategy with the aim of matching VM capacity with VM load in IaaS cloud. Their approach proposes a new VM capacity through time division multiplexing and adjustments proposed can be completed by existing virtualization technologies. The authors have compared their strategy with migration and they have reported that performance degradation resulting from their approach is much lower and for a short time as compared to VM migration.

In [32] a framework is presented for predicting performance loss and energy consumption caused by VM live migration. This framework is used to support decisions on auto-scaling and solve two VM consolidation sub-problems *1)* VM selection and *2)* VM allocation. VM selection for migration and host selection to host the migrated VM is decided after forecasting the resource usage and energy consumption before and after the live migration. This framework uses ARIMA model for prediction. ARIMA models have also been used in [33] to predict workload for achieving QoS.

Finally, in [29] the authors proposes a data centre resource usage prediction that is based on either Autoregressive Integrated Moving Average (ARIMA) or Autoregressive Neural Network (AR-NN). Real-time resources usage in a server is monitored at intervals and is used to forecast future resource demands. If the resource usage collected over time follows Gaussian distribution, ARIMA is used to forecast, otherwise, AR-NN is used. The authors report that AR-NN performs better than ARIMA on a number of accuracy metrics such as Mean Error (ME), Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE). A similar conclusion has also been made in [34,35]. Neural network techniques have been successful in predicting cloud resource usage as seen in [36,37,38,39,40,41].

## III. Cloud Model, System Architecture and Resource Prediction Model

In this section, we describe the target cloud service model for this work and the system components of our proposed solution.

### *A. Cloud service model*

The target cloud service model in this paper is a public IaaS multi-tenant cloud, which offers service to the public. In this cloud model, users request a VM to be created by selecting predefined machine types (sizes) such those provided by Google cloud [42]. The VMs are then created and place on available PMs the hypervisor. The user has full control of the VM and can run any type of applications in the VM. From the CSP point of view, applications are a black box host in a VM. However, in public clouds, users do not have access to VMM, only the CSPs do [13]. We assume that the CSP has put in place an effective real-time system for monitoring VM resource usage. We also propose a change to the default hypervisor resource scheduling policies such that a VM is allocated a specific CPU core until when migration is needed. This is feasible because a customer specifies the number of CPU cores before a VM is created. This idea is different from CPU affinity [43] but it is close to the technique proposed in [44]. The reason for having a fixed core for each VM is to achieve per-core DVFS [45] for each VM.

### *B. System architecture*

Our proposed architecture is shown in **Fig. 1** and its components are explained in this section. Our system components are separate from the core datacenter infrastructure to avoid unnecessary overhead on the core datacenter resources.

*1) VM size calculator: this component* receives resource usage harvested from the VM. It then analyzes VM resource usage and then determines the right size of the VM (VM fixed size). We have adopted VMware KB threshold setting for VM resource usage. We set CPU usage ceiling $CPU_{ceiling}$ and memory usage ceiling, $MEM_{ceiling}$ to represent the percentage of resource usage instances that were below $MEM_{ceiling}$ *%* and $CPU_{ceiling}$ *% for* memory and CPU respectively. The resource warnings for these resources are set at 5% above the ceiling. The percentile rankings, $R_{cpu}$ *and* $R_{ram}$, for the effective VM size is given by;

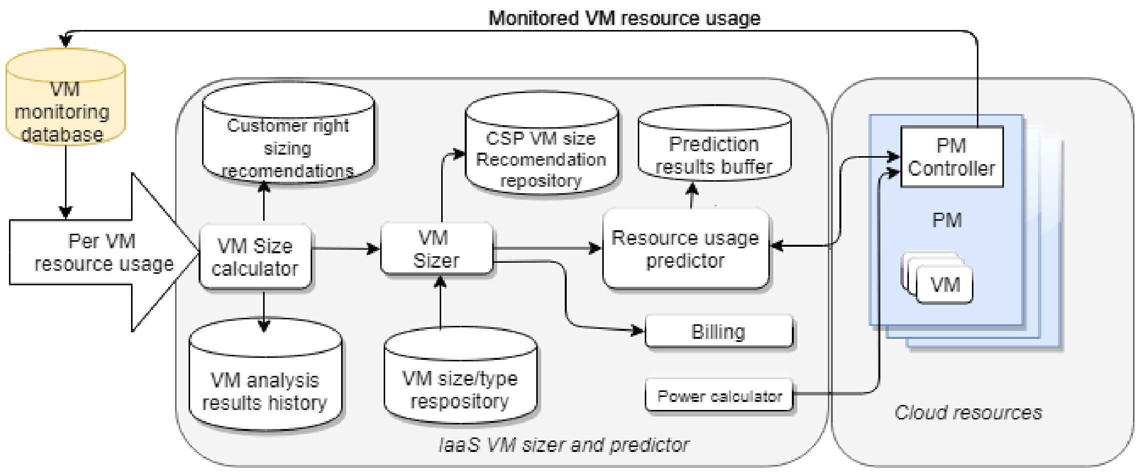
Monitored VM resource usage
VM monitoring database
Per VM resource usage
Customer right sizing recomendations
VM Size calculator
VM analysis results history
CSP VM size Recomendation repository
VM Sizer
VM size/type respository
Prediction results buffer
Resource usage predictor
Billing
Power calculator
IaaS VM sizer and predictor
PM Controller
PM
VM
Cloud resources

Fig.1. Proposed Architecture

$$R_{cpu} = \left\lceil \frac{CPU_{ceiling}}{100} * N \right\rceil \quad (2)$$

$$R_{ram} = \left\lceil \frac{MEM_{ceiling}}{100} * N \right\rceil \quad (3)$$

where $N$ is the total number of observations for either CPU or memory. The ranking, $R_{cpu}$ *and* $R_{ram}$ , are then used to get values, which are 80% of effective VM sizes. We have preferred to consider a ceiling for $R_{cpu}$ *and* $R_{ram}$ because resource warning is set above resource usage ceiling and not below and thus this is an advantage. If a function *f(.)* gives the values at rank $R_{cpu}$ *and* $R_{ram}$, the effective new values for VM CPU, *CPU,* and memory, *MEM*, are given in equation**s** (4) and **5** below.

$$CPU = \frac{5}{4} f(R_{cpu}) \quad (4)$$

$$MEM = \frac{5}{4} f(R_{ram}) \quad (5)$$

*2) VM analysis results history*: this is a database that stores the results from VM resource usage analysis.

