Cost Optimization - A Recommendation Analysis of Azure Workloads

Yuxuan Zhang[1], David Li[2] and Sophia Zhang[3]

[1]UC San Diego University
[2]Westwood High School
[3]Vandegrift High School

ABSTRACT

In a rapidly modernizing world, cloud computing has emerged as an innovative and ground-breaking technology. It is a sector that is experiencing rapid growth, accelerated by the COVID-19 pandemic and remote work environment. Many businesses have started to make the transition from traditional storage systems to the cloud. However, the transition can be opaque and result in an excessive amount of spending on cloud resources. Additionally, these resources are often underutilized, further contributing to cloud waste. This paper seeks to address this problem by designing a comprehensive algorithm based on a detailed analysis of the Microsoft Azure 2019 virtual machine (VM) traces. We first analyzed the cost of approximately 2.7 million VMs and identified that many VMs are underutilized and over consume resources. Then, using a policy-oriented algorithm we constructed a list of VMs that were considered to be "wasteful". Finally, we implemented a recommendation algorithm to sort through this constructed list and recommend lower prices and options for users without interfering with the purpose and function of the VM. Our results show that if all users were to follow our recommendations, the potential cost reduction is over $1.4 million. We conclude with a discussion of the implications of this research and make recommendations for future studies.

I. INTRODUCTION

Cloud computing is currently one of the fastest growing sectors in the technology industry. Fueled by the recent COVID-19 global pandemic, cloud computing has become a key cornerstone in many businesses. In a world where many things becoming digitized, the cloud will become one of the most important parts of the economy. Already, many businesses, both large and small, have started the transition to cloud servers.

However, the transition, especially for many small businesses, can be difficult. Lack of information and expertise can often lead to overspending and over-allocating into unnecessary cloud resources. As a result, many businesses are searching for ways to best optimize their costs and expenditures. Cost optimization is one of the leading initiatives that many businesses are looking to improve on.

This paper aims to expand on existing literature and provide cloud users with a more transparent analysis on cloud spending and resource allocation. We provide a data-driven analysis specifically targeted to help users and businesses to reduce their cost and waste. This research paper studies cost optimization and waste reduction in a cloud computing environment. Specifically, our research pulls data from the 2019 Microsoft Azure traces to analyze trends among user purchased virtual machines. A policy-based recommendation algorithm along with quantitative metrics are employed to identify VMs (virtual machines) as “wasteful.” These VMs are then added to a list from which a recommendation algorithm is used to suggest an optimized cost without drastically reducing the end purpose of the VM.

Our results indicate that if all users accepted and followed the recommendations, approximately $1,471,520 could be saved in cost optimization. Though our recommendations are relatively conservative to preserve the integrity and purpose of each VM, we are still able to vastly reduce potential waste. Our results are scaled down due to the nature of our recommendation algorithm, however, our experimental results show that our cost reduction could be scaled up by loosening our restrictions and thresholds.

II. RELATED WORK

Rapid innovation and growing popularity in cloud computing has opened up new possibilities for data driven analysis within relatively low-risk fields. Mass datasets are becoming increasingly available and open for interpretation. Many companies are starting the transition from traditional working and storage systems to the cloud.

The cloud is beneficial to many users and businesses due to its unique ability to cut down on infrastructure costs [1]. Additionally, lower software costs, scalability, remote-access, collaboration, and higher storage capabilities all make cloud computing an attractive prospect for businesses seeking to grow and innovate [8].

However, like many new technologies, there are also many emerging problems that continue to plague the cloud industry. Cloud computing introduces problems relating to availability, confidentiality, access control, data integrity, security issues, and legal concerns [6].

Many businesses and organizations, however, believe that the most important issue to prioritize is cost optimization and reduction [4]. Overspending on cloud resources can be detrimental to businesses. Small businesses, especially, are often disproportionately affected from a lack of expertise and knowledge.

This paper attempts to fill the literature gap in the use of recommendation algorithms to optimize costs in a rapidly growing sector.

Cost optimization. To the extent of our knowledge, only a few works attempt to integrate data from real public cloud providers such as [7], [9]. None of these studies, however, explicitly consider cost-optimization in their central research.

Waste quantification. Our method of analysis specifically looks to prevent the over-allocating of resources by reducing existing waste. The only other study that attempts to do something similar is [3]. However, our study takes into account waste quantification as part of a broader scope of policies to determine which VMs qualify as "wasteful." The analysis and filtering of VMs via multiple policies ensures a more comprehensive discussion and implementation of a recommendation algorithm.

