Journal of Information Technology and Computer Science
Volume 6, Number 2, August 2021, pp. 170-181
Journal Homepage: www.jitecs.ub.ac.id

Virtual Machine Customization Using Resource Using Prediction for Efficient Utilization of Resources in IaaS Public Clouds

Derdus Kenga*[1], Vincent Omwenga [2], Patrick Ogao [3]

[1,2] Faculty of Information Technology, Strathmore University Nairobi, Kenya
[3] School of Computing and Information Technology, Technical University of Kenya, Nairobi, Kenya
[1]derduskenga@gmail.com, [2]vomwenga@strathmore.edu
[3]ogaopj@gmail.com
*Corresponding Author

Received 24 April 2020; accepted 03 August 2021

Abstract. The main cause of energy wastage in cloud data centres is the low level of server utilization. Low server utilization is a consequence of allocating more resources than required for running applications. For instance, in Infrastructure as a Service (IaaS) public clouds, cloud service providers (CSPs) deliver computing resources in the form of virtual machines (VMs) templates, which the cloud users have to choose from. More often, inexperienced cloud users tend to choose bigger VMs than their application requirements. To address the problem of inefficient resources utilization, the existing approaches focus on VM allocation and migration, which only leads to physical machine (PM) level optimization. Other approaches use horizontal auto-scaling, which is not a visible solution in the case of IaaS public cloud. In this paper, we propose an approach of customizing user VM's size to match the resources requirements of their application workloads based on an analysis of real backend traces collected from a VM in a production data centre. In this approach, a VM is given fixed size resources that match applications workload demands and any demand that exceeds the fixed resource allocation is predicted and handled through vertical VM auto-scaling. In this approach, energy consumption by PMs is reduced through efficient resource utilization. Experimental results obtained from a simulation on CloudSim Plus using GWA-T-13 Materna real backend traces shows that data center energy consumption can be reduced via efficient resource utilization

Keywords: Virtual machines, cloud computing, data centre energy consumption, virtual machine auto-scaling, CloudSim Plus

1. Introduction

1.1. Background

In the recent past, cloud computing has been indispensable in supporting computing needs in organizations. These organizations include individual cloud users, business corporations and educational entities [1]. The growth of cloud computing is because of its benefits such as, cost-saving, mobile access, flexibility and scalability and resource maximization, which traditional computing cannot offer. As a result, many cloud services provider (CSPs) such as Google, HP, Amazon, Facebook, IBM and Salesforce to small companies such as Linode, CloudSigma, Vultr and Digital Ocean are putting up many data centre to meet cloud computing demands [1]. Unfortunately, data centres consume a lot of energy, some of which is wasted. Currently, data centres consume about 3% of global electricity consumption and is expected to triple by the year 2020 [2]. The main cause of energy wastage in cloud data centres is the low level of server utilization [3]. Low server utilization is a consequent of allocating more resources than required to running applications. For instance, in Infrastructure as a Service (IaaS) public clouds, cloud service providers (CSPs) deliver

computing resources in the form of virtual machines (VMs) templates, which users have to choose from. For inexperienced cloud users, who have little knowledge of how much resources are required by their applications, it is difficult to make an optimal decision [4]. More often, the non-expert users overprovision resources, which go to waste despite consuming energy. There are various industry standards, which guide on how resources are supposed to be consumed such as Data Centre Maturity Model (DCMM) % [5] and VMware Knowledge Base (VMware KB) [6]. DCMM's visionary level holds that the monthly average CPU utilization should be above 60%. On the other hand, VMware KB holds that 80% CPU utilization and 85% memory are considered a *ceiling* or a *warning* if CPU utilization is 90% for 5 minutes and memory utilization is 95% for 10 minutes. The thresholds set by such references can be used for VM auto-scaling.

To determine the amount of resources, which need to be allocated to a given VM, VM's resources have to be monitored and analyzed [7]. The insights from the analysis can then be used to propose VM sizes. A number of big CSPs offering public cloud have services that are used to give insights to cloud users about their VM's resources usage. For instance, Google cloud provides a service for applying right-sizing to VM instances [8]. Microsoft Azure [9] [10] and Amazon cloud have similar services [11]. Unfortunately, these services are proprietary. Furthermore, the services offered do not customize the VMs automatically based on resources usage insights – cloud users have to access the VM resource usage insights and then take actions manually. This does not take full advantage of the already mature hypervisors, such as Xen and VMWare, which can complete such recommendations automatically. This can be accomplished through auto-scaling, which is done after VM resource usage forecasting [12]. Ordinarily, data centre backend traces are collected over time and thus regarded as a time series data and resources prediction methods such as Autoregressive Integrated Moving Average (ARIMA) and artificial neural network (ANN) can be used to aid in auto-scaling. To address the problem of low server utilization, current approaches have concentrated on VM allocation and migration [13] [14] [15]. These approaches only achieve server/host level optimization and not VM level optimization [16].