*3) Customer right-sizing recommendations:* after VM size calculator generates new VM sizes, they are stored in this component for the customer to access.

*4) VM sizer:* this is a very important component in this architecture because it is responsible for implementing VM size recommendations from *VM sizer*. Recommended VM sizes are used to resize a particular VM according to **Algorithm 1** if the recommended size is not equal or close to current VM size. A VM to be used to resize a current VM is considered close to it if its size is not more than 5% of the current VMs. If memory or vCPU core addition or removal is required, we propose the use of CPU hot-plug for CPU and dynamic memory management for memory. In case a full vCPU core is not necessary, we propose per-core DVFS. The required CPU core frequency DVFS process, $f_{core}$, is determined according to equation (6) [46]. Any time a VM size is adjusted, billing for that VM is also adjusted.

**Algorithm 1: VM sizer operation**

```
Input: CalculatedVMSize_(cpu,ram), VmTypeList, CurrentVMSize_(cpu,ram)
1. rightVMSizeFoudInVmTypeList equals false
2. if CalculatedVMSize_(cpu,ram) approx. equals CurrentVMSize_(cpu,ram) then
3.         return; //nothing should happen
4. else
5.         for each vmType in VmTypeList do
6.                 if CalculatedVMSize_(cpu,ram) approx. equals vmType_(cpu,ram) then
7.                         rightVMSizeFoudInVmTypeList equals true
8.                 end if
9.         end for
10.        if rightVMSizeFoudInVmTypeList equals true then
11.                Scale resources_(cpu,ram) to new size using hot-plug
12.        else
13.                Scale resource_(ram) using hot-plug
14.                For resource_(cpu), add cores OR reduce cores OR apply per-core DVFS as appropriate
15.                Store new recommendations in CSP VM size recommendation repository
16.        end if
17.        Adjust billing
18.        Pass VM new details to resource usage predictor
19         return;
20.    end if
```

$$f_{core} = \frac{(f_{high} * t_{high}) + (f_{low} * t_{low})}{t_{high} * t_{low}} \quad (6)$$

where $f_{high}$ the highest frequency, $t_{high}$ the number of occurrences of the high frequency and similarly for the low frequency.

*5) VM type/size repository*: this is a database of all VM types that have been preconfigured by the CSP. The customer picks a VM type for this repository the first time they request VM creation.

*6) CSP VM size recommendation repository:* if the *VM sizer* cannot get a VM type/size from *VM type/size repository* as recommended by *VM size calculator*, it will recommend that this VM type is availed. This recommendation will be stored in this repository for CSP's access.

*7) Resource usage predictor*: This component is used to predict future VM resource usage. In the next subsection, we have used artificial neural networks (ANN) and not ARIMA for resource usage prediction because our time series data does not follow a normal distribution. ANN has been elaborated in subsection (d) of this section. The aim of this component is to solve a problem, which can be stated as follows; *for a VM, given a time t and a history of VM resource utilization before t, we can forecast resource utilization of the VM at time t.* In this paper, we focus on one resource, CPU, for simplicity. The CPU capacity, $CPU_{add_{(t)}}$, which needs to be added to the VM at time $t$ is determined according to Equation (7).

$$CPU_{add(t)} = CPU_{utilization(t)} + CPU_{fixed} \quad (7)$$

where $CPU_{utilization_{(t)}}$ is the total CPU capacity required by the VM at time $t$ and $CPU_{fixed}$ is the fixed CPU capacity of the VM. Prediction results are stored in a *Prediction results buffer*.

*8) Billing*: this component is used to calculate and report the cost of running a VM. The cost of running a VM, $C_{total}$, is determined according to equation (8).

$$C_{total} = C + (\alpha_{(ram,cpu)} - \beta_{(ram,cpu)}) + \mu \quad (8)$$

where $C$ is the cost of the fixed size VM, $\alpha_{(ram,cpu)}$ is the cost of resources borrowed at spiky times, $\beta_{(ram,cpu)}$ is the cost of resources lent to other VMs and $\mu$ are other costs associated with the VM such choice of operating system, data centre region, extra services like use of snapshot and multiple Internet Protocol (IP) address.

*9) PM controller*: this component runs in the physical machine and is responsible for monitoring resource usage of the VM via the virtualization layer and stores it in the *VM monitoring database*. This component can also give the *resource usage predictor* and *power calculator* access to PM real-time resource usage in the PM.

*10) Power calculator:* this is a simple component that estimates the power consumed by all active hosts at any given time *t* during execution of an application. Total power is given by a model shown in equation (9).

$$P_{total} = \sum_{i=1}^{k} \left( (P_i^{'} - P_i) * (\frac{n_i}{100}) + P_i \right) \tag{9}$$

where *k* is the number of active hosts at any time, *P'* is the maximum power consumption of the $i^{th}$ host, *P* is the power consumed by the host when completely idle and *n* is the percentage CPU utilization of the host. Energy, *E*, can be calculated as shown in equation (10).

$$E = PT \tag{10}$$

where *P* is average power consumption (in watts) and *T* is a time (in seconds) interval.

*11) Prediction results buffer*: this component is used to hold output results, which are prediction values from the *Resource usage predictor*.

## *C. Workload data cleaning, preprocessing and initial characterization*

In this subsection, we present selected useful statistics about the dataset we will use in this paper. The complete workload description and nature of the workload can be found in [27]. The first set of Materna dataset shows resource provisioned and used for a total of 520 VMs.

Before characterization, data has to be cleaned. In this paper, we have cleaned our data because of the following reasons;

*1) Wrong data format:* some columns in the VM data had formats that could not work for our architecture. For instance, the columns showing percentage CPU and memory usage had values with a comma in place of a decimal point. In addition, the timestamp column had a date and time format that could not be used in the Waikato Environment for Knowledge Analysis (WEKA). The date format was has been converted to WEKA's data and time format and commas replaced with decimal points in CPU and memory usage percentages. Normally, WEKA works well with csv files, but due to format issues with some columns, we have converted all our CSV files into WEKA's ARFF file format.