Recommendation Algorithm. The literature on recommendation algorithms is extensive. Our algorithm attempts to recommend different actions to users based on waste quantification and other policies. Many recommendation algorithms have been studied and implemented in various social media platforms such as YouTube [2], TikTok [10], and Instagram [5]. However, few works study the use of recommendation algorithms in cost optimization in cloud workloads. Our approach attempts to integrate a broader framework of the recommendation algorithm into waste quantification in cloud resources.

III. AZURE DATA ANALYSIS

A. Dataset Background

Currently, Microsoft Azure is one of the three largest cloud providers, and many users use it for various purposes. One of the most highly utilized functions of Azure is it’s ability to create virtual machines (VMs). In 2019, Azure created the AzurePublicDatasetV2: a dataset that tracks VMs created over 30 consecutive days in one of Azure’s geographical regions. During this time period, approximately 6,500 users created over 2.7 million VMs with varying sizes. The size of a VM is defined by the amount of cores (2, 4, 8, 12, 24, and 32) accompanied with the amount of gigabytes of memory storage (2, 4, 8, 32, 64, and 70). Additional relevant information for each VM includes the average core processing unit utilization, the maximum and maximum 95 percent core utilization, the timestamp at which the VM was created and deleted, the subscription identification string of the user that created the VM, and the unique VM identification string.

B. Data Analysis Tools: Orange and R

Throughout the research process, we utilized Orange–a visual programming software package–to visualize, analyze, and pre-process data. This tool consists of a canvas interface where users place widgets to create a data analysis workflow. These components provide basic functionalities such as displaying a data table, selecting features, training predictors, and comparing learning algorithms.

An example of how we employed Orange can be seen in Figure 1. Here we used it to visualize our filtered dataset. Orange makes it easy to filter out certain rows and columns in the entire 2.7 million row dataset. In the beginning, we filtered out unnecessary columns such as deployment id, VM id, and timestamps through the select columns widget.

Even though Orange is a great tool to visualize and analyze our data, it is difficult to add data to an existing dataset. Therefore, we decided to employ the use of R. This enabled us to add and fill in additional relevant information. R served as our main tool for which we developed our final recommendation algorithm due to it's flexible and versatile application.

C. Pricing Model

Before continuing in our analysis, we must first determine the cost of each VM. The initial step toward determining the price is composing the pricing model. These models can easily be constructed through the pricing calculator on Azure's official website. However, when looking through Azure's official website, we noticed that the VM sizes found in the dataset, don't match any of Azure's current series. Therefore, we had to make some assumptions. We decided to use the Av2 series because matched most similarly with the sizes in the dataset. Moreover, we noticed that 37.33 percent of the users only created five or less virtual machines, most of them running for an extremely short period of time. Thus, we can conclude that a large portion of the users are creating virtual machines primarily for testing purposes–whether trying to familiarize themselves with cloud computing or testing different sizes of VMs–and are likely not using them for business activities. When selecting a series to base our pricing matrix on, these users imply that this series is used greatly for testing. When researching all the Azure series, the A series is most notable for being an economical and a low-cost option when getting started with Azure, implying that this series would be relatively accurate for the purposes of our analysis.

After deciding on the A-series, we started to create our pricing model. First, we pulled the prices per hour of the series from the official pricing calculator. The price of a VM changes based on the size (amount of cores and the amount of storage). VMs with a larger size are priced at a higher rate per hour. Limitations in data access, however, prevented us from gathering information on all sizes that were needed to fill the dataset. This limited our ability to assign a cost for many VMs.

To solve this, a linear regression was employed to most accurately determine the price per hour of VMs whose sizes had an unknown cost. Using the prices of sizes that we did have access to, a linear model was constructed to predict values using a line of best fit. In the regression, cost was set as the dependent variable and cores and memory were set as the two independent variables. Predictions were then performed for sizes that didn't have an already existing price. This allowed us to create a pricing matrix with the price per hour of all sized VMs. A summary of the linear model is shown in Table 1. The complete pricing matrix is shown in Table 2.