1.2. VM Auto-scaling

In [17], the authors have observed that IaaS's model of provisioning fixed VM sizes, which are referred to as VM instances is likely to change to varying VM sizes and is economically driven. The reason is that the fixed VM instances force cloud users to provision VM resources using peak resources demands for a time-varying resources demand. This is where auto-scaling is needed. Auto-scaling gives cloud computing its elasticity property, which is the ability to allocate and release computing resources on demand. Auto-scaling can be accomplished in two ways – horizontal auto-scaling and vertical auto-scaling [12]. In the former, one or more replica VMs are added or removed to match the application demands. This approach is suited for Software as a Service (SaaS), where cloud users access just the application and do not own VMs. On the contrary, vertical auto-scaling allows adjusting a VM size on the fly. For instance, more virtual CPUs (vCPU) can be added to a VM to increase the speed of application processing. Modern hypervisors such as Xen and VMWare support automatic CPU and memory scaling. With memory, scaling is accomplished via memory ballooning [18]. On the other hand, CPU scaling can be accomplished in two ways [12]. The first approach is by adding or removing vCPU via hot-plug. The second approach involves controlling the number of physical CPU cycles for a given VM during run time. In IaaS public, since VMs are directly owned by cloud users, vertical auto-scaling is the visible scaling approach.

An auto-scaling solution needs to adopt an auto-scaling process, which can be summarized as a MAPE (Monitoring Analyzing Planning Execution) loop [19]. Monitoring provides measurements and recording of resource usage over time. More often, monitoring gives rise to data centre backend traces. Analyzing examines the traces whereas the execution does the actual allocation of resources to

VMs. Analysis of trace logs can be done using many different methods such as those described by the work in [7] [20] and [21]. Before execution, the planning phase decides what action to take.

1.3. Cloud Backend Traces

According to [7], there three major sources of cloud workload traces, which can be used in cloud computing experiments - real workloads, synthetic workload and workloads obtained by using workload generators. Cloud backend traces fall in the category of real workloads because they are obtained from production data centres. They are the most preferred workload to be used in investigating cloud computing application behaviour. There are many real backend traces, which have been published and are publicly available. They include Facebook Hadoop workloads, Yahoo cluster traces [22], Google cluster trace (GCT) [20] [23] and Grid workload archive (GWA) [24]. GWA has published many datasets featuring different characteristics. For instance, the most recent datasets in studying VM resource usage are GWA-T-12 Bitbrains and GWA-T-13 Materna. Among the factors to consider before using a given dataset in experiments in the cloud is if its characteristics match research objectives. For example, to study VM resources usage, GWA's GWA-T-12 Bitbrains and GWA-T-13 Materna are the best. GWA-T-13 Materna, which has been used in this paper, is collected from the distributed Materna's data centre hosting a variety of highly critical business applications (e.g. SAP, government, IT, etc.) over a period of 1-month, three times – it has three sets of data collected from the same set of VMs [25]. The first set consists of 520 VMs, the second set consist of 527 VMs and the third one consists of 547 VMs. The data is organized into CSV files and shows resources allocated to the VM (such as CPU and memory) and resources used by application workloads.

1.4. Cloud Simulations

Testing cloud computing technologies can be an expensive affair if it has to be done on real hardware testbeds. Furthermore, cloud applications require timely, repeatable and controllable methodologies for evaluations before deployment, which cannot be guaranteed on real testbeds. Cloud simulators can be used to address this gap. CloudSim Plus is one such cloud simulators, which can be used to test the different characteristics of cloud computing [26]. CloudSim Plus is Java-based cloud simulator forked from CloudSim [27] except that it is easier to use because it follows software engineering standards with code duplication entirely removed. CloudSim Plus perfectly emulates a cloud data centre – it has the following components; a Cloudlet, VM, Broker, Host and Data centre. A cloudlet is similar to user applications, which are executed inside VMs. VMs are held in hosts, which are typically servers in a data centre. A data centre is comprised of hardware with physical computing resources and all the software that is used to manage the hardware. CloudSim Plus framework allows the creation of the aforementioned components in Java code and execution of workloads using various VM allocation algorithm of choice inbuilt in the simulator. The simulators can be used to test many different cloud characteristics including energy consumption.