*2) Missing information*: for some reasons, such as errors in computer program that generated CVS files or lack of system monitoring, some data may be missing. In our workload data, some column's data was missing and thus indicated as zero. In our case, for some VM CSV, the column showing CPU capacity provisioned in MHz had zeros. This was obviously an omission since the VM had executed workloads and its CPU usage had consistent values. This column values, $CPU_{provisioned}$, is calculated according to equation (11).

$$CPU_{provisiond} = \frac{CPU_{used}}{CPU_{\%}} * 100 \tag{11}$$

where $CPU_{used}$ is the value in the 'CPU usage' column and $CPU_{\%}$ is the value in 'CPU usage [%]' column.

In **Fig. 2**, we have shown a time series for VM 405 in our dataset. Because of the time series has a higher frequency, we apply a 150-point simple moving average (SMA), filter to remove the higher noise frequency

The resultant SMA filtered time series is shown in **Fig. 3.** We have noticed that the resource usage of almost all VMs is very low. For instance, **Fig. 4** shows memory provisioned and memory that was used to execute workloads for VM 01. It is clear that the VMs' memory usage was extremely low as compared to allocated memory. Summarily, the average CPU and memory usage for the 520 VMs is 4.5% and 8.3% respectively. In fact, the percentage of VMs with CPU and memory utilization below 20% is 96.3% and 90.6% respectively. The average highest VM CPU and memory usage stand at 69.3 % and 82.2 % respectively.

The CPU cores provisioned to the VMs are 1, 2, 4, 6 and 8. 78.8% of the VMs have either 1 or 2 cores and the rest have 4 or 6 or 8 cores. Average resource usage peak to mean ration, which can be used to show dynamicity [22] is 16:1 for CPU and 10:1 for memory. **Fig. 5-8** shows CPU and memory usage's 90th percentile for various VMs (01, 45, 467 and 492). We have shown these because we had indicated earlier that our VM sizing techniques use percentiles. In **Fig. 9**, we have shown a frequency distribution plot of memory usage for VM 172. It is seen that the distribution has a positive skewness (skew value = 1.9021426). Skew values for VM 45 CPU usage, VM 172 CPU usage, VM 45 memory usage are 5.2309817, 2.670302 and 1.6542248 respectively. We have computed autocorrelation for CPU and memory for a lag value ranging from 5 minutes (lag 1) to 2 hours and all were very low except for lags that are multiples of 3 minutes (lag 3, lag 6, lag 9,…), which had a value of 0.3 to 0.4.