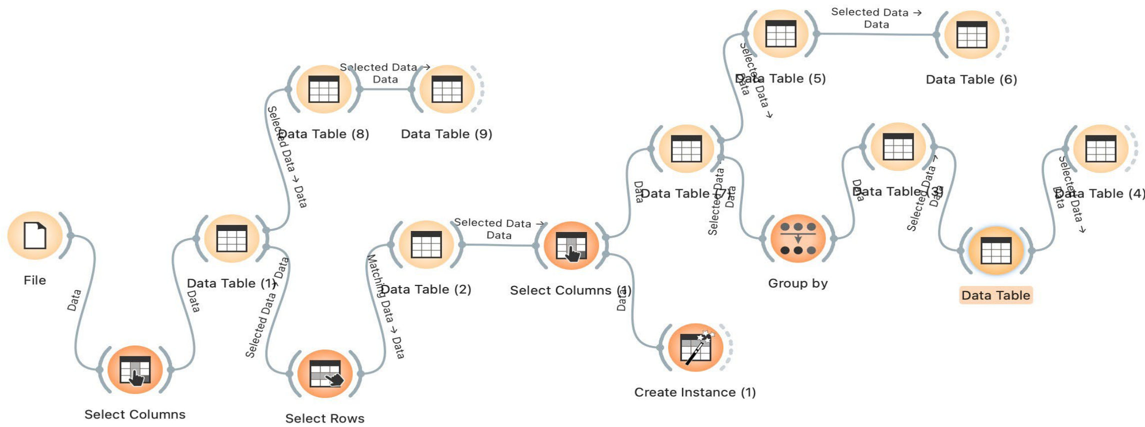

Fig. 1. Orange Workflow: analyzing the dataset

TABLE I
SUMMARY OF PRICING MATRIX LINEAR REGRESSION

Coefficients	Linear Regression			
	Estimate	*Std. Value*	*t Value*	*Pr(>|t|)*
(Intercept)	-0.0038	0.0055	-0.6830	0.5649
cores	0.0468	0.0011	41.778	0.0006
memory	0.0017	0.0001	12.284	0.0066

TABLE II
PRICING MATRIX (PRICE PER HOUR)

	2 GB	*4 GB*	*8 GB*	*32 GB*	*64 GB*	*70 GB*
2 cores	0.093	0.096	0.103	0.142	0.196	0.206
4 cores	0.187	0.190	0.191	0.238	0.289	0.299
8 cores	0.374	0.377	0.384	0.424	0.475	0.486
12 cores	0.561	0.564	0.571	0.611	0.664	0.674
24 cores	1.122	1.126	1.133	1.173	1.225	1.235
30 cores	1.404	1.407	1.414	1.453	1.506	1.516

After constructing the pricing matrix, it is also necessary to calculate the run time in hours for each VM. This is easily done by subtracting the timestamp at which the VM was started from the timestamp at which the VM was ended. Since the timestamps are given in seconds, dividing this number by 3,600 yields the number of hours that each VM was run for.

To calculate the actual cost of each VM the run time is multiplied with the price per hour of the corresponding VM size on the pricing matrix.

D. Waste Quantification

After calculating the price of each virtual machine, it is now possible to accurately create a waste metric.

To further simplify the data into relevant information, certain values can be used to compute each VMs "core hours." Core hours refers to the number of processor units employed to run a virtual machine multiplied by the duration of the job in hours. To accomplish this, a simple algorithm is run to loop through each of the 2.7 million VMs and multiply run time with the amount of cores to compute the core hours.

$Time * Cores = CoreHours$ (1)

When quantifying waste, it is first necessary to determine the dependent variables in order to construct an equation. Thus, we determined that five factors–number of cores, gigabytes of memory, average CPU utilization, run time, and cost–were necessary to include in the equation. Because the cost of a VM already considers the number of cores, amount of storage, and run time, we can further simplify the potential waste equation by only considering cost and utilization. It is important to consider CPU utilization because, by not taking advantage of the VMs full potential, under utilization also contributes to waste.

It is important to note that there exists a direct relationship between cost and waste: when cost increases, waste will also increase. In a similar fashion, we can also observe that utilization and waste have an indirect relationship: as utilization increases, waste decreases. Using these two factors, we constructed our waste equation.