1.5. Time Series Prediction Methods

Time series is a very common way of analyzing a sequence of data. Therefore, it is useful in predicting future resource demands in an entire data centre or individual VMs. There are many different models that can be used to predict future VM resource usage such as moving averages (MA), auto-regression (AR), auto-regressive integrated moving averages (ARMA) and artificial neural networks (ANN) [12]. ARIMA and ANN are commonly preferred because they can discover patterns in a time series to improve prediction accuracy. According to [28], if a time series follows a normal distribution, an ARIMA model is used, otherwise, ANN prediction model is used. A Jarque-Bera test can be applied to check for time series normality.

2. Related Works

In recent times, there has been a growth of literature related to virtual machine customization to support the need to move from fixed-size VMs to varying VM sizes. We have reviewed research works focusing on either VM customization or VM auto-scaling and VM resource usage prediction.

The work in [4] has proposed a VM sizing approach, which maps a group of tasks to customized VM types for container-based cloud applications. The mapping is based on task resource usage patterns, which are obtained from an analysis of historical resource utilization data extracted from a production cloud. The resources assigned to a VM should match the resources needs of the processed tasks leading to efficient utilization of cloud resources. As a consequence, energy consumption is decreased because fewer PMs are required. The proposed approach has been evaluated using GCT and authors report that if VM resources are allocated based on the discovered usage patterns, significant energy saving can be achieved.

In [29], the authors have proposed an approach, which involves VM sizing and VM placement. This approach is based on co-locating heterogeneous workloads in a similar server by creating copies of the same VM in different servers to reduce the aggregate demand of similar resources in the same server. The incoming load is then distributed to the copies of the VM thus allowing the CSP to achieve more aggressive consolidation without performance loss, which is normally caused by homogenous workloads. Apart from reducing energy consumption resulting from the use of fewer servers, the authors have reported improved reliability resulting from the existence of copies of similar VMs.

In [28], the authors have proposed a real-time time resource usage prediction system for IaaS cloud VMs. In this system, real-time resource usage collected in time intervals (time series) is fed into the proposed system and prediction is either accomplished using ARIMA or ANN. If the time series follows a Gaussian distribution, ARIMA is used, otherwise, ANN is used. Predicted resource values are them used to scale VMs resources. The authors have evaluated the proposed system using randomly selected VMs from GWA-T-12 Bitbrains real backend traces (mentioned earlier). Other research works, which have reported successful resource usage prediction include [30], [31] and [32].

In [33], the authors claim that the current auto-scaling approaches employed in container-based cloud applications use auto-scaling rules with static thresholds. An example of a static threshold in allocation resources based on a 95^{th} percentile or an average of historical resource demands. Furthermore, the rules rely only on infrastructure-related monitoring data such as memory and CPU usage. Based on this, a dynamic auto-scaling method has been proposed, which uses both infrastructure-related monitoring data and application-level monitoring data (such as response time or application throughput). The proposed method performs better when compared with seven different auto-scaling approaches.

The authors in [34] claim that using threshold-based auto-scaling approaches fail because it is difficult to set thresholds with the right values. To address this challenge, the authors have proposed a dynamic threshold approach, which predicts resources using Long Short-Term Memory Recurrent Neural Network and auto-scale virtual resources based on predicted values. The main problem addressed by the proposed approach is to handle Slashdot – a situation where auto-scaling might fail due to a sudden influx of traffic. The evaluations results show that the proposed algorithm outperforms existing algorithms.

3. Proposed Approach

Although many VM customization techniques have been proposed, most of them do not use mixed approaches - static threshold and dynamic threshold. In this paper, we propose a technique, which customizes VMs by assigning static resources (using static thresholds i.e. percentiles) based on historical CPU and memory usage. Because of the varying nature of resources demand, the demands, which exceed the fixed resources demands are predicted for VM auto-scaling. In this approach, energy consumption is reduced by reducing PMs via efficient resource utilization. Fig. 1 summarizes

the proposed approach. The target cloud is IaaS, where cloud users are forced to provision VM resources using peak resources demand for a time-varying resources demand and without the knowledge of the resources requirements of their applications.

