HIGH SCHOOL EDITION
Journal of Student Research
Volume 11 Issue 3 (2022)

Using Virtual Machine Size Recommendation Algorithms to Reduce Cloud Cost

Kenhao Chin[1], Justin Zhao[1], Eric Shan[1] and Ziliang Zong[#]

[1]Seven Lakes High School, USA
[#]Advisor

ABSTRACT

Cloud spending has risen on a year-to-year basis, with the pandemic acting as the primary catalyst for its recent growth; however, "cloud waste," referring to cloud resources that are not used to their full capacity, also follows this upward trend and causes the loss of an increasingly large amount of money. Unfortunately, present-day cloud research lacks data-driven studies that analyze why cloud users are wasting resources, or suggestions to users on how to lessen such waste. In order to prevent this over-expenditure, it is vital to choose the best-suited options when it comes to virtual machines (VM), especially for small to mid-sized businesses with limited funds and a lack of expertise. In this paper, we first analyzed the 235 GB Azure user dataset from the users' perspective. We then implemented machine learning to determine our pricing model and the VM costs. With these statistics, we then delineated our methodology to calculate the wasted cost of each VM, and using this data, we propose an algorithm that can identify potential candidates with wasteful VMs and assist users in reducing costs. By applying our algorithm to approximately 2.7 million VMs, we demonstrate that it has the ability to help 66,721 VMs created by 1,520 users lower their monthly costs by $14.9 million. We conclude that businesses, while still reaping the benefits of cloud services, can do so at a much lighter cost and save on their VMs.

Introduction

As cloud computing technology grows more popular, its spending also accelerates annually; in a 2021 report conducted by Flexera involving 753 respondents, which include large enterprises (organizations with 10,000 or more employees) as well as small to mid-sized businesses (organizations with fewer than 1,000 employees), 37 percent of enterprises are found to spend more than 12 million that year. For small businesses, 53 percent spent more than 1.2 million, an increase from 38 percent last year. Gartner furthers this by measuring cloud computing's global growth from 2020 to 2021: worldwide end-user spending was forecasted to increase by 18.4%, from $257.5 billion to a new total of $304.9 billion.

However, the rapid expansion in cloud expenditure is closely linked to a paralleled surge in cloud storage waste and overspending. For example, the Flexera report indicates that an estimated 32 percent of cloud resources were wasted according to users in 2021, compared to 30 percent the previous year. As cloud waste increases, the monetary waste accumulated from running these applications rises as well. For smaller businesses especially, funding is limited, and it is to the users' advantage that they are able to perform the same work with a more efficient solution; in other words, reducing expenditure would directly improve organizational profit for companies using cloud computing technology.

Cloud computing services are often charged on a pay-as-you-go basis, allowing the enterprises to control and raise their resources when needed. Essentially, cloud computing is beneficial to business owners in the way that they are able to plan for provisions. However, users without an IT team may not be wholly knowledgeable or aware of its most efficient applications, thus leading them to overestimate their storage needs and, in turn, overspend. While prior studies have succeeded in determining the costs and waste of using cloud computing services, no previous work has determined data-driven reasons for cloud waste, or suggested an algorithm to help users in reducing the waste. After analyzing virtual machine workloads, the two common reasons as to why waste occurs consist of the following two factors: resources that are paid despite being unused (idle resources), and resources that are larger in capacity than needed (over-provisioned resources).

Online price calculators for major cloud service providers such as Microsoft Azure, Amazon Web Services (AWS), and Google Cloud Platform (GCP) provide a diverse selection of virtual machines to its users. Without having proper resource management and a thorough understanding of their workload demands, users will be likely to overspend and purchase a virtual machine with excess resources. While the issue of idle resources may not be preventable in some cases (i.e. if a company only requires the use of a virtual machine for a given amount of time), the second factor, overprovisioned resources, can be solved with downsizing in size. We argue that if a user is able to complete their workload with a smaller capacity, they should reduce their virtual machine's size, allowing for more effective and practical use of both the machine and the spending that would be otherwise "wasted."