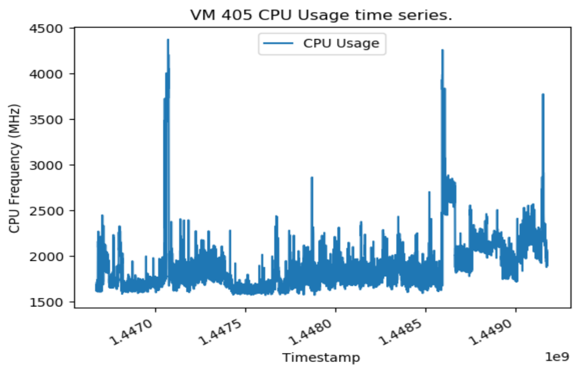

Fig.2. CPU usage time series for VM 405

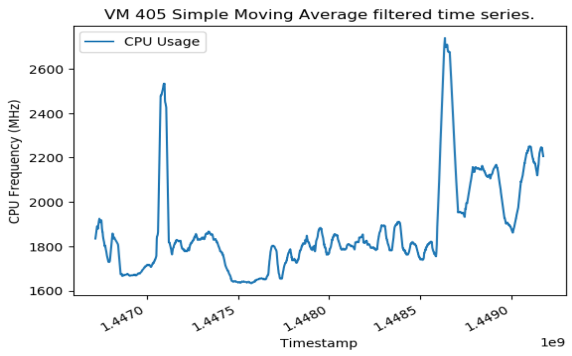

Fig.3. CPU usage Simple Moving Average (window=150) filter series for VM 405

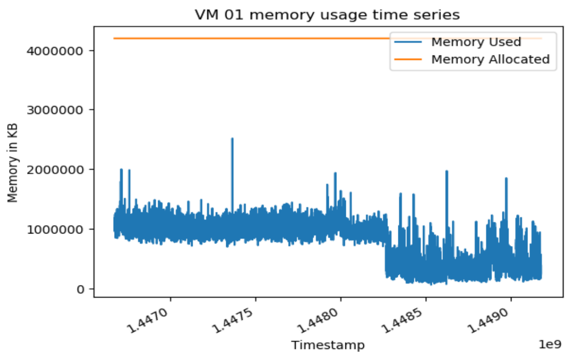

Fig.4. A plot of memory allocated, and memory used for VM 01

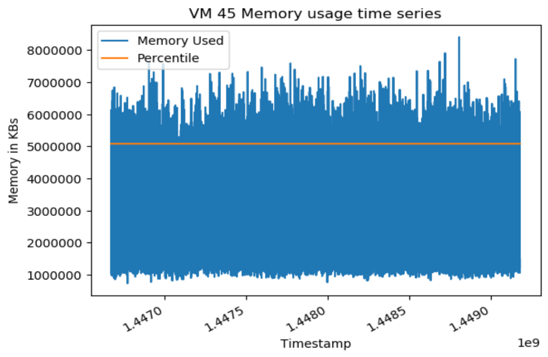

Fig.5. A plot of memory usage showing 90th percentile for VM 45

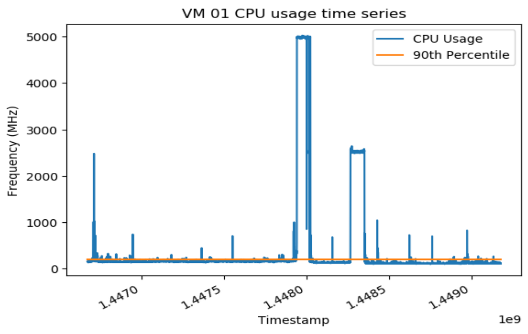

Fig.6. A plot of CPU usage showing 90th percentile for VM 01

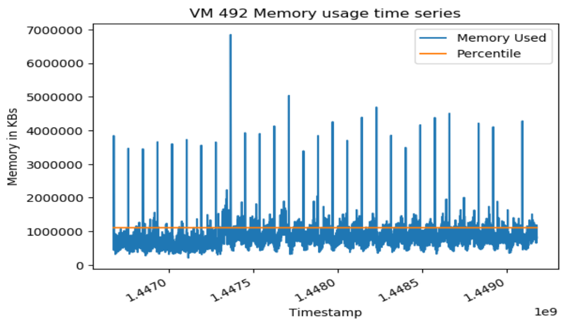

Fig.7. A plot of memory usage showing 90th percentile for VM 492

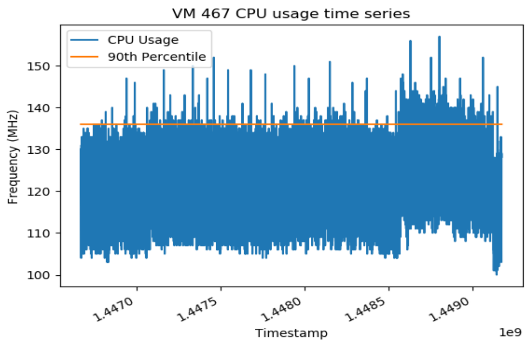

Fig.8. A plot of CPU usage showing 90th percentile for VM 467

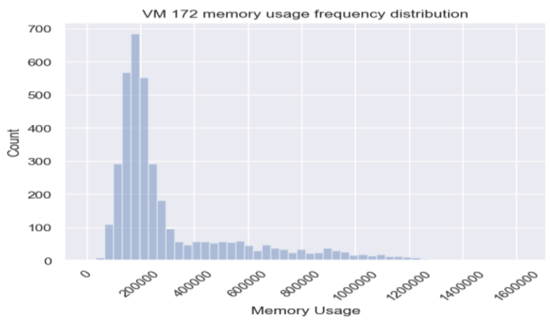

Fig.9. Frequency distribution of memory usage for VM 172

Additionally, Table 1 shows Jarque-Bera results for memory usage from VM 172. The computed p value is less than the alpha value (0.05), thus we conclude that the variable is not normally distributed

Table 1. Jarque-Bera (JB) test results for memory usage of VM 172

| Item | Value |
|---|---|
| JB | 3802.958114 |
| *p* value | 0 |
| Alpha | 0.05 |

For each of the 520 VMs, we have identified all *peak points,* their corresponding timestamps and if there are peaks in other VM, which occur at the same time (**see algorithm 2**). We define a *peak point* of a VM as any point whose value is higher than a 90th percentile value. We have shown that, indeed, there are peaks happening simultaneously but their percentage is very low, which opens the opportunity of using statistical multiplexing [47], exploited where unutilized resources of one VM can be borrowed by a co-located VMs. For instance, out of the 520 VMs (with over 4.3 million data points), a memory peak at timestamp 1.447967e+09 happened simultaneously only in 25% of the VMs and it was the highest. For CPU, a peek at timestamp 1.449137e+09 happened simultaneously only in 24% of the VMs and it was the highest. Other peaks for CPU and memory occur simultaneously in less than 24% and 25% respectively. These results mean that resource over-commitment is never a problem with workloads, which do not peak simultaneously.

**Algorithm 2: Finding the number of VM peaks that occur simultaneously**

```
Input: # of VMs n, a set of time series for each VM $Y_t = \{Y_{t_1}, Y_{t_2}, ....Y_{t_{n-1}}, Y_{t_n}\}$
Output: uniquePeakCountList // a list of the frequency of ‘peak points’
foreach $y_t$ in $Y_t$ set do
  percentileValue = get90thPercentileFor$y_t$ ()
  foreach value in $y_t$ do
    if the value is greater than percentileValue then
      get timestamp for this value
      add timestamp to list timestampList variable
        /*now percentileValue has peeks for all the VM time series. If a timestamp
        appears x times, it means there was a peek at that timestamp in x VMs */
    end if
  end for
end for
peakUniqueList = timestampList.getUnique //get all unique timestamps from list
/*find the number of times each unique appears in timestampList */
foreach unique in peakUniqueList do
  count = timestampList.count(unique) //the frequency of unique in timestampList
  //add count to uniquePeakCountList
  uniquePeakCountList.add(count).
end for
return uniquePeakCountList
```

## *D. Datacenter resource prediction using neural networks*

ANN is a suitable method of prediction in our case because we have shown that it data has a non-Gaussian distribution. It has also shown good performance than other prediction models [29,38]. Moreover, ANN has a great ability to model a non-linear function thus able to handle complex time series [39,48,49]. Our ANN architecture has three layers, which are an input layer, hidden layer and an output layer as shown in **Fig. 10**.

*1) Input layer*: the input layer receives a multivariate time series – each input variable is a time series. These attributes are contained in VM data in our dataset in CSV format. The inputs used include CPU usage ($x_{cpu}$), memory usage ($x_{mem}$), Disk read throughput ($x_r$) , disk write throughput ($x_w$), Network received throughput ($x_{neti}$) and network transmitted throughput ($x_{neto}$) represented as $x_{inputs} = \{x_{cpu}, x_{mem}, x_r, x_w, x_{neti}, x_{neto}\}$. ). Feature reduction has been accomplished by using WEKA machine learning tool taking *CPU usage* as the target class [50,51]. The least correlated attributes to the target class are eliminated: CPU capacity provisioned, memory capacity provisioned, disk size, memory usage %, and CPU usage %. A bias node is also included in this layer to ensure that the network will be able to fit that data by avoiding null values.

*2) Hidden layer*: the hidden layer serves to improve prediction accuracy and consists of nodes with activation functions.

*3) Output layer*: the output layer is used to produce outputs from ANN.

For our inputs from above, the output, $O_j$ of node $j$ in our hidden layer at a time, $t$, is given by;

$$O_{i,t} = g\left(w_{0,j} + \sum_{i=1}^{p} w_{i,j}\cdot x_{input_i}\right) = \frac{1}{1+e^{-\left(w_{0,j}+\sum_{i=1}^{p} w_{i,j}\cdot x_{input_i}\right)}} \tag{12}$$

Thus, the relationship between the output, $y_t$, at time $t$, and input is given by;

$$y_t = w_0 + \sum_{j=1}^{q} w_j \cdot g\left( w_{0,j} + \sum_{i=1}^{p} w_{i,j} \cdot x_{input(t-i)} \right) + \varepsilon_t \qquad (13)$$

where *g(.)* is a logistic function, $\varepsilon_t$ is an error of the model, $w_{ij}(i=1,2,\ldots p,\ j=1,2,\ldots q)$ $w_j(j=0,1,2,\ldots q)$ are connection weights, which are parameters of the model, *p* is the number of input nodes and *q* is the number of hidden nodes. Specifically, $w_{0,j}$ is the bias included to the input nodes and $w_0$ is the bias added to the output of hidden nodes. The initial values to the parameters represent a state of knowledge. We partition our input data so that 90% of input is used for training and 10% is used for testing. We use a 90:10 ratio because we have a substantive amount of data points – each VM has at least 8352 points. The predicted values in our model have been tested on 4 accuracy metrics i.e. Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE) and Success Rate (SR). SR is the percentage of all predictions which are equal or greater than the actual value.