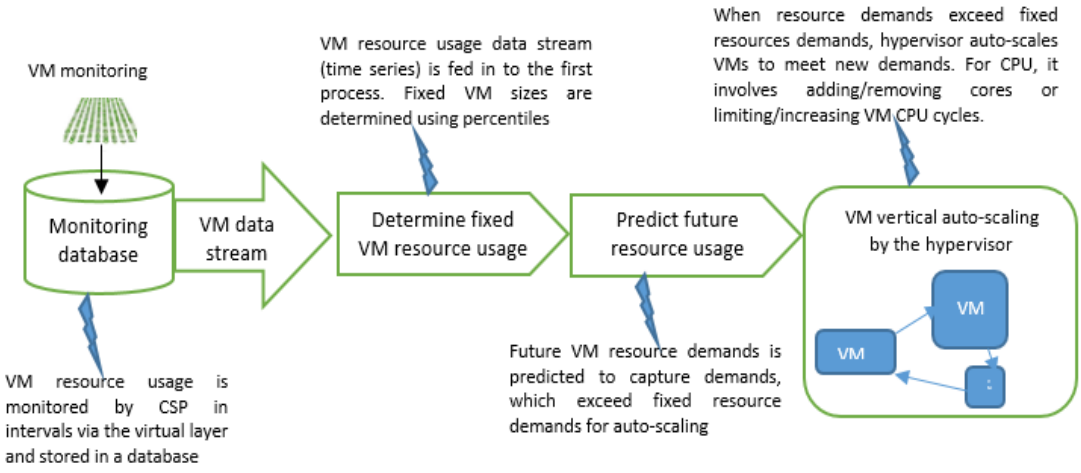

Fig. 1. Proposed approach methodology

The fixed VM resources are fixed so that 90% of sampled demands are covered (i.e. 90th percentile), then scaled such that resource consumption is at 80% (from VMware KB). For instance memory's percentile ranking, R_{memory}, is given as shown in **equation 1**.

$$R_{memory} = \left\lceil \frac{90}{100} * N \right\rceil, \qquad (1)$$

where N is the total sampled memory usage observations. A similar treatment is applied to CPU usage. If a function, *f(R),* returns the actual memory usage from memory ranking *R,* then fixed memory for the VM is given according to **equation 2**.

$$Memory = \frac{5}{4} f(R_{memory}), \qquad (2)$$

If the predicted resources usage values exceed the fixed resource usage, the hypervisor is triggered to auto-scale the VM vertically to appropriate values.

The dataset used in this paper is GWA-T-13 Materna, which consists of 520 VMs (discussed earlier). We have carried out some preliminary analysis on randomly selected VMs from the dataset such as comparing resources provisioned and resources actually used, time series normality testing and percentiles. Fig. 2 shows the amount of CPU used as monitored during the entire time. The time series plot shows that resources usage by VM's application workload is highly dynamic. Therefore, provisioning VM resources using peak resources leads to serious resource wastage.

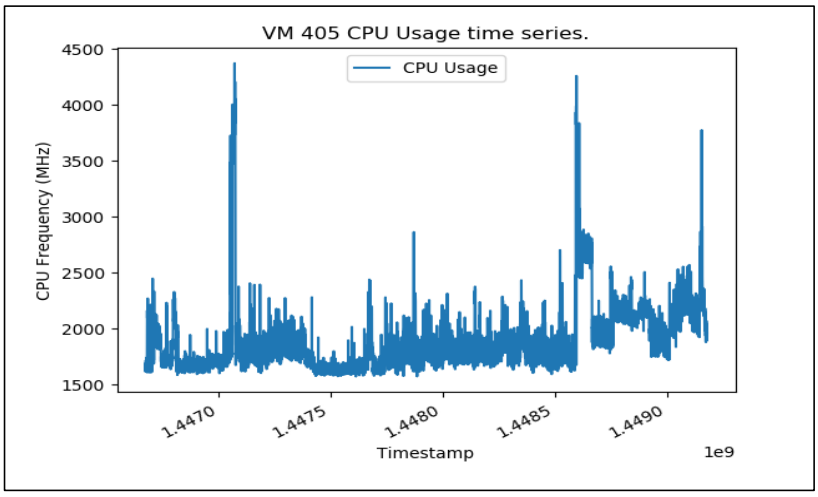

Fig. 2: CPU usage for VM 405 for the entire time

In Fig. 3, the plot shows a comparison between memory provisioned to VM and memory actually used to process workloads. It can be observed that resources actually used are very small as compared to resources provisioned. In fact, an analysis performed in all the 520 VMs, showed that the monthly average CPU usage is about 4.5% whereas memory usage has a monthly average of about 8.5%. This resource utilization is very low as compared to that recommended by VMware KB and DCMM industry standards.

Further, Fig. 4 shows the 90th percentile for memory consumption for VM 405. It can be observed that provisioning resources at 90th percentile would not be sufficient. This is the reason why predicting future resources demands is necessary. In addition, we have tested time series normality using Jarque-Bera test a number of randomly selected VMs and results show that the test variable has a non-Gaussian distribution of values. For instance, Table 1 shows the results of VM 172's memory usage normality test. As observed, the *p* value is less than *alpha* value, thus we conclude that our variable has no non-Gaussian distribution of values. Therefore, ANN is used to predict future resource values.