$Waste = Cost(1 - Utilization)$ (2)

After creating the waste equation, it is necessary to verify that it is accurate. To do this, pairs of virtual machines that have the same amount of core hours but different prices are selected. If two VMs have different prices but the same core hours then we know that one of the VMs is wasteful (relative to the other VM). This observation is possible because the core hours represents the total amount of processing power that a VM has, thus concluding how powerful it is. A manual sort and search was applied to 40 pairs of VMs. The waste equation was then applied to these 40 pairs to produce a quantified waste for each VM. All the pairs demonstrated a higher waste number in the VM that, out of the two, had the higher cost. Table 3 exemplifies one of 40 such pairs. The two virtual machines share the same amount of core hours–560–but different prices. VM 1's price is over double the amount of VM 2's price. Therefore, we can determine that VM 1 is more wasteful than VM 2. The results of the waste equation are consistent with our observations.

TABLE III

VM PAIR

VM	*core hour*	*cost*	*cloud waste*
1	560	33.32	32.1048
2	560	15.49	14.5572

E. Observations

- Fig. 2 shows the distribution of core hours of each VM. We can observe that 2-core VMs occupy more than half the total workload in the dataset. 4-core VMs occupy 27.2%, and 8-core VMs occupy 13.6%. 24-core VMs and 30-core VMs make up a small percentage of the total workload.
- Fig. 3 shows the distribution of cost of each different VMs. If we compare Fig. 3 with Fig. 2, we can find that, although 2-core VMs occupy the majority workload, the total cost of 2-core VMs only occupy 42% of the total cost for all VMs. 4-core VMs occupy 34.4%, and 8-core VMs occupy 15.7%, respectively. 2-core VMs occupy more than half of the workload but amount for less than half the cost. 4-core VMs and 8-core VMs occupy less than half of the workload but cost more than half of it. This relationship between Fig. 2 and Fig. 3 demonstrates the opportunity of reducing cost via our recommendation algorithm

- Fig. 4 shows the distribution of average CPU utilization of all VMs. Based on the analysis, 50% of VMs have an average CPU utilization smaller than 8.197% and 80% of VMs have that smaller than 27.7%. This observation demonstrates that many VMs are not utilized to their full potential.
- Fig. 5 shows the distribution of average CPU utilization of all VMs under different cores. We can observe that the CPU utilization of 4-core VMs, 24-core VMs, and 8-core VMs centralize more around a low utilization rate. These configurations are more likely to have wasteful VMs.

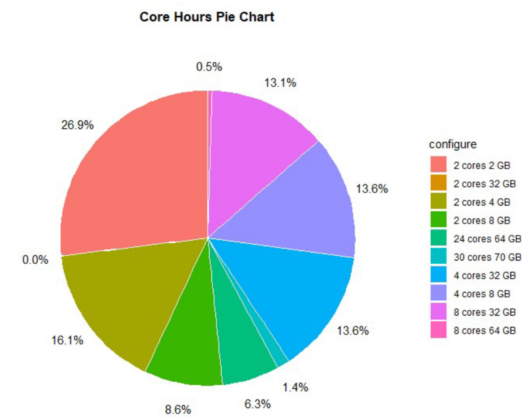

Fig. 2. The Distribution of VM Core Hours.

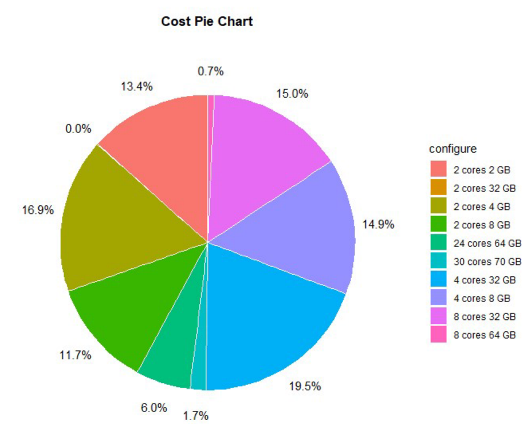

Fig. 3. The Distribution of VM Cost.

TABLE IV
SAMPLE OF ADDED DATA

time period	*cost*	*core hour*	*cloud waste*
309.83	131.4	2478.67	130.41
0.25	0.06	1	0.058
0.167	0.04	0.667	0.0396
719.83	69.1	1439.67	48.14
0.417	0.04	0.833	0.034

F. User Analysis

After we observed all VMs and users holistically, we turned our focus to individual users to figure out why these users were wasting the amount they were. We selected users with extremely high and low waste to find key differences. For each user, all created VMs were searched and the CPU usage, number of cores, and number of gigabytes of each VM was plotted. To distinguish between a wasteful or non-wasteful user, we then filtered out all user created VMs that have a waste below or above a certain threshold. After analyzing these graphs, we were able to determine that size of a VM affects waste the most. A common trend in wasteful users was that the size of their created VMs was often skewed higher. For example, Figure 6 shows a wasteful user using over 1000 large size VMs and only a few small size VMs.