In this paper, we first analyzed a characterization of Azure's virtual machine workload, which includes the virtual machine's size, lifetime, deployment size, and usage rates. From this user dataset, we observed that there were groupings of inefficient and efficient users, meaning that while some users were able to use their virtual machines to their fullest capacity, the majority of users did not do so. Next, we constructed our pricing model using the online Azure price calculator and used a linear regression model, to fill in missing cost data points for CPU and memory storage options. For this purpose, we used Java programming to calculate the cost of each individual virtual machine using the lifetime and virtual machine size data.

Our evaluation of the waste for each virtual machine begins with our methodology considering the costs that were calculated with the usage rates. Once the waste for the virtual machines was found, we used a waterfall-based recommendation system to help users regulate storage sizes. Based on the number of virtual machines created by a single user, the 95th percentile latency of usage rates, and the wasted cost, users could downsize their virtual machines in both memory and CPU to have a more cost-effective utilization of cloud computing resources.

Results indicate that by reducing CPU and memory sizes, we can achieve significant cost savings for users that were previously unable to make full use of their virtual machines. To quantify these savings, we implemented our recommendation algorithm — setting specific parameters in order to maximize its effect while also making an effort not to notify too many users — onto the Azure user dataset. Our findings show that we were able to help 66,721 virtual machines created by a total of 1,520 users lower their monthly costs by 14,988,203.34 USD.

Related Work

The need for a recommendation algorithm is built on the foundations of prior work, which show that despite the massive growth of the cloud computing scene, there is an increasingly large amount of cloud waste to match this rise in cloud expenditure. As spending rises and the percentage of that spending being cloud waste also rises, the money being wasted on unused cloud resources accelerates. The expansion of cloud computing services is best depicted through the studies conducted by both Flexera and Gartner (Adler 2022) (Costello and Rimol 2020). Flexera also includes the wasted cloud spend: 32%, which is a 2% increase from the preceding year (Adler 2020).

For our big data analysis, the characterization of workload behaviors used in our analysis of cost and the wasted cost was introduced in the work done by Eli Cortez, et al. (Cortez, et al. 2017). Several pieces of data about each virtual machine — including its lifetime, utility rates, CPU sizes, and memory sizes —

were provided. However, the costs that each user had spent on each virtual machine were not given; in order to calculate the costs, our pricing model (that was created using linear regression) relied on the lifetime and sizing data.

There was an inherent lack of previous studies on several topics, such as why cloud waste exists and recommendation algorithms to alleviate waste. Some sources conducted reviews of existing pricing models (Soni and Hasan 2017). Still, the closest prior work to our purposes constructed a new pricing model and proposed three different approaches to calculating the waste cost: uniform distribution, linear inverse distribution, and proportional inverse distribution. Out of the three, linear inverse distribution was selected out of a consideration of complexity and accuracy; it was found to be the most efficient and consistently accurate method of calculating waste (Vogel 2019). In contrast, our waste cost was determined using a methodology based on the cost and average utility rates.

The literature regarding recommendation algorithms is extensive, especially with regard to social and entertainment platforms like Netflix and Youtube (Airoldi, Beraldo, and Gandini 2016) (Varela and Kaun 2019). These algorithms are personalized and tailored to personal accounts, ultimately created for the purpose of limiting searches and nudging users toward suggested content. Youtube Music, specifically, produces groupings that are interpreted as crowd-generated music categories, which are shared by a community of listeners. By using “comparable situational frames,” Youtube is able to create stylistic congruence (Airoldi, Beraldo, and Gandini 2016). These sources analyze how recommendation algorithms work to accommodate individual user preferences, and we extend the same concept (of having customized suggestions) over to our recommendation algorithm, which employs a waterfall approach to adapt to the utility rates of different users. While a user with a higher utility rate would not be required to reduce the CPU and memory sizes of their virtual machine, another user with a lower utility rate may downsize to a lower tier.