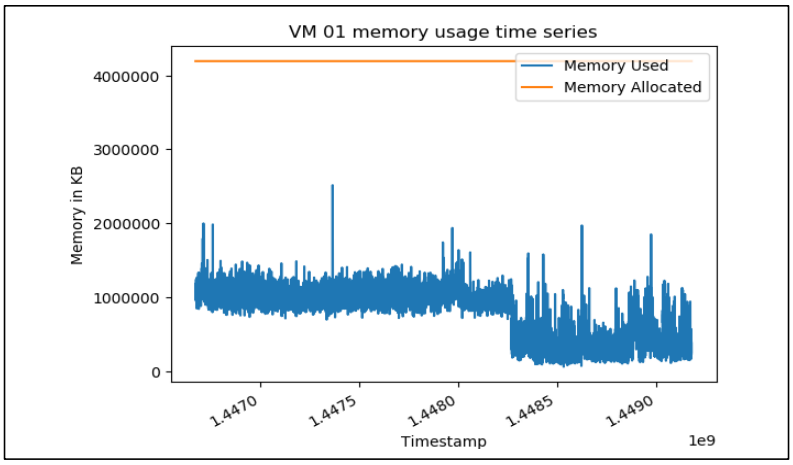

Fig. 3: A comparison between memory used and memory allocated for VM 1 for the entire time

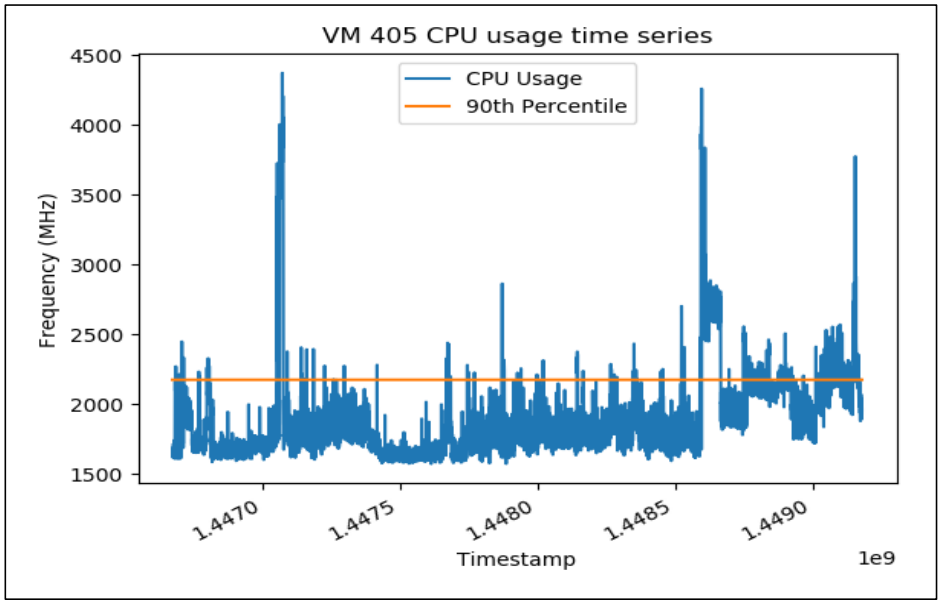

Fig. 4: VM 405's memory usage for the entire period showing 90th percentile

Table 1: Results of the Jarque-Bera test on VM 172's memory usage

Test metric	**Value**
JB	3802.958114
***p* value**	0
Alpha	0.05

The proposed ANN model is shown in **equation 3** and is represented in the architecture shown in Fig. 5.

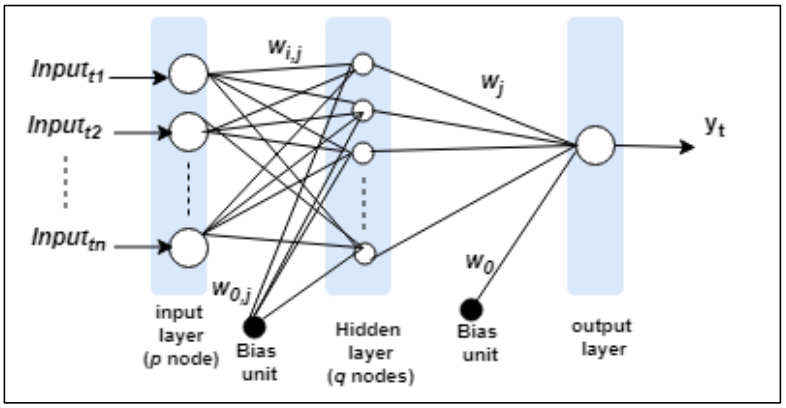

Fig. 5: NN diagram representing ANN model (equation 3)