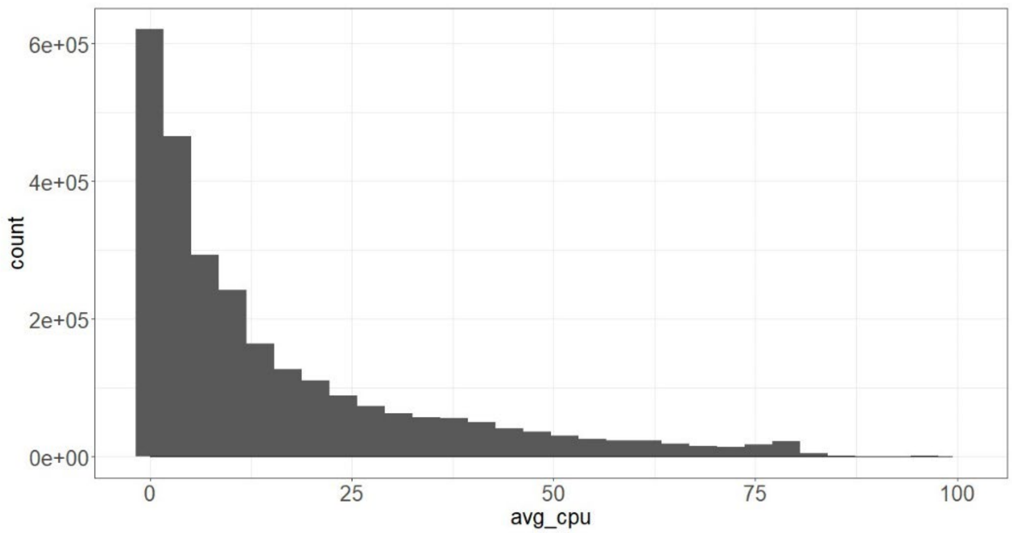

Fig. 4. The Distribution of Average CPU Utilization.

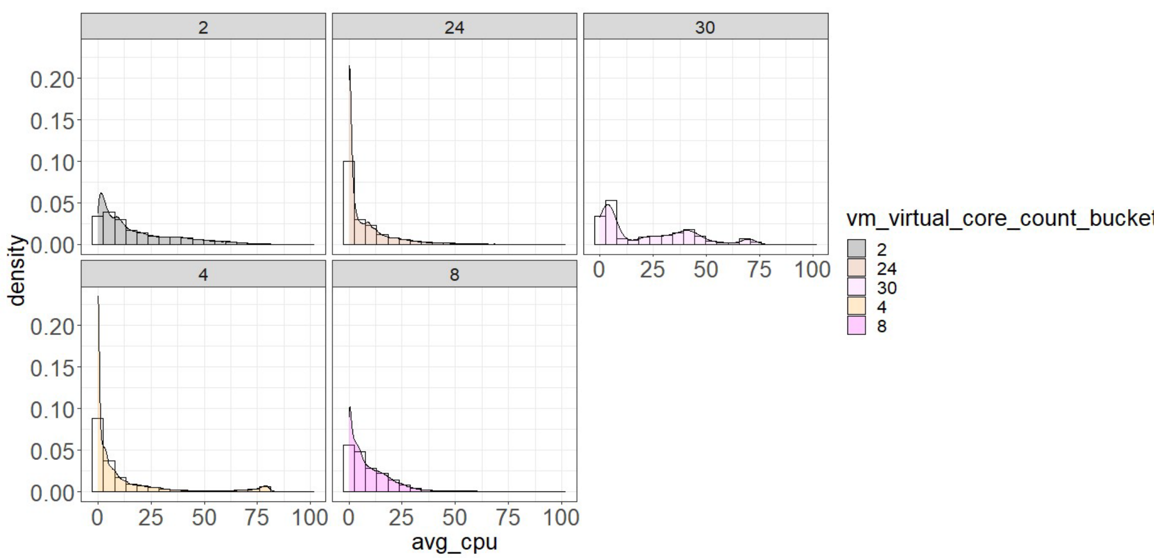

Fig. 5. The Distribution of Average CPU Utilization under Different Cores.

