Cross Platform Analysis of Cloud Utilization Patterns for Optimized Resource Allocation

Project Progress Report - CS 8803 DNS

Akshat Karwa, Mehul Rastogi

March 28, 2025

Project Objective

This project involves comapring some of the largest cloud service providers — *Google Cloud, Microsoft Azure, Alibaba Cloud.* Utilizing their publicly available datasets (released about twice a decade), we will be looking at the resource patterns on one of their clusters.

We will try to quantify ***the utilization gap*** which is the discrepancy between the resources allocated and the observed actual utilization. We wish to understand the magnitude and patterns of inefficiencies in the workings of cloud providers. Furthermore, we will contrast the ways of resource allocation of the providers and compare their scheduling algorithms and strategies. We will also evaluate how effectively the clusters handle workloads under diverse conditions. A goal is to compare the different providers in how they handle unpredictable changes in workloads such as spikes or seasonal variations or changing usage patterns with time. Another goal is to identify the best resource management techniques that can be generalized as the best practices in the industry. We will look at provider-specific approaches as well that have the best outcomes.

Overall, our insights will help reduce over allocation of resources, reduce idle capacity at most times and consequently lower operational costs. Through this, we hope to provide the community with a better understanding of resource patterns and management strategies.

Motivation

With the continuous growth of Cloud computing, efficient resource management has become a critical challenge. Our project is motivated primarily by the utilization gap that leads to over-provisioning, idle capacity, and suboptimal responses to variations in workloads leading to unnecessary costs and reduced efficiency. Individual studies in the past have analyzed traces from specific providers. Our comparative analysis examines different providers and how they handle challenges. There is a lack of valuable insights into common patterns and unique approaches of different cloud providers, and there is limited research in comparative analysis across platforms. By normalizing and analyzing heterogeneous datasets from Google Cluster Data (2019), Microsoft Azure Public

Dataset (2017-2019), and Alibaba Cluster Trace Program Data (2018), we aim to determine whether effective techniques can be generalized as industry best practices.

The economic stakes are substantial, Microsoft's research [2] demonstrates that companies with large cloud deployments can gain valuable insights through deep trace analysis. The Azure dataset we work with alone comprising of 2.7 Million VMs and 469.4 Million core hours and thus, even small improvements in resource utilization can significantly save costs and benefit the environment by reducing data center energy consumption. We can potentially help companies reduce operational costs and make better design decisions. Generalizable insights would lead to more accurate resource allocation reducing cost of operating data centers.

Challenges

In this project, there are numerous challenges some of which we've already encountered in our implementation so far. We anticipate encountering more challenges as we proceed with the project. The primary ones are:

- The trace data which we are using from all the three sources - Alibaba, Google, and Azure is extremely heterogeneous. Data from different cloud providers has different schema and relationship models. The metrics calculated and reported by each of them are also completely different. Due to this, it becomes exceedingly difficult to perform analysis which compares and contrasts the trace data across the three providers in an efficient manner. For example, Azure doesn't have data on memory usage. It only contains data on CPU/compute usage.

- A big challenge is to analyse the huge datasets with the limited amount of compute resources we have. It's challenging to efficiently process the data from all the three sources because of their sheer magnitude. Therefore, compute bottlenecks is a big challenge for this project. For example, we have Azure's data for approximately $\approx$ 2.7 million virtual machines. For Alibaba, we have the data for 4000 machines for a period of 8 days. For Google, the dataset is $\approx$ 2.14 TiB compressed. Thus, we will be using data from one of the eight clusters. For now, we have utilized 10 million rows for cluster a for the Google Dataset.

- Another challenge is a lack of available documentation to efficiently understand the schema for the Alibaba traces. Different approaches need to be tried and tested in-order to find the correct set of tables to merge in-order to get accurate values that represent the metrics we are looking for. On top of that, the three cloud providers have different mechanisms for job/task scheduling and processing and how they tackle workloads. This makes it difficult to find generalizable insights across the three datasets. The timestamp durations for which the data is given for each is different as well. Different technologies also need to employed for different datasets, BigQuery, Pandas, etc. to use the most efficiency way to process data.

- Since the variables present in the data are different, doing predictive modeling across the three cloud providers is hard. Due to this, giving generalizable trends, becomes even harder. Therefore, finding the right variables to perform predictive analysis and generalize trends for compute/memory usage across three providers is another challenge.

Approach

Firstly, our central goal is to standardize the data into a homogeneous format. For each of the cloud providers, the data needs to be standardized such that for a machine or VM instance, we know the percentage of compute resources used as compared to the compute resources that were assigned. Similarly, for a machine or VM instance, we need to know the percentage of memory resources used as compared to the resources that were allocated. We successfully completed this for the three cloud providers.