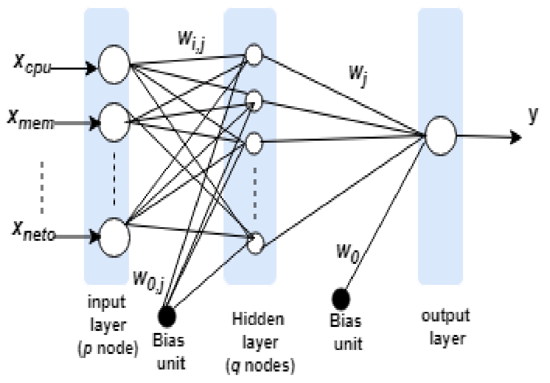

Fig.10. Neural network architecture

## IV. Experiment Set Up

In this paper, the evaluation experiment is divided into two parts;- 1) VM sizing and 2) VM resource prediction. The design of the VM sizing technique has been explained in section III and we have evaluated the energy savings resulting from our technique using Cloudsim Plus cloud simulator. We have configured a datacenter in this simulator according to the characteristics in Table 2. The datacenter host resources have been set to satisfy the demands of the VMs before and after VM sizing.

The workloads used in this experiments is Grid Workload Archive Trace 13 (GWA-T-13) Materna cloud, which is well explained in [27]. We executed the workloads in the datacenter and then measure the energy consumption before and after VM sizing using well known First Fit (FF), Worst Fit (WF) and Best Fit (BF) VM allocation algorithms.

For the VM resource usage prediction, we have evaluated the performance of the use of ANN prediction using WEKA. An ANN model is trained using the following parameters: number of hidden layers = 1, learning rate = 0.3, momentum = 0.2 and training time or epoch = 1000. The model prediction performance has been tested using the accuracy metrics described in section III (A) above.

Table 2. Cloudsim Plus datacenter configuration before and after VM sizing.

| Item | Before VM sizing | After VM sizing |
|---|---|---|
| No. of hosts | 49 | 28 |
| No. of VMs | 520 | 520 |
| No. of CPU cores | 1298 | 535 |
| Memory size (in GB) | 6780 | 4142 |
| Hypervisor | VMware ESX | VMware ESX |
| No. of cores allocated per VM | Varying (1,2,4,6 and 8) | Varying (1, 2 & 3) |
| Memory size allocated per host (in GB) | Varying (2,4,8 and 16) | Varying (1, 2, 4, & 6) |
| Host static power | 60 % of host peak power | 60 % of host peak power |

Next, we present and explain the evaluation results of our VM sizing and VM resource usage approaches.

## V. Results and Discussion

As a result of VM sizing, Table 2 shows that we have theoretically reduced CPU cores from 1298 cores to 535 cores and memory from 6780 GB to 4142 GB for processing workloads for the 520 VMs. Consequently, the number of VM hosts in the datacenter has reduced from 49 to 28. The reduction of the amount of resources required to process workloads definitely reduces the fixed cost of running VM workloads in the cloud. This cost is the component *C* from equation (8). Assuming $\mu$ does not change after VM sizing *(α-β)* cancels out within the multitenant cloud from the CSP point of view. In addition, the reduction in the number of hosts in the datacenter reduces the amount of energy consumed. Fig. 11 shows the amount of energy consumed when using WF, BF and FF VM allocation algorithms before and after VM sizing.

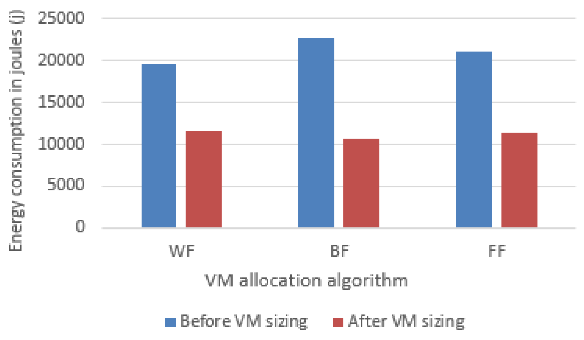

Fig.11. A comparison of energy consumption in datacenter before and after VM sizing using WF, BF and FF VM allocation algorithms

It is can be observed that the energy consumed to process the same workload has reduced under the different VM allocation algorithms. These results show that VM sizing has a big potential in managing cloud costs and energy consumption and thus opens up for more inquiries to design more techniques on the same.

Table 3. ANN prediction performance metrics on VM 01, VM 172, VM 405, VM 467 and VM 492

| **VM No.** | **Performance metrics** | **Resource considered** | |
|---|---|---|---|
| | | CPU | Memory |
| **01** | MAE | 23.1 | 183012.5 |
| | MAPE | 13.8 | 17.9 |
| | RMSE | 109.4 | 219973.7 |
| | SR | 61.1 | 72.5 |
| **172** | MAE | 6.7 | 142359 |
| | MAPE | 5.8 | 58.9 |
| | RMSE | 17.7 | 201971 |
| | SR | 31 | 39.8 |
| **405** | MAE | 198.9 | 518969.5 |
| | MAPE | 9.5 | 24.6 |
| | RMSE | 265.8 | 742942.4 |
| | SR | 87.3 | 36.6 |
| **467** | MAE | 7.5 | 106509 |
| | MAPE | 6.1 | 28.7 |
| | RMSE | 13.4 | 143803 |
| | SR | 43 | 57.6 |
| **492** | MAE | 34.7 | 158126.8 |
| | MAPE | 31.9 | 20.4 |
| | RMSE | 66.1 | 232459.2 |
| | SR | 90.4 | 67.7 |

Our ANN model is evaluated by predicting future 835 instances. We have evaluated the performance of the model on actual values and predicted values on a number of performance metrics and on 5 different VMs (VM 01, VM 172, VM 405, VM 467 and VM 492). Table 3 summarizes the performance metrics.