IV. ALGORITHM DEVELOPMENT

A. Recommendation Algorithm

Once we calculated waste and cost and made sufficient observations, we moved on to developing our recommendation algorithm. Our comprehensive analysis seeks to make sense of the relationship between cores, run time, and cost of corresponding VMs. Based on the result, large-sized VMs that are considered to be wasteful are recommended to be changed into smaller-sized VMs. Since we want to find virtual machines that generate a lot of waste, we used a series of policies to filter and set aside VMs that were considered too wasteful. After analyzing these potential VMs, we determined various thresholds that are utilized in our final recommendation algorithm. Finally, we use our recommendation algorithm to provide corresponding recommendations and calculate the reduced cost.

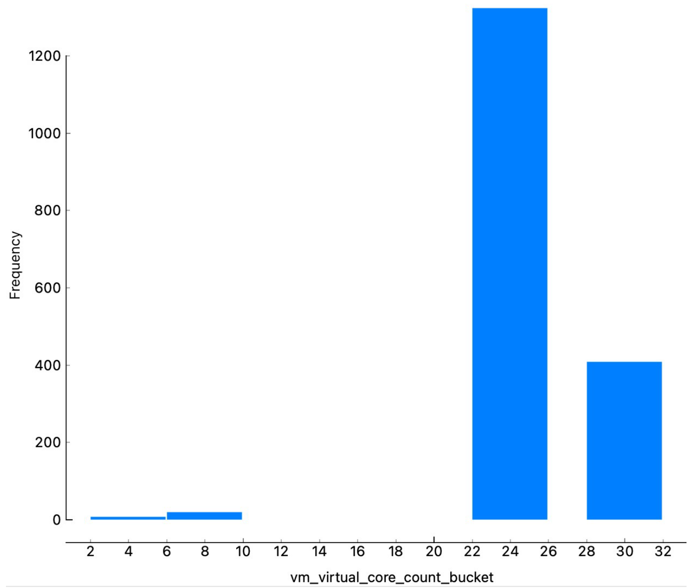

Fig. 6. Wasteful Users: Core distribution

We establish a series of policies to filter the dataset to search for potential wasted VMs. Each individual VM is searched, and the policies are applied as follows:

1) The number of VM cores is larger than 2
2) The average CPU utilization rate is smaller than or equal to 10%
3) The 95_{th} percentile of maximum CPU utilization rate is smaller than 80%
4) The running time period of each VM is larger than or equal to 1hr
5) The customer who created the VM has a total VM count of greater than or equal to 100.
6) The cloud waste of each VM should exceed 50.

The justifications for the policies are as follows. 1. Since the smallest possible VM size is 2 cores, all VMs we consider must be larger than such. 2. A 10% average CPU utilization rate is considered to preserve the conservative nature of our algorithm and overall distribution of average utilization rate. 3. To maintain consistency in our conservative approach, we set a maximum 95_{th} percentile CPU utilization threshold. Because the maximum CPU utilization rate is an extreme value, it can't effectively evaluate the wastefulness of a VM. Thus, we use the 95_{th} percentile maximum CPU utilization rate as a more accurate descriptor. The threshold for percentile 95_{th} maximum CPU utilization rate is set to 80% to account for the jump between a 30-core VM to a 24-core VM. 4. VMs that do not run for a prolonged period of time are likely of lesser value to our analysis due to their temporal status. 5. To reduce the workload and focus on main customers, the data is filtered to only include customers that have at least 100 VMs. 6. VMs must have substantial levels of waste.

In accordance with the policies above, the entire dataset was filtered and evaluated to only include wasteful VMs. Using this new dataset, we developed an algorithm to provide recommendations for these VMs. To implement this algorithm, we must first determine the conditions for which it is permissible to decrease the size of a VM. We assume that only the VM size is affected by our algorithm while all other properties of the VM remain the same. The following is an example of how the algorithm works. Based on how CPUs function, if a user fully utilizes 100% of a 30-core VMs, each core of the VM should shoulder around 3.33% of the total processing power. If we replace a 30-core VM with a 24-core VM, each core would also take on the same amount of power. Thus, a 24-core VM can maintain approximately 80% of the power of a 30-core VM. If the maximum power of a 30-core VM doesn't exceed 80%, we can reduce the VM size to 24-cores. A similar method of calculation can allow us to determine the proper threshold for changing a 24-core VM to a 12-core VM. Therefore, to determine the threshold when lowering a VM, we should look for the highest power requirement of each VM. But because the maximum CPU utilization rate is an extreme and arbitrary number, we decided to utilize 95_{th} percentile maximum CPU utilization instead. To quantify this idea, we calculated the threshold of 95_{th} percentile maximum CPU utilization for each VM size.