Azure Workload Analysis

Microsoft Azure Dataset

We used the Microsoft Azure Public Dataset V2, which was recorded from 2,695,548 VMs in 2019. It contains 235GB of VM usage readings across 30 days, recording the CPU usage of each VM every 5-minutes. As a result, many VMs with less than 5 minutes of usage appear to run for 0 minutes. At most, these VMs will cost a few cents and cannot be optimized with our recommendation system, so we have omitted them from our data.

The dataset provides important information such as the VM ID, the ID of the user, and the subscription ID. It also provides the start and stop times of the VM, in seconds, with 0 starting at the beginning of the month. It contains the maximum CPU usage reached, the average CPU usage, and the 95th percentile of the max utilization, which means that it is higher than 95% of the utilization readings in that VM. It contains the VM category, which is separated into Delay-insensitive, Interactive, and Unknown. Interactive VMs run when a user is awake and using it, while delay-insensitive VMs run regardless of the time. Finally, the last two columns are the numbers of CPU cores and GBs of memory. Table 1 demonstrates the first 9 VM readings provided by the file.

Table 1. First 9 VM readings from the dataset.

Machine ID	User ID	Subscription ID	Start-time	Stop-time	Max CPU	Avg. CPU	P95 CPU	Category	Cores	GB Memory
71fJw0x+...	GB6uQ...	2sh/Zj...	558300	1673700	91.77689	0.72887	20.75962	Delay-insensitive	8	32
rKggHO/...	ub4ty8y...	+ZraID...	424500	425400	37.87926	3.32535	37.87926	Unknown	4	32

HIGH SCHOOL EDITION
Journal of Student Research

YrR8gPt...	9LrdYR...	GEyIE...	1133100	1133700	0.30436	0.22055	0.30436	Unknown	4	32
xzQ++JF...	0XnZZ8...	7aCQS...	0	2591400	98.57342	30.34005	98.21250	Interactive	2	4
vZEivnh...	HUGaZ...	/s/D5V...	228300	22980 0	82.58144	13.87629	82.58144	Unknown	2	4
MqvcZ...	p14cXG...	ZFCk8...	1395600	1397700	0.09787	0.03521	0.09787	Unknown	4	32
034PavX...	L9utvn...	k2nh5l3...	1422300	1422600	0.07127	0.03270	0.071277	Unknown	4	32
fBpt5H...	IwABY...	uYvK2...	2414400	2414700	0.25996	0.07230	0.25996	Unknown	24	64
ZdSiRJ...	5tTf4IJ...	vLlC4aS...	165900	16830 0	0.0985	0.03419	0.09854	Unknown	4	32

Data Analysis Tools

Orange (Demsar, et al. 2013) is a data visualization and machine learning tool that we used to create graphs, organize, and analyze our data set. For example, we used Orange's Linear Regression function to predict missing prices of different VM sizes. We also used Java to code our recommendation algorithm, due to it being a fast and efficient programming language.

Azure Workload Analysis

As pointed out by the Flexera report, cloud waste is estimated to be 32%. This hypothesis (that some cloud users do waste resources) can be confirmed by analyzing the Azure dataset.

From examining utilization percentages from the entire dataset using Orange, we found that the majority of users do not effectively utilize the purchased cloud resources, as shown in Fig. 1. In fact, a shocking 72.44% of users have an average utilization of below 20%. The reason for such inefficient VM usage is due to inexperienced users who spend heavily on a small number of powerful VMs. For example, the median VM in our data, with an average utilization of 8.196% would be wasting over 91% of processing power and cost. An experienced user, on the other hand, would instead spread their spending amongst many, smaller VMs that are utilized to near max capacity, and can add more if needed.