Google

We worked with the official Google Cluster Trace Data from 2019 - [5]
We analysed Google's cluster traces and looked at the two most important tables - *instance_events_data* and *instance_usage_data* - merging them using *instance_index*, *machine_id*, and *collection_id* as join keys. Through this merge, we were able to correlate resource requests with actual utilization patterns across all the instances. The data was from one cluster (a) out of the 8 clusters (a through h) in the dataset. For now, we have processed our implementation against 10 Million rows of data. Our plan is to further extrapolate the same implementation to the entire dataset by scaling our compute resources. The visualizations below reveals the utilization patterns we found - CPU usage followed a distribution with most machines operating below 20% capacity and memory utilization showed a distribution with peaks at $0-5\%$, $30-40\%$, and $55-65\%$. Based on these patterns, we see significant opportunities for resource optimization because we see that many machines maintain low CPU utilization. Memory usage also clusters in some specific thresholds which means that we can have more efficient memory allocation strategies to optimize allocations.

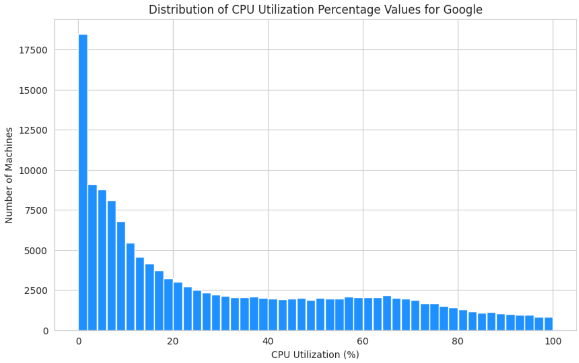

Figure 1: CPU Utilization Distribution - Google

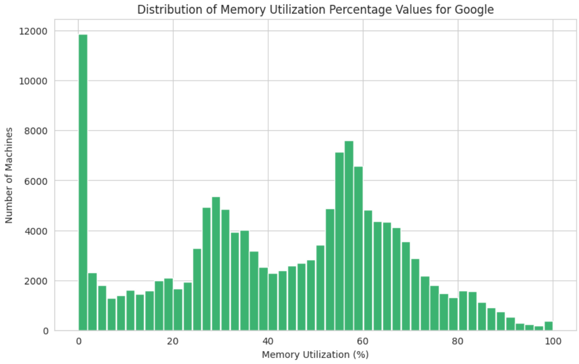

Figure 2: Memory Utilization Distribution - Google

Azure

We used the Azure's Publicly available Trace Data from 2019 - [9]
For Azure Trace data, we have primarily worked with data in the `vmtable.csv` file. This csv file has data for approximately 2.7M VM instances. For each of these instances, we used the percentage values for the Average, Maximum, and P95 Max utilization rates for CPU resources. This essentially depicts the average, maximum, and P95 maximum percentage of CPU resources used by the respective VMs.

As we can see in Figure 3, the Average CPU utilization is relatively low for all the VMs. This gives us insight that on average, CPU utilization is not that high as compared to the allocated resources. And, in Figure 4 and 5, the Max and P95 Maximum CPU utilization tells us that some of the VMs actually end up using all or most of their allocated CPU resources. These distributions actually give us immense insight into the allocation strategies used at Azure.

We plan on further analyzing the data to get more generalizable insights about allocation strategies used at Azure as compared to other cloud providers.