TABLE V
THRESHOLD OF P95 MAXIMUM CPU UTILIZATION.

configure	4-8	4-32	8-32	8-64	24-64	30-70
2-8	50					
2-32		50	25	25	8.4	6.7
4-32			50	50	16.7	13.3
8-64					33.4	26.7
24-64						80

These thresholds are applied throughout the entire the dataset to generate recommendations for each VM. Although some VMs in the dataset are not given a recommendation, the majority of the VMs receive one. When VMs are recommended a lower size, the amount of memory is kept consistent in order to decrease the amount of variation and inaccuracies in our algorithm.

Algorithm 1: Recommendation for potential wasted VMs

Given: The T threshold Table V contains the thresholds of the percentile 95% CPU utilization under each different configuration from 4 cores 8 GB VMs to 30 cores 70 GB VMs. The preprocessed, augmented dataset V contains a set of VM records.

client input: V.subscription_id, V.avg_cpu, V.p95_max_cpu, V.vm_virtual_core_count_bucket, V.vm_memory_(gb)_bucket, V.time_period, V.cost, V.cloud_waste T.4-8, T.4-32, T.8-32, T.8-64, T.24-64, T.30-70

function FILTER_VM_OUT(V)

newV = V[V.vm_virtual_core_count_bucket > 2 *and* V.avg_cpu ≤ 10 *and* V.p95_max_cpu ≤ 80 *and* V.time_period ≥ 1 *and* V.cloud_waste > 50,] newV ▷ a dataframe only contains VMs with more than 2 cores, average CPU utilization rate lower than or equal to 10%, percentile 95_{th} maximum CPU utilization lower than or equal to 80%, and at least 1 hour of run time

end function

function FILTER_100_CUSTOMERS(V)

count = table(V.subscription_id) ▷ count the number of VMs of each customer

```
        qualified_customers = count.subscription_id[count.Freq ≥ 100]
        for    for    each_qualified_customer ∈ qualified_customers do
            temp_df  =  V[V.subscription_id == each_qualified_customer, ]
            newV = rbind(newV, temp_df) ▷ combine each dataframe together
        end for newV ▷ a dataframe only includes customers that have more than or equal to 100 VMs
end function
function PROVIDE_SINGLE_RECOMMENDATION(one_row_V,
 T)
    config = V.vm_virtual_core_count_bucket +
 V.vm_memory_(gb)_bucket
        thres = T.config
        for for each thre ∈ thres do
          if vm.p95Max ≤ thre then
              return a new level of VM
          end if
     end for
end function
--------------------------------------------------
function PROVIDE_ALL_RECOMMENDATIONS(V,T)
        filtered_V = FILTER_100_CUSTOMERS(V)
        filtered_V = FILTER_VM_OUT(filtered_V)
        for for each potential cloud-wasted vm ∈ filtered_V do
            new_vm = Provide_Single_Recommendation(vm, T)
            all_new_vm = all_new_vm.append(all_new_vm, new_vm)
     end for all_new_vm    ▷ a list of all recommendations
end function
```

V. EXPERIMENTAL RESULTS

After implementing our algorithm, the total cost of the new set of VMs was \$2,240,626. Before the implementation, the cost of all of the VMs that were considered wasteful was \$3,712,146. The total reduction in cost is \$1,471,520. Using these numbers, we determined that there was a 39.641% reduction in customer cost. Due to the conservative nature of our policies, the total cost of the VMs we marked as wasteful makes up only 18.53445% of the total cost of the original dataset. If we were to take a more liberal and loose approach to our policies, we would likely be achieve higher optimization rates. For example, if we reduced the 95_{th} percentile maximum CPU utilization threshold, more VMs likely would've been marked as wasteful, therefore increasing the amount of possible waste reduction. In a similar logic, increasing the average CPU utilization threshold to 15%, decreasing the run time threshold to 0.5 hours, or even decreasing the required customer VM count to 50 can all contribute to an increase in cost reduction.