For all the VMs that we considered, the MAPE of is below 32%, which shows a good prediction performance considering the dynamic nature of the cloud. In fact, the MAPE of the model for CPU on VM 01, VM 172, VM 405 and VM 467 is below 14%, which is impressive. The MAE for the prediction memory shows a good performance – the values for memory seems bigger but it is because the memory unit of measurement is in Kilobytes.

We have also computed SR, which measures the degree to which a VM is likely not to borrow resources from other neighbouring VMs. This is not a problem because, if a VM demands more resources than was predicted, it can use resources from other more idle VMs. This is called statistical multiplexing and is possible as seen from algorithm 2.

We have also presented graph plots that visually compare predicted and actual values of CPU and memory for different VMs (Fig. 12 - 14). The graph plots only show a portion of the predicted values with the order of observations is preserved. The results from Fig. 14 – 16 and those from Table 3 shows that the model performs differently on the different VMs. For instance, CPU prediction for VM 172 (Fig. 14) is more accurate than that of VM 01 (Fig. 12). This means that each VM's model will be generated independently.

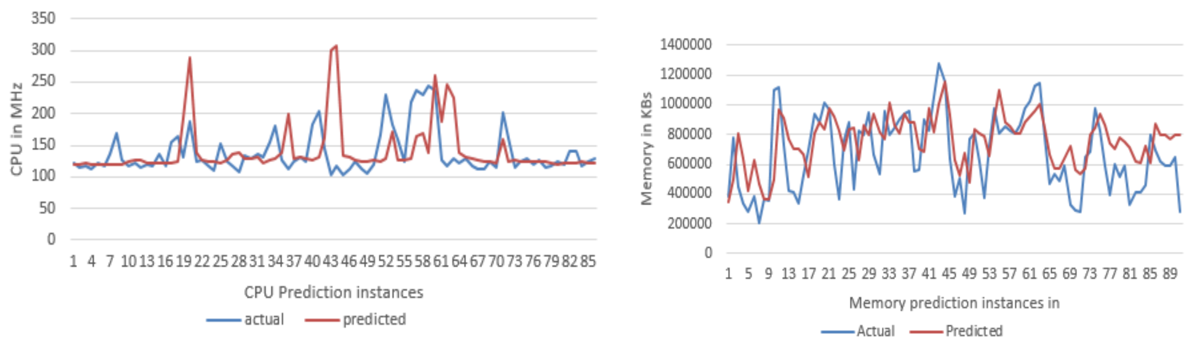

Fig.12. Graphical representation of a comparison between predicted and actual values for CPU and memory for VM 01

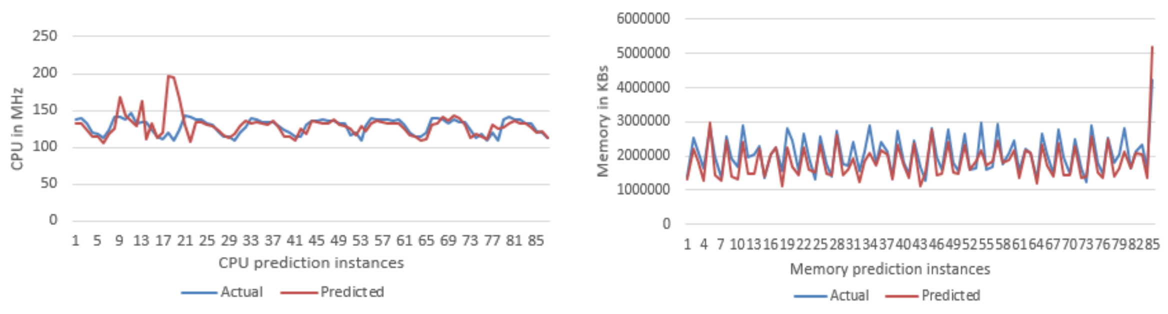

Fig.13. Graphical representation of a comparison between predicted and actual values for CPU and memory for VM 405

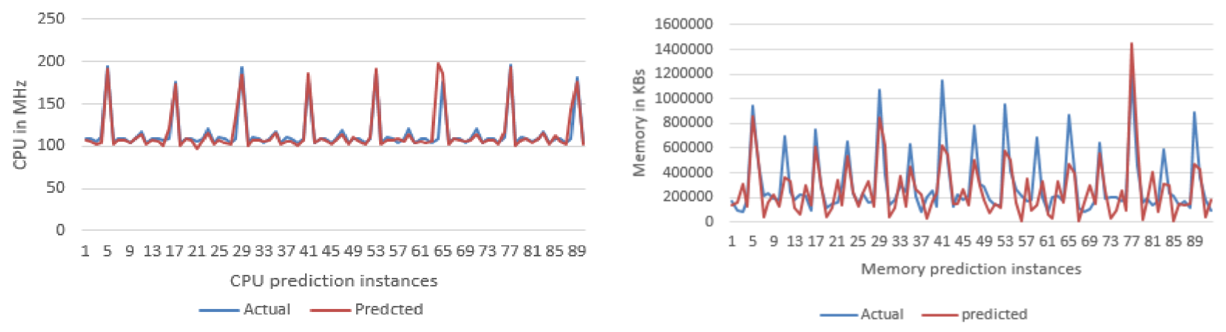

Fig.14. Graphical representation of a comparison between predicted and actual values for CPU and memory for VM 172

## VI. Conclusion

In this paper, we have proposed an approach for automatic VM sizing and VM resource usage prediction for multitenant IaaS cloud. In our proposed architecture, historical resource usage for a VM are extracted and are used to determine a fixed size for that VM. Due to the dynamicity of the cloud, we have proposed an approach for future resource usage prediction using the ANN model to foresee resource usages that may exceed the fixed ones.

Since VM resource usage does not peak at the same time, VM resources for one VM can be borrowed by another VM, a technique known as statistical multiplexing. We have evaluated our VM sizing on Cloudsim Plus cloud simulator using real workload trace and results show that VM sizing can improve resource utilization and thus reduce cloud costs as well as energy wastage. Additionally, we have evaluated our ANN model on WEKA using real workload trace and results show that ANN model can achieve good levels accuracy even on a time series data that does not follow a normal distribution. As future work, we wish to increase the scope to cover techniques such as deep learning and applying our technique on a wider range of cloud workloads harvested from different infrastructures.