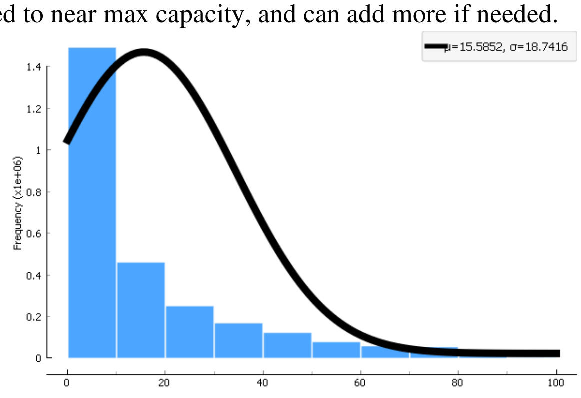
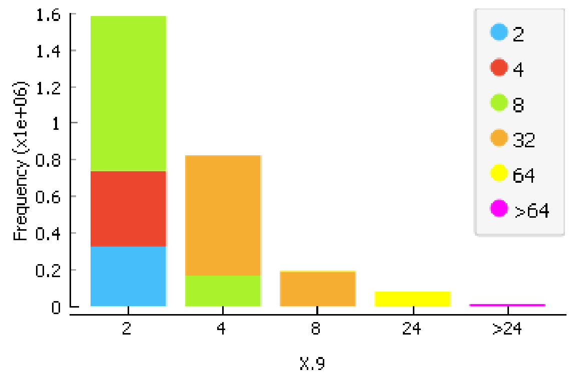


Figure 2. This figure shows the distribution of the VMs by average CPU utilization. Each bucket is the number of VMs in 10% of CPU utilization.

Also using Orange, we can see that 58.86% of VMs in the data have 2 cores, while 30.56% have 4 cores. VMs with 2 cores and 8 gigabytes of memory are the most popular VM size, and account for 31.43% of VMs in the data. Through this distribution graph, it is shown how experienced users use a massive amount



of 2 and 4 core VMs.

Figure 1. This figure shows the VMs distributed by their core count, further split by their GB of memory.

Price Model and Cost of VMs

Our previous analysis confirms the existence of cloud usage waste, as evidenced by the low utilization. To find the exact amount, one needs pricing information of the different VM sizes.

However, the original Azure data set does not provide any cost-related information. In order to analyze the efficiency of user spending, we need to construct a pricing model, as well as calculate the estimated wasted resources. We used the prices given by the Azure price calculator (standard Linux as the operating system) and West US as the region. Due to the Azure price calculator not including some of the VM sizes in our data, we used machine learning to create a price model for all VM sizes. This was achieved by using linear regression to calculate the coefficients a, b and c in the following equation. The first two stand for the increase in price for every additional core and GB of memory, and c stands for the intercept when a and b are zero.

$$a(corecount) + b(memory) + c = \gamma$$

The official Microsoft Azure price calculator was used to find x, y, and γ, which are the core count, GBs of memory, and cost per hour. In the finalized equation, a was 0.006 and b was 0.024. The total cost was calculated by multiplying γ by the time the VM was used.

Cloud Waste

Next, we estimated the amount of spending that the VM was wasting using the unused CPU processing power of the VM. This was achieved by multiplying total spending (γ) and (1 - U), with U being the average utilization as a percentage. We also used core hours, calculated by multiplying core count and run time (t), as a unit of measurement for the processing power of a VM.

$$W = (1 - U) * \gamma$$

This is one of the parameters we used to identify cloud waste, as a high amount of wasted spending is an obvious sign that a smaller VM could provide the same amount of processing power at a lower cost. As shown in Table II, a larger 8 core and 32 GB memory machine provides the same amount of core hours as a smaller 4 core and 8 GB memory VM, but has double the cost due to a low average utilization.

Table 2. These two VMS have very similar core hours, but the larger VM costs almost twice as much.