If we modify the policies according to the descriptions above, the total cost of wasteful VMs is \$4,277,758, and the total reduced cost is \$2,638,791. This is a 38.314% reduction in cost, demonstrating the effectiveness of our algorithm.

VI. CONCLUSION

While our results are fairly accurate, there are some limitations to our research. An example of such is our pricing model. Because cloud computing is experiencing rapid growth, the sector is constantly changing. For instance, the VM sizes that appeared in the original dataset did not match any existing Azure VM offers. There was a high possibility that the series was no longer available. Furthermore, due to a lack of information, we had to assume that every VM in the dataset was of the same series. Thus, we had to select the best possible replacement, the A series, to employ when creating our pricing model.

A linear regression had to be implemented to absolve and accommodate for all possible VM sizes, losing some precision in our analysis.

There are also limitations in our consideration of storage and memory usage in our waste calculation. Due to the nature of the dataset, there lacks sufficient information such as memory utilization in order to properly factor it into our analysis.

Due to the nature of our recommendation algorithm, we do not account for any VMs that have only 2 cores. However, 2-core VMs occupy a large portion in the cloud services. Potential future work could seek to accommodate for these VMs in order to further identify and reduce cloud waste.

This research plays a significant role in filling the literature gap between the cost optimization of cloud workloads and a policy-driven recommendation analysis. By analyzing over 2.7 million VMs from a real Azure workload, we were able to identify trends in consumption and utilization. Via a quantitative analysis of waste and additional policy implementation, certain VMs were identified as wasteful. And finally, by constructing a recommendation algorithm, we were able to greatly reduce cloud waste in the workload.

ACKNOWLEDGMENT

We would like to thank our mentor, Dr. Zong, for helping guide us through our research process. We would also like to thank Azure for providing us with the dataset that was used in our analysis.

REFERENCES

[1] Aljabre, A. (2012). Cloud Computing for Increased Business Value. *International Journal of Business and Social Science*, *3(1)*, 234-239.

[2] Buntain, C.L., Bonneau, R., Nagler, J., Tucker, J.A. (2021). YouTube Recommendations and Effects on Sharing Across Online Social Platforms. Proceedings of the ACM on Human-Computer Interaction, 5, 1 - 26. https://doi.org/10.1145/3449085.

[3] Everman, B., Gao, M., Zong, Z. (2022). Evaluating and educing cloud waste and cost—A data-driven case study from Azure workloads. *Sustainable Computing: Informatics and Systems*, *35*. https://doi.org/10.1016/j.suscom.2022.100708.

[4] Flexera. (2022). 2022 State of the Cloud Report.

[5] Mehlhose, F., Petrifke, M., Lindemann, C. (2021). Evaluation of Graph-based Algorithms for Guessing User Recommendations of the Social Network Instagram. *2021 IEEE 15th International Conference on Semantic Computing*, 409-414. https://doi.org/10.1109/ICSC50631.2021.00075.

[6] Namasudra, S. (2018). Cloud computing: A new era. *Journal of Fundamental and Applied Sciences*, *10(2)*, 113-135.

[7] Shahrad, M., Fonesca, R., Goiri, I., Chaudhry, G., Batum, P., Cooke, J., Laureano, E., Tresness, C., Russinovich, M., Bianchini, R. (2020). Serverless in the Wild: Characterizing and Optimizing the Serverless Workload at a Large Cloud Provider. *2020 USENIX Annual Technical Conference*, 205-218.

[8] Truong, D. (2010). How Cloud Computing Enhances Competitive Advantages: A Research Model for Small Businesses. *The Business Review*, *15(1)*, 59-65.

[9] Vilarinho, E.C., Bonde, A., Muzio, A., Russinovich, M., Fontoura, M., Bianchini, R. (2017). Resource Central: Understanding and Predicting Workloads for Improved Resource Management in Large Cloud Platforms. *Proceedings of the 26th Symposium on Operating Systems Principles*, 153-167. https://doi.org/10.1145/3132747.3132772.

[10] Zhang, M., Liu, Y. (2021). A commentary of TikTok recommendation algorithms in MIT Technology Review 2021. *Fundamental Research*, *1(6)*, 846-847. https://doi.org/10.1016/j.fmre.2021.11.015.