The output y_t in the diagram is a predicted value at a time t, given that values in previous time are available. Thus, y_t is given according to **equation 3**.

$$y_t = w_0 + \sum_{j=1}^{q} w_j \,.\, g\left(w_{0,j} + \sum_{i=1}^{p} w_{ij} \,.\, Input_{t-i}\right) \qquad (3),$$

where p and q represent the number of inputs and hidden nodes respectively, $w_{ij}(i=1,2,...p, j=1,2,...q)$ and $w_j(j=0,1,2,...q)$ are connection weights ($w_{0,j}$ and w_0 are the biases applied to the input nodes and output of hidden nodes respectively) and $g(.)$ is a sigmoid activation function.

4. Experimental Evaluation

The main objective of this experimental setup is to evaluate the success of static VM sizing from an energy consumption perspective as well as the performance of the prediction of future resource usage. For static VM sizing, we execute application workloads in GWA-T-13 Materna trace 1 using a data centre configuration similar to the one that produced traces. The data centre is simulated on CloudSim Plus using FF, BF and WF VM allocation algorithm. These algorithms are readily implemented in CloudSim Plus. The data centre consists of 49 hosts, with 1298 CPU cores and 6780 GB of memory and runs 520 VMs. The resources allocated to each VM is contained in each VMs file in the dataset. The data centre uses VMware ESX hypervisor and the host's idle power is set at 60% of its peak power. The same application workload is executed after VM sizing. The amount of energy consumed during the execution before and after VM sizing is compared. Power, P_T, consumed by the data centre hosts is computed according to **equation 4**. The amount of resources (CPU and memory) allocated to VMs before and after VM sizing is also compared.

$$P_T = \sum_{i=1}^{n} ((P_i^p - P_i^b) * \left(\frac{N_i}{100}\right) + P_i^b, \qquad (4)$$

where n is the number of hosts in a data centre, P^p is the peak power consumption of the i^{th} host, P^b is the host's idle power and N is the percentage CPU utilization of the host. Energy, E, computed as shown in **equation 5**.

$$E = PT, \qquad (5)$$

where P is equivalent to P_T (measured in watts) and T is a time (in seconds) interval.

The ANN model has been done in the Waikato Environment for Knowledge Analysis (WEKA) using its GUI option. Selected VM's data is converted into WEKA's ARFF data format and loaded into WEKA. Feature selection is accomplished using correlations using CPU consumption and memory as target classes in turn. The data is then partitioned (with the order of observations preserved) into 80% for training and 20% for testing. The ANN model is trained using the parameters shown in Table 2. The predicted values of the model are tested using three accuracy metric - Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE).

Table 2: ANN model training parameters

Parameter	Value
Learning rate	0.3
Momentum	0.2
Training time/epoch	1000
No. of hidden layers	1

5. Results and Discussion

After fixed VM customization using the proposed technique, we have theoretically reduced the amount of resources provisioned to the VMs – from 1298 cores to 535 cores for CPU and from 6780 GB to 4142 GB of memory. A reduction in the amount of resources results in a reduction of the number of active data centre hosts. As a consequence, the amount of energy consumption in the data centre reduces. This claim is confirmed in Fig. 6, which shows a comparison of the energy consumption by hosts in the data centre before and after VM customization.

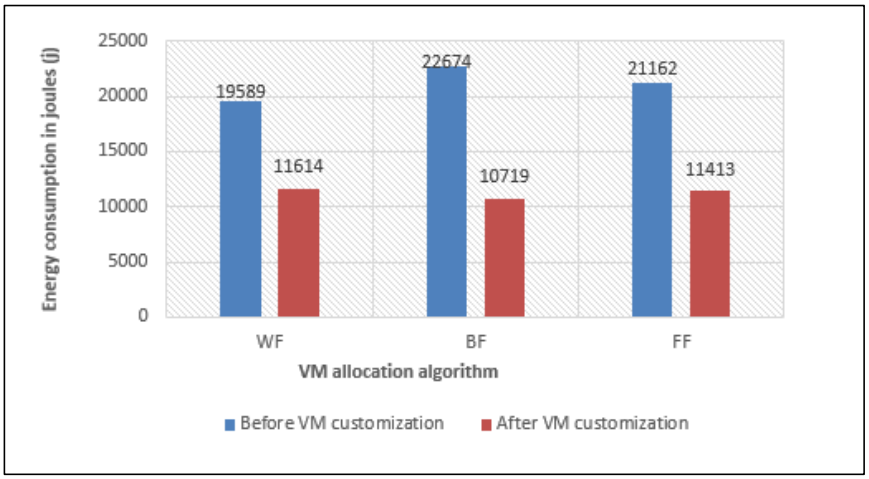

Fig. 6: A comparison between energy consumption before and after VM customization across different VM allocation algorithms