Total Cost	Core Hours	Cores	GB Memory
120.428	1180.57	8	32
62.424	1156	4	8

Recommendation Algorithm

Algorithm Parameters Selection

Our algorithm is made based on important information from the data analysis on VMs. In the end, our algorithm identifies the virtual machines as wasteful if the user had at least 25 virtual machines created, the p95 of processing utilization was under 75, and the wasted cost was over 75 US dollars. The minimum threshold was 25 virtual machines in order to ensure the algorithm focused on users that had enough VMs to be impacted by the cost of the cloud. Next, we determined the p95 of processing utilization under 75 because we wanted the users to be utilizing cloud services to a substantial degree. Finally, we determined that the wasted cost had to be over 75 US dollars because we did not want to give users warnings that had minimal impact on cost. Overall, our parameters ensure the algorithm targets users that create a substantial amount of cost wasted, in order to create better utilization of the cloud.

Augmented dataset for the Algorithm

The data given by Microsoft Azure's study was not enough to conduct cost-wasted analysis, so we added more data on top of the data given by Azure. First, we changed the time data from start to end to total hours run. Next, we calculated the cost by using machine learning, based on the data we got from Microsoft's pricing calculator. After, we used the formula [(1-p95 utilization)*cost] to determine the wasted cost.

Table 3. Augmented Data (Shows 9 rows of example input)

Machine ID	User ID	Subscription ID	Time (hrs)	Max CPU	Avg. CPU	P95 CPU	Category	Cores	GB Memory	Wasted Cost
71fJw0x+...	GB6uQ...	2sh/Zj...	309.833	91.77689	0.72887	20.75962	Delay-insensitive	8	32	236.218
rKggHO/...	ub4ty8y...	+ZraID...	0.25	37.87926	3.32535	37.87926	Unknown	4	32	0.185615
YrR8gPt...	9LrdYR...	GEyIE...	0.166667	0.30436	0.22055	0.30436	Unknown	4	32	0.127718
xzQ++JF...	0XnZZ8...	7aCQS...	719.833	98.57342	30.34005	98.21250	Interactive	2	4	48.1378
vZEivnh...	HUGaZ...	/s/D5V...	0.416667	82.58144	13.87629	82.58144	Unknown	2	4	0.0344495
MqvcZ...	p14cXG...	ZFCk8...	0.583333	0.09787	0.03521	0.09787	Unknown	4	32	0.447842
034PavX...	L9utvn...	k2nh5l3...	0.0833333	0.07127	0.03270	0.071277	Unknown	4	32	0.0639791
fBpt5H...	IwABY...	uYvK2...	0.0833333	0.25996	0.07230	0.25996	Unknown	24	64	0.127907
ZdSiRJ...	5tTf4IJ...	vLlC4aS...	0.666667	0.0985	0.03419	0.09854	Unknown	4	32	0.511825

Algorithm Input

The columns represent from left to right: MachineID, User ID, Subscription ID, Time Run(Hours), Max CPU, Avg CPU, p95 CPU, Instance, Core, Memory, Wasted Cost. See Table 3.

Algorithm 1 Our Recommendation Algorithm

1: Given: VMTable2 consisting of VM V UserID that created it, time VM ran, p95,cores, memory, Cost Wasted

```
function FINDTARGET(V)
    if UserIDcnt <= 25, p95 <75, wastedcost >75 then
        WastedList.add(V)
    end if
     return WastedList
end function
function DROPVM(WastedList)
    for each V in WastedList
    if p95 <25 then
        Drop VM by 3 levels
    end if
    else
    if p95 <50 then
        Drop VM by 2 levels
    end if
    else
    Drop VM by 1 level
     return RecommendationList
end function
```

Algorithm Explanation

There are 2 important steps in our algorithm. First, the algorithm goes through the function FINDTARGET(V) in order to sort virtual machines from the given data into a list of wasted virtual machines. Finally, the list is put through a drop recommendation algorithm (DROPVM(WastedList))to decide how many levels each virtual machine can drop to.