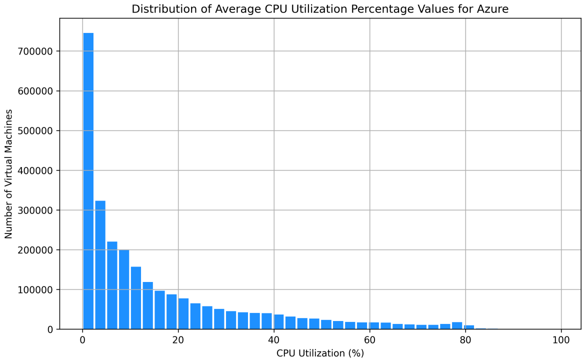

Figure 3: Average CPU Utilization Distribution - Azure

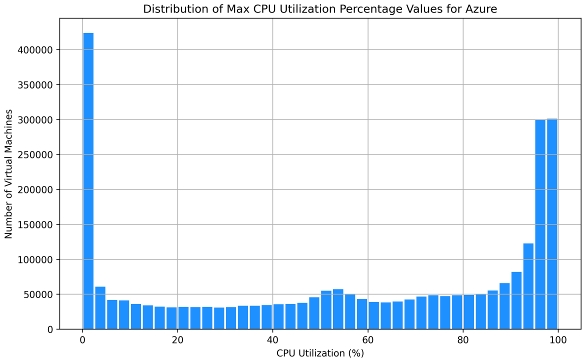

Figure 4: Maximum CPU Utilization Distribution - Azure

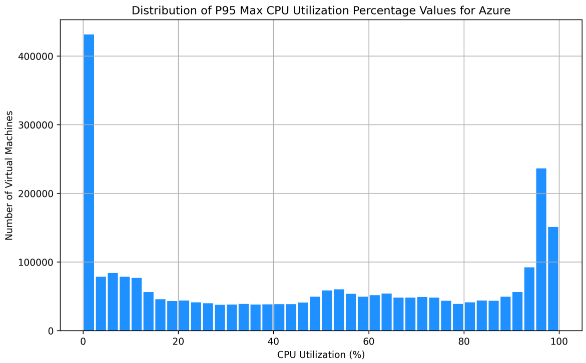

Figure 5: P95 Max CPU Utilization Distribution - Azure

Alibaba

For Alibaba, we are using Alibaba's cluster trace data from 2018 - [7]
To analyze the resource utilization patterns in Alibaba's cluster data, we extracted 10 million rows of data from the machine usage dataset. Due to compute linmitations, we have worked with 10 million rows for now. Then, we calculated the CPU and memory utilization percentages and created histogram visualizations to understand the distribution patterns across the cluster. Figure 6 shows the number of machines with their CPU utilization percentages and Figure 7 shows the number of machines with their Memory utilization percentages. The plots below reveal the broad patterns: CPU utilization follows a normal distribution centered around $30-40\%$ and memory utilization shows a highly skewed distribution where most machines operate at $85-95\%$ capacity. Memory is probably constrained in the cluster. We feel that the allocation of CPU needs to be optimized within Alibaba's infrastructure.

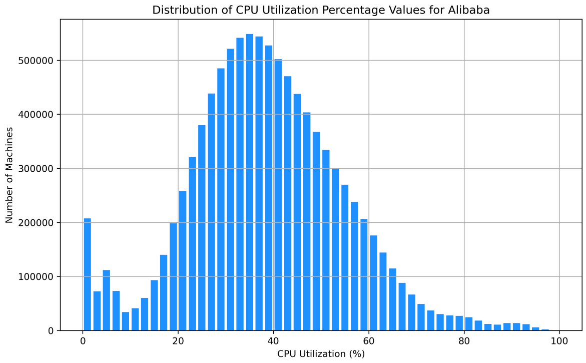

Figure 6: CPU Utilization Distribution - Alibaba

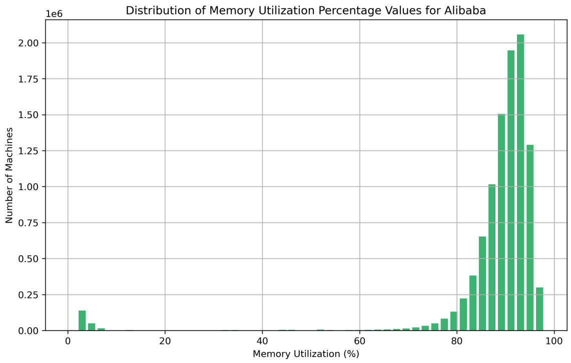

Figure 7: Memory Utilization Distribution - Alibaba

Overall, we can see that the CPU Utilization for Alibaba is the best as most machines utilize more than 35% of the CPU allocated. For Azure and Google, we see most machines only utilizing less than 35% of the CPU allocated which indicates over-allocation of CPU. We do not have the memory utilization data for Azure. For Azure, on average, the CPU utilization is extremely low. However, for some instances utilization peaks for many VMs as can be seen from the P95 and Max utilization graphs. Comparing Google and Alibaba, we find that most machines utilize more than 80% of the memory allocated to them by Alibaba which is extremely good optimization on their part. For Google, we see three peaks, most machines either utilize less than 5%, or ≈ 30%, or ≈ 55% of the memory allocated. This implies that Google overallocates memory by a huge margin and there are significant optimizations required.

Future Work

Moving forward, we wish to do the following things:

- Using the standardized data we have, we aim to conduct a deeper analysis on each of the three trace datasets in isolation.
- Utilize the machine-level and instance-level data for CPU and memory, to conduct trace-based simulation for each cloud provider.
- Formulate concrete generalizable trends across the three providers by analyzing the combined and homogeneous dataset we have created, and present actionable results.
- We will also explore CPU/Memory usage and allocation with respect to timestamps for all the three datasets. Looking at time series data for different machine/VM instances across providers, we want to be able to arrive at more generalizable trends.

- We will also look at the other data tables we have to gather more metrics which can help us in analyzing resource usage vs. allocation patterns more coherently. Understanding these would help us better understand the internal workings of cloud providers.
- We will also perform prediction analysis depending on other metrics to understand the compute and memory resources required for tasks based on Task Priority, Allocated Resources (CPU and Memory), Task Type, Machine ID, etc.