On ANN model prediction performance, 5 randomly selected VMs were evaluated on 3 accuracy metrics – MAE, MAPE and RMSE. Table 3 shows the performance metrics results on VM 172.

Table 3: VM 172's performance metrics results

Metric	Value	
	CPU	Memory
MAE	23.1	183012.5
MAPE	13.8	17.9
RMSE	109.4	219973.7

As observed from the table, there is a high prediction's accuracy. Generally, MAE is affected by the unit of measurement used and thus the MAE for memory, which is 183012.5 may be misleading. Fortunately, the memory unit is in kilobytes (KB) (183012.5 KB is approximately equal to 0.183 GB), which shows a small deviation of the predicted values from the actual values. A similar argument applies to RMSE. In this case then, we rely on MAPE, which again shows a small difference – 13.8% for CPU and 17.9% for memory. The highest MAPE value for the 5 VMs considered was about 30% for CPU and 24% for memory. The accuracy metrics values show the size of variation of the predicted values from the actual values – over or below. Generally, if the prediction is higher than the actual value, there is no worry because the VM will not suffer from lack of enough resources. On the contrary, predictions, which are lower than the actual values makes the VM suffer from lack of enough resources. In this case, we recommend combining this approach with statistical multiplexing - a technique where a VM borrows resources from a co-located VM at peak demands. We have shown in our earlier work, [35], that this technique is visible because VM resource demands do not peak simultaneously.

6. Conclusion

In this paper, we have proposed a mixed approach of VM customizing to match resources demand by application workloads – fixed VM size resources and VM auto-scaling via prediction. The fixed VM resources use a static threshold (90^{th} percentile) while auto-scaling uses ANN for prediction. We tested our approach using real backend traces from a production data centre and results show that using VM size customization can lead to energy savings in a data centre. As future work, we plan to implement our approach on Xen hypervisor and perform trials on real hardware.

Reference

1. P. Jemishkumar, I.-L. Y. Vasu, B. Farokh, Jindal, X. Jie and G. Peter, "Workload Estimation for Improving Resource Management Decisions in the Cloud.," in 2015 IEEE TwelfthInternational Symposium on Autonomous Decentralized Systems, 2015.
2. I. 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.
3. G. Chaima, "Energy efficient resource allocation in cloud computing Enviroment," Institut National des T´el´ecommunications, Paris, France , 2014.
4. 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 centre Resources," The Computer Journal, 2015.
5. 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.
6. VMware, "Performance Best Practices for VMware vSphere 6.0," VMware, Inc, Palo Alto, CA, 2015.
7. D. Kenga, V. Omwenga and P. Ogao, "Statistical Techniques for Characterizing Cloud Workloads: A Survey," in 4th Strathmore International Mathematics Conference, Nairobi, 2017.
8. 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].
9. Amazon, "ParkMyCloud Cost Optimization, Scheduler & Management Deprecated," Amazon, 2019. [Online]. Available: https://aws.amazon.com/marketplace/pp/B07K2L9YZW. [Accessed 10 January 2019].
10. Amazon Web Services, "Right Sizing: Provisioning Instances to Match Workloads: AWS Whitepaper," Amazon Web Services, Inc., 2018.
11. ParkMyCloud, "Why Azure Right Sizing is Important," ParkMyCloud, 2018. [Online]. Available: https://www.parkmycloud.com/azure-right-sizing/. [Accessed 01 November 2018].
12. L. Yazdanov, "TOWARDS AUTO-SCALING IN THE CLOUD: ONLINE RESOURCE ALLOCATION TECHNIQUES," Technische Universitat Braunschweig, 2016.
13. N. Krishnaveni and G. Sivakumar, "Survey on Dynamic Resource Allocation Strategy in Cloud Computing Environment," International Journal of Computer Applications Technology and Research, vol. 2, no. 6, pp. 731 - 737, 2013.
14. X. Zhanga, T. Wu, M. Chen, T. Wei, J. Zhou, S. Hu and R. Buyya, "Energy-aware virtual machine allocation for cloud with resource reservation," The Journal of Systems and Software, vol. 147, no. 2019, pp. 147-161, 2019.
15. S. Kaur and V. Pandey, "A Survey of Virtual Machine Migration Techniques in Cloud Computing," Computer Engineering and Intelligent Systems , vol. 6, no. 7, 2015.
16. F. P. Sareh, "Energy-Efficient Management of Resources in Enterprise and Container-based Clouds," The University of Melbourne, 2016.