Waterfall Model

Our algorithm classifies the drop recommendation model in 3 levels: 25, 50, and 75. We chose these 3 as identifiers for how much we should drop because the utilization rate for each virtual machine is maximized if the user requests less computing power (resulting in cost efficiency. If the p95 is less than 25, the virtual machine will be recommended to drop 3 levels. Our recommended drop will reach 1 if the machine is less than 75 but greater than 50. Our level drop is defined as dropping the core and memory to the next available tier of computing power. For example, a user with 4 cores and 4 gigabytes of memory that is dropping one level would be dropped to 2 cores and 2 gigabytes of memory. Furthermore, if the next less computing virtual machine does not exist, it will be kept at 2 cores and 2 gigabytes of memory. However, this model assumes that the drop in memory will still allow users to run their needed functions in the cloud. The memory of each procedure is not given by the user, so this is assumed for our algorithm.

Algorithm Output

At the end of our algorithm, we send users a table of the identified wasted virtual machines along with the original given data, the new core and memory, and the total saved cost. In the end of the experimental data from the Azure table, we helped 1520 users save a total of 14,988,203.34 US dollars. See Table IV.

Table 4. Output Table (shows 5 rows of example output)

Machine ID	User ID	Subscription ID	Time (hrs)	Max CPU	Avg. CPU	P95 CPU	Category	Cores	GB Memory	Original Cost	Wasted Cost	New Cores	New GB Memory	Saved Cost
71fJw0x+...	GB6uQ...	2sh/Zj...	309.833	91.77689	0.72887	20.75962	Delay-insensitive	8	32	237.952	236.218	2	2	222.708
vlb8POlnY…	iGfBdg-MCx...	t9UC3N+/G...	719.833	98.433	3.4993	57.9796	Delay-insensitive	4	8	138.208	133.372	2	4	68.2402
BcGvAPhri...	6d+esuBiA...	D0Q3SRMPs...	719.833	73.8714	2.14896	25.4091	Interactive	4	32	552.832	540.952	2	4	482.864
geYEP2ZIk...	izi-VPtq8b...	QhuPipO0t...	250.417	67.9572	1.93893	8.12882	Delay-insensitive	4	32	192.32	188.591	2	2	179.999
uxD7b2CFh...	cLrGN/bi4...	vWzq77yoJ...	123.25	84.1843	18.5928	39.0993	Unknown	4	32	94.656	77.0568	2	4	82.6761

Experimental Results

Experimenting with Parameters

In order to find the correct parameters for our identification of the wasted machines, we experimented with different parameters in order to see how many virtual machines would be affected, how many users would be affected, and the total cost that would be reduced by our waterfall drop algorithm. Specifically, we tested the parameters of virtual machines created by the user and the wasted cost. We tested each parameter individually in order to get the most accurate perception of the effect of each parameter.

Parameter Cost Wasted Analysis			
Cost Wasted	VM's Affected	Total Cost Saved	Users Affected
100	70226	1.73E+07	3867
75	76923	1.78E+07	3951
25	224272	2.03E+07	5541

Parameter VM Machines Created Analysis			
User VM Count	VM's Affected	Total Cost Saved	Users Affected
100	2612411	1.78E+07	961
75	2617720	1.84E+07	1023

50	2628025	1.92E+07	1199
25	2662110	2.11E+07	2234

Analysis of Experimental Results

Through the parameter cost-wasted analysis, we found that the best fit was 75 because it had the right balance of total cost saved and users affected. This way our algorithm would be able to save a significant amount of money without affecting an excessive number of users. On the other hand, through the parameter VM Machines Created Analysis, we found that if the user created at least 25 virtual machines, then the total amount saved would be maximized. By using the "at least 25 VM" parameter, our algorithm would not incorrectly identify users that are using the cloud minimally or simply testing the cloud. With these two identified parameters, our algorithm was able to create a balance between warning too many users and maximizing cost efficiency.