## References

[1] Salam, R. Karim and M. Ali, "Proactive dynamic virtual-machine consolidation for energy conservation in cloud data centres," *Journal of Cloud ComputingAdvances, Systems and Applications.*

[2] G. Albert, H. James, A. M. David and P. Parveen, "The cost of a cloud: research problems in data centre networks," *The ACM Digital Library is published by the Association for Computing Machinery,* vol. 39, no. 1, 2009.

[3] Khosravi, "Energy and Carbon-Efficient Resource Management in Geographically Distributed Cloud Data Centers," The University of Melbourne, Melbourne, Australia, 2017.

[4] F. P. Sareh, "Energy-Efficient Management of Resources in Enterprise and Container-based Clouds," The University of Melbourne, 2016.

[5] J. Patel, V. Jindal, I.-L. Yen, F. Bastani, J. Xu and P. Garraghan, "Workload Estimation for Improving Resource Management Decisions in the Cloud," in *2015 IEEE Twelfth International Symposium on Autonomous Decentralized Systems*, Taichung, Taiwan, 2015.

[6] B. Anton, "Energy-Efficient Management of Virtual Machines Data Centers for Cloud Computing," The University of Melbourne, 2013.

[7] M. Dabbagh, B. Hamdaoui, M. Guizani and A. Rayes, "Toward energy-efficient cloud computing: Prediction, consolidation, and overcommitment," *IEEE Network*, vol. 29, no. 2, 2015.

[8] G. Hadi and P. Massoud, "Achieving Energy Efficiency in Datacenters by Virtual Machine Sizing, Replication, and Placement," in *Energy Efficiency in Data Centers and Clouds*, Elsevier Science, 2016.

[9] R. Neha and J. Rishabh, "Cloud Computing: Architecture and Concept of Virtualization," *International Journal of Science, Technology & Management*, vol. 4, no. 1, 2015.

[10] B. Carmody, "Infrastructure On Demand Is Giving Small Businesses An Edge," Inc, 2018. [Online]. Available: https://www.inc.com/bill-carmody/infrastructure-on-demand-is-giving-small-businesses-an-edge.html. [Accessed 01 OCtober 2018].

[11] F. P. Sareh, R. N. Calheiros, J. Chan, A. V. Dastjerdi and R. Buyya, "Virtual Machine Customization and Task Mapping Architecture for Efficient Allocation of Cloud Data Center Resources," *The Computer Journal*, 2015.

[12] R. Hu, J. Jiang, G. Liu and L. Wang, "Efficient Resources Provisioning Based on Load Forecasting in Cloud," *The Scientific World Journal*, vol. 2014, no. 321231, 2014.

[13] D. Jiaqing, S. Nipun and Z. Willy, "Performance profiling in a virtualized environment," in *HotCloud'10 Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, Boston, USA, 2010.

[14] ParkMyCloud, "Why Azure Right Sizing is Important," ParkMyCloud, 2018. [Online]. Available: https://www.parkmycloud.com/azure-right-sizing/. [Accessed 01 November 2018].

[15] Google, "Applying Sizing Recommendations for VM Instances," Google, 2018. [Online]. Available: https://cloud.google.com/compute/docs/instances/apply-sizing-recommendations-for-instances. [Accessed 1 November 2018].

[16] Amazon Web Services, "Right-Sizing: Provisioning Instances to Match Workloads: AWS Whitepaper," Amazon Web Services, Inc., 2018.

[17] V. Patel and H. Bheda, "Reducing Energy Consumption with Dvfs for Real-Time Services in Cloud Computing," *IOSR Journal of Computer Engineering (IOSR-JCE)*, vol. 16, no. 3, pp. 53-57, 2014.

[18] VMware, "vSphere Resource Management," VMware, Inc, Palo Alto, CA, 2015.

[19] X. Bronson, R. P. and S. S. Raja, "A Dynamic Memory Allocation Strategy for Virtual Machines in Cloud

Platform," i*nternational Journal of Pure and Applied Mathematics*, vol. 119, no. 15, pp. 1423-1444, 2018.

[20] S. Perera, "Multi-tenancy after 10 years of Cloud Computing," Hackernoon, 2016. [Online]. Available: https://hackernoon.com/multi-tenancy-after-10-years-of-cloud-computing-19de782ef899. [Accessed 01 November 2018].

[21] VMware, "Performance Best Practices for VMware vSphere 6.0," VMware, Inc, Palo Alto, CA, 2015.

[22] S. Shen, V. v. Beek and A. Iosup, "Statistical Characterization of Business-Critical Workloads Hosted in Cloud Datacenters," in *2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, Shenzhen, China, 2015.

[23] P. Xuesong, P. Barbara and V. Monica, "Virtual Machine Profiling for Analyzing Resource Usage of Applications," in *International Conference on Services Computing*, Milano, Italy, 2018.

[24] S. K. Tesfatsion, "Energy-efficient cloud computing: Autonomic resource provisioning for datacenters," Umea University, Umea, 2018.

[25] S. S. David and A. R., "Autonomic Resource Provisioning Algorithm for Cloud Computing using Match Making Technique," *International Journal of Advanced Research in Computer Science and Software Engineering* , vol. 6, no. 9, pp. 168 -173, 2016.

[26] G. ̈ Urul, "ENERGY EFFICIENT DYNAMIC VIRTUAL MACHINE ALLOCATION WITH CPU USAGE PREDICTION IN CLOUD DATACENTERS," Bilkent University, 2018.

[27] Delf University, "The Grid Workloads Datasets," Delf University, 2018. [Online]. Available: http://gwa.ewi.tudelft.nl/datasets/. [Accessed October 2 2018].

[28] M. Amiri and L. Mohammad-Khanli, "Survey on prediction models of applications for resources provisioning in cloud," *Journal of Network and Computer Applications*, vol. 82, 2017.

[29] Q. Z. Ullah, S. Hassan and G. M. Khan, "Adaptive Resource Utilization Prediction System for Infrastructure as a Service Cloud," *Journal of Computational Intelligence and Neuroscience: Hidawi*, vol. 2017, 2017.

[30] M. Chen, H. Zhang, Y.-Y. Su, X. Wang, G. Jiang and K. Yoshihira, "Effective VM sizing in virtualized data centres," in *12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011) and Workshops*, Dublin, Ireland , 2011.

[31] R. Hu, G. Liu, J. Jiang and L. Wang, "A New Resources Provisioning Method Based on QoS Differentiation and VM Resizing in IaaS," *Journal of Mathematical Problems in Engineering - Hidawi*, vol. 2015, no. 215147, 2015.