17. O. A. Ben-Yehuda, M. Ben-Yehuda, A. Schuster and D. Tsafrir, "The rise of RaaS: the resource-as-a-service cloud," Communications of the ACM, vol. 57, no. 7, pp. 76-84, 2014.
18. A. X. Bronson, R. P. and S. S. Raja, "A Dynamic Memory Allocation Strategy for Virtual Machines in Cloud Platform," International Journal of Pure and Applied Mathematics, vol. 119, no. 15, pp. 1423-1444, 2018.
19. V. Emeakaroha, M. Netto, R. Calheiro, I. Brandic, R. Buyya and C. Rose, "Towards autonomic detection of SLA violations in Cloud infrastructures," Future Generation Computer Systems, vol. 28, no. 7, pp. 1017-1029, 2012.
20. M. Alam, A. S. Kashish and S. Shuchi, "Analysis and Clustering of Workload in Google Cluster Trace Based on Resource Usage," in 2016 IEEE Intl Conference on Computational Science and Engineering (CSE) and IEEE Intl Conference on Embedded and Ubiquitous Computing (EUC) and 15th Intl Symposium on Distributed Computing and Applications for Business Engineering (DCABES), Paris, France, 2016.
21. S. Shen, V. v. Beek and A. Iosup, "Statistical Characterization of Business-Critical Workloads Hosted in Cloud Data centres," in 2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, Shenzhen, China, 2015.
22. Yahoo, "Computing Systems Data," Yahoo, 2019. [Online]. Available: https://webscope.sandbox.yahoo.com/catalog.php?datatype=s&guccounter=1&guce_referrer=aHR0cHM6Ly93ZWJzY29wZS5zYW5kYm94LnlhaG9vLmNvbS8&guce_referrer_sig=AQAAAC_THkiHAd-3c25yQ-faDODXLIKkWaUwVtxotuRxLvHDGn3mxbcWBQm9XEiH9rMjByu7Cfs-KbZ1p5JqKI1tK9rC0c5PTiiKaVRz. [Accessed 12 January 2019].
23. C. Reiss and J. Wilkes, "Google cluster-usage traces: format + schema," Google , 2011.
24. Delft University of Technology, "The Grid Workloads Archive," Delft University of Technology, 2019. [Online]. Available: http://gwa.ewi.tudelft.nl/. [Accessed January 10 2019].
25. Delf University of Technology, "GWA-T13-materna-trace," Delf University of Technology, 2018. [Online]. Available: http://gwa.ewi.tudelft.nl/datasets/gwa-t-13-materna. [Accessed 23 November 2018].
26. F. Manoel, R. Oliveira, C. Monteiro, P. Inácio and M. Freire, "CloudSim Plus: A cloud computing simulation framework pursuing software engineering principles for improved modularity, extensibility and correctness," in 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), Lisbon, Portugal, 2017.
27. C. Rodrigo, R. Rajiv, B. Anton, D. R. Cesar and B. Rajkumar, "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms," Journal of Software: Practise and Experience , vol. 4, no. 1, pp. 23-50, 2011.
28. 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: Hindawi, vol. 2017, no. 4873459, 2017.
29. G. Hadi and P. Massoud, "Achieving Energy Efficiency in Data centres by Virtual Machine Sizing, Replication, and Placement," in Energy Efficiency in Data centres and Clouds, Elsevier Science, 2016.
30. 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.
31. 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.
32. 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.
33. S. Taherizadeh and V. Stankovski, "Dynamic Multi-level Auto-scaling Rules for Containerized Applications," The Computer Journal, vol. 62, no. 2, p. 174–197, 2019.

34. A. Shahin, "Automatic Cloud Resource Scaling Algorithm based on Long Short-Term Memory Recurrent Neural Network," International Journal of Advanced Computer Science and Applications, vol. 7, no. 12, pp. 279-285, 2016.
35. K. Derdus, V. Omwenga and P. Ogao, "Virtual Machine Sizing in Virtualized Public Cloud Data Centres," International Journal of Scientific Research in Computer Science, Engineering and Information Technology, vol. 5, no. 4, 2019.