Overall, this will help us better understand how each of the providers handle workloads and allocate resources. We will be able to contrast their techniques and highlight the best of each.

Related Work

Previous research has primarily focused on analyzing cloud traces from individual providers instead of conducting comparative analyses across different cloud platforms. For Google Cloud, Reiss et al. [1] completed a comprehensive analysis of Google traces to reveal a lot of heterogeneity in the cluster workloads also identifying inefficiencies in infrastructure. Similarly, Lu et al. [3] analyzed Alibaba's trace data and highlighted imbalances during resource allocation. Their findings shows how workload characteristics keep impacting resource utilization. For Microsoft Azure, Cortez et al. [2] developed Resource Central, a system which predicts workloads for improved resource management in large cloud platforms. Their research leveraged Azure's large dataset and demonstrated the value of trace data analysis in optimizing cloud infrastructure. The gap in comparative research across cloud providers presents a significant opportunity. There are many individual trace analyses which provide valuable insights into specific platforms but they don't address whether observed patterns are provider-specific or represent industry-wide phenomena. These studies do not show any common issues or common effective techniques. This limits our ability to develop generalizable principles for resource management. Our work aims to address this gap.

Evaluation Plan

The results so far have been discussed in depth in the Approach section to ensure that the background is clear. Then, we have discussed the future plans. In this section, we dive into future timeline and measures of success.

29th March - 5th April

- Scale compute by paying for more resources and then process our logic for the entire dataset (Google and Alibaba). We've figured out a way to scale the compute resources using Google Colab.

- Using the homogeneous dataset that we have created across the cloud providers, we want to perform even deeper data analysis.

6th April - 13th April

- Summarize all the generalizable trends in a succinct and coherent manner so that we are able to provide actionable results. This will help the cloud providers to improvise resources and decrease costs.
- Perform time series analysis for each of the three cloud providers at scale.

13th April - Project Due

- Involve different regressor variables in our analysis, in-order to perform prediction and/or forecasting using Machine Learning/Reinforcement Learning models.
- Finalize Report and Presentation.

Responsibilities

Both of the team members, Mehul Rastogi and Akshat Karwa, contributed equivalently to the project. We both worked together in-order to understand the datasets, write code, perform analysis, and compile the project.

Bibliography

References

[1] Reiss, C., Tumanov, A., Ganger, G. R., Katz, R. H., Kozuch, M. A., Intel Science and Technology Center for Cloud Computing, & Carnegie Mellon University. (2012). Towards understanding heterogeneous clouds at scale: Google trace analysis (Report ISTC-CC-TR-12-101). Intel Science and Technology Center for Cloud Computing. https://www.pdl.cmu.edu/ftp/CloudComputing/ISTC-CC-TR-12-101.pdf

[2] Cortez, E., Bonde, A., Microsoft, Muzio, A., Russinovich, M., Fontoura, M., & Bianchini, R. (2017). Resource Central: Understanding and predicting workloads for improved resource management in large cloud platforms. In *Proceedings of SOSP'17*. ACM. https://www.microsoft.com/en-us/research/wp-content/uploads/2017/10/Resource-Central-SOSP17.pdf

[3] Lu, C., Ye, K., Xu, G., Xu, C.-Z., & Bai, T. (2017). Imbalance in the cloud: An analysis on Alibaba cluster trace. In *Proceedings of the IEEE Conference*. IEEE. https://ieeexplore.ieee.org/document/8258257

[4] Google. (n.d.). Google cluster-usage traces v3. Google Docs. https://drive.google.com/file/d/10r6cnJ5cJ89fPWCgj7j4LtLBqYN9RiI9/view

[5] Google. (n.d.). GitHub - google/cluster-data: Borg cluster traces from Google. GitHub. https://github.com/google/cluster-data

[6] Alibaba. (n.d.). clusterdata/cluster-trace-v2018/schema.txt at master · alibaba/clusterdata. GitHub. https://github.com/alibaba/clusterdata/blob/master/cluster-trace-v2018/schema.txt

[7] Alibaba. (n.d.-b). clusterdata/cluster-trace-v2018/trace_2018.md at master · alibaba/clusterdata. GitHub. https://github.com/alibaba/clusterdata/blob/master/cluster-trace-v2018/trace_2018.md

[8] Azure. (n.d.-a). AzurePublicDataset/AzurePublicDatasetLinksV2.txt at master · Azure/AzurePublicDataset. GitHub. https://github.com/Azure/AzurePublicDataset/blob/master/AzurePublicDatasetLinksV2.txt

[9] Azure. (n.d.). GitHub - Azure/AzurePublicDataSet: Microsoft Azure Traces. GitHub. https://github.com/Azure/AzurePublicDataset