[32] M. Aldossary and K. Djemame, "Performance and Energy-based Cost Prediction of Virtual Machines Live Migration in Clouds," in *Proceedings of the 8th International Conference on Cloud Computing and Services Science. 8th International Conference on Cloud Computing and Services Science*, Madeira, Portugal, 2018.

[33] R. N. Calheiros, E. Masoumi, R. Ranjan and R. Buyya, "Workload Prediction Using ARIMA Model and Its Impact on Cloud Applications' QoS," *IEEE TRANSACTIONS ON CLOUD COMPUTING*, vol. 3, no. 3, pp. 449-458, 2016.

[34] J. Xue, F. Yan, R. Birke, L. Chen, T. Scherer and E. Smirni, "PRACTISE: Robust prediction of data centre time series," in *2015 11th International Conference on Network and Service Management (CNSM)*, Barcelona, Spain, 2015.

[35] Mozo, B. Ordozgoiti and S. Gómez-Canaval, "Forecasting short-term data center network traffic load with convolutional neural networks," PLOS one, 2018.

[36] J. J. Prevost, K. Nagothu, B. Kelley and M. Jamshidi, "Prediction of Cloud Data Center Networks Loads Using Stochastic and Neural Models," in *Proceeding of the 2011 6th International Conference on System of Systems Engineering*, Albuquerque, New Mexico, USA, 2011.

[37] S. Frey, S. Disch, C. Reich, M. Knahl and N. Clarke, "Cloud Storage Prediction with Neural Networks," in *The Sixth International Conference on Cloud Computing, GRIDs, and Virtualization*, 2015.

[38] M. Duggan, K. Mason, J. Duggan, E. Howley and E. Barrett, "Predicting Host CPU Utilization in Cloud Computing using Recurrent Neural Networks," in *The 8th International Workshop on Cloud Applications and Security*, 2017.

[39] H. Xu, X. Zuo, C. Liu and X. Zhao, "Predicting Virtual Machine's Power via a RBF Neural Network," in *International Conference in Swarm Intelligence*, Bali, Indonesia, 2016.

[40] Y. Lu, J. Panneerselvam, L. Liu and Y. Wu, "RVLBPNN: A Workload Forecasting Model for Smart Cloud Computing," *Scientific Programming: Hidawi*, vol. 2016, no. 5635673, 2016.

[41] R. Cao, Z. Yu, T. Marbach, J. Li, G. Wang and X. Liu, "Load Prediction for Data Centers Based on Database Service," in *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, Tokyo, Japan, 2018.

[42] Google, "Machine types: Google compute documentation.," Google, 2018. [Online]. Available: https://cloud.google.com/compute/docs/machine-types. [Accessed November 02 2018].

[43] C. Ribeiro, M. Castro, M.-M. Vania and J.-F. Méhaut, "Evaluating CPU and Memory Affinity for Numerical Scientific Multithreaded Benchmarks on Multi-cores," *IADIS International Journal on Computer Science and Information Systems*, vol. 7, no. 1, pp. 79-93, 2012.

[44] J. M. Szefer, "Architectures for Secure Cloud Computing Servers," Princeton University, 2013.

[45] W. Kim, M. S. Gupta, G.-Y. Wei and D. Brooks, "System level analysis of fast, per-core DVFS using on-chip switching regulators," in *2008 IEEE 14th International Symposium on High Performance Computer Architecture*, Salt Lake City, UT, USA, 2018.

[46] C. M. Kamga, "CPU frequency emulation based on DVFS," in *2012 IEEE Fifth International Conference on Utility and Cloud Computing*, Chicago, IL, USA, 2013.

[47] X. Meng, X. Meng, X. Meng, X. Meng, X. Meng and X. Meng, "Efficient resource provisioning in compute clouds via VM multiplexing," in *Proceedings of the 7th international conference on Autonomic computing* , Washington DC, USA , 2010.

[48] G. Shmueli, R. P. Nitin, C. B. Peter, I. Yahav and C. L. Kenneth, "Neural nets," in *Data Mining for Business Intelligence: Concepts, Techniques, and Applications in Microsoft Office Excel with XLMiner*, Wiley, 2017, p. 271.

[49] M. Khashei and M. Bijari, "An artificial neural network (p, d, q) model for timeseries forecasting," *Expert Systems with Applications: Elsevier*, vol. 37, no. 1, pp. 479-489, 2010.

[50] Waikato University, "Machine Learning at Waikato University," Waikato University, 2018. [Online]. Available: https://www.cs.waikato.ac.nz/ml/index.html. [Accessed 25 November 2018].

[51] S. Karsoliya, "Approximating Number of Hidden layer neurons in Multiple Hidden Layer BPNN Architecture," *International Journal of Engineering Trends and Technology,* vol. 3, no. 6, pp. 714-717, 2012.

## Authors' Profiles

**Derdus M. Kenga** is a PhD candidate in the Faculty of Information Technology of Strathmore University, Kenya. He completed his Bachelor Degree in Computer Science from Moi University, Kenya (2012) and MSc. In Mobile Telecommunications and Innovation from Strathmore University (2014). His research areas is cloud computing and health informatics.

**Vincent O. Omwenga** is a senior lecturer and an academic director in the Faculty of Information Technology of Strathmore University, Kenya. He earned his PhD in Mathematical Statistics from The University of Nairobi, Kenya. His areas of interest is systems modelling, computational models and cloud computing.

**Patrick J. Ogao** is an Associate Professor of visualization and geoinformatics. He earned a PhD from Utrecht University, where he studied visual exploratory environments and techniques for knowledge discovery in large multi-dimensional, time-varying spatial data sets and an MSc from ITC, University of Twente, Enschede, in The Netherlands. He has had previous appointments in Computer Science Departments of University of Groningen, The Netherlands, Makerere University, Uganda, and Masinde Muliro University of Science and Technology in Kenya. His research areas are in Information Systems and Information Visualization and consults regionally in Information Systems Development and geospatial sciences.

**How to cite this paper:** Derdus M. Kenga, Vincent O. Omwenga, Patrick J. Ogao, "Autonomous Virtual Machine Sizing and Resource Usage Prediction for Efficient Resource Utilization in Multi-Tenant Public Cloud", International Journal of Information Technology and Computer Science(IJITCS), Vol.11, No.5, pp.11-22, 2019. DOI: 10.5815/ijitcs.2019.05.02