Conclusion

In this paper, we provide insight into the spending and \waste costs of cloud computing, from the perspective of cloud consumers. We discussed the linear regression model used for producing our price prediction equation. Using this pricing model, we constructed our methodology for computing each virtual machine's wasted cost. Finally, we discussed the implementation of our recommendation algorithm; this included both the parameters, as well as the waterfall model of the algorithm itself. Further configurational nuances are presented as well, reinforcing the optimal parameters for our algorithm. The effectiveness of our proposed solution is tested with the application of the algorithm onto the Azure user dataset. Out of the 2,695,748 virtual machines in question, we demonstrate our ability to save 14,988,203.34 USD across a total of 66,721virtual machines created by 1520 users. This finding concludes that by reducing the CPU and memory sizes of virtual machines with low utility rates, users are able to save significantly on cloud computing resources while still being able to meet workload demands.

However, this study is based upon several key assumptions. First, while calculating the prices for the virtual machines, we assumed that the region in which these virtual machines were created was West US, the operating system was Windows, the type of virtual machine was operating system only, the tier was standard, and the instance type was Dsv5-series. While the first few choices were interchangeable with other options available on Azure's pricing calculator, the last assumption is made upon the fact that the Dsv5-series instance has no temporary storage provided; this suits the virtual machines in the user dataset better for the purposes of determining the costs. Another assumption was with regard to VMs in the dataset that had more than 64 GB of memory or 24 virtual cores; for these, we assumed that they had 128 GB and 32 virtual cores, respectively. We also assumed that the costs for these virtual machines were as shown in our pricing model, which was created using a linear regression model. Finally, in calculating the effects of our algorithm, we assumed that the users would always accept our recommendations for downsizing CPU and memory sizes.

Acknowledgment

We would like to thank our parents for supporting us throughout the research process. We would also like to give credit to the Microsoft Azure cloud public dataset that allowed us to conduct our research. Finally, we would like to thank our mentor for guiding us throughout this research.

References

[1] Adler, B,“Cloud Computing Trends: Flexera 2022 State of the Cloud Report,” March 2022.

[2] Airoldi, M., Beraldo, D., and Gandini A, “Follow the algorithm: An exploratory investigation of music on YouTube,” in Poetics, vol. LVII, 2016, pp.1–13.

[3] Chapel, J, “The Cloud Is Booming — But so Is Cloud Waste,” 2020.

[4] Cortez, E., Bonde, A., Muzio, A., Russinovich, M., Fontoura, M., and Bianchini, R, “Resource Central: Understanding and Predicting Work- loads for Improved Resource Management in Large Cloud Platforms,” 2017.

[5] Costello, K. and Rimol, M. G, “Forecasts Worldwide Public Cloud End- User Spending to Grow 18% in 2021,” 2020.

[6] Gunawardana, A. and Shani, G, “A Survey of Accuracy Evaluation Met- rics of Recommendation Tasks,” Journal of Machine Learning Research, vol. X, December 2009.

[7] Hadary, O., Marshall, L., Menache, I., Pan, A., Greeff, E. E., Dion, D., Dorminey, S., Joshi, S., Chen, Y., Russinovich, M., and Moscibroda, T, “Protean: VM Allocation Service at Scale,” Microsoft Azure and Microsoft Research, November 2020.

[8] Reddy, V. K., Rao, B. T. , and Reddy, L. S. S., “Research Issues in Cloud Computing,” Global Journal of Computer Science and Technology, July 2011.

[9] Soni, A. and Hasan, M, “Pricing Schemes in Cloud Computing: a Review,” International Journal of Advanced Computer Research, vol. VII, February 2017.

[10] Varela, D., and Kaun, A, “The Netflix Experience: A User-Focused Approach to the Netflix Recommendation Algorithm,” New York: Peter Lang Publishing Group, 2019, pp. 197–211.

[11] Vogel, P, “Computing the Cost and Waste of Cloud Computing,” August 2019.

[12] Demsar, J., Curk, T., Erjavec, A., Gorup, C., Hocevar, T., Milutinovic, M., Mozina, M., Polajnar, M., Toplak, M., Staric, A., et al., ``Orange: Data Mining Toolbox in Python," Journal of Machine Learning Research, vol. XIV, 2013.