Project Proposal

Akshat Karwa, Mehul Rastogi

January 31, 2025

1 Background

A major challenge in cloud resource management is the mismatch between allocated resources and the actual utilization, also known as *the utilization gap*, which leads to inefficiency in cloud infrastructures. Another challenge is to determine how to accurately predict resource requirements for workloads which if not done accurately leads to overestimation of resource needs and overprovisioning or overallocation of resources. Overprovisioning leads to idle capacity during normal operations. Rapid workload variations also cause suboptimal responses. Overall, these scenarios cause unnecessary costs for consumers and reduced infrastructure efficiency for providers.

Recent research and studies have analyzed the cloud traces of individual cloud providers such as Google's Cloud Traces [1] and Alibaba Public Traces [3]. However, there is a lack of valuable insights into common patterns and unique approaches of different cloud providers like Google Cloud, Microsoft Azure, and Alibaba Cloud. There is limited research in comparative analysis of cloud providers and limited availability of comparison data. This data and research would help in contrasting their load handling techniques, resource allocation strategies, and workload characteristics. Our research aims to conduct this much needed comparative analysis.

2 Motivation

The purpose of this study is to analyze publicly available trace data in a manner such that it leads to reduced overhead and prevents over-allocation of resources. At a large scale, this would lead to reduced costs and less data center usage.

There has been research done on Google's trace data [1] that also highlights key inefficiencies in Google's infrastructure. Similarly, [3] shows the imbalances during resource allocation in Alibaba Cloud. However, generalization of the patterns in Google's trace data and Alibaba's trace data have not been proved. There is a need to compare and contrast the traces from different cloud provider companies in-order to get insights that are generalizable. If we can come up with common inefficiencies across the providers, then it would be greatly benefit for all cloud stakeholders. Generalizable insights would lead to more accurate resource allocation thereby reducing cost of operating data centers. Consequently, companies would make more informed design choices.

In this project, we aim to understand what makes Google, Microsoft and Alibaba's Cloud infrastructure management strategies effective or ineffective. The motivation is to contrast trace data in a manner such that we can highlight similarities and dissimilarities related to resource allocation and utilization between different cloud providers. This would potentially help the companies reduce cost or make better design decisions for the future. Also, based on a research paper authored by Microsoft [2], organizations that have extremely large cloud bandwidths can gather a large amount of important information and insights into inner workings by doing a deep analysis of their trace data. This scrutinization of trace data would in turn help them manage their available resources in a much more robust and efficient manner. Through this project, we aim to not only analyse a single cloud provider but to compare and contrast different companies to understand the most effective methods and best practices of the industry.

3 Approach

We will compare resource utilization patterns across public trace data from Google Cloud, Alibaba Cloud, and Microsoft Azure.

3.1 Data

We will utilize the following data: **Google Cluster Data (2019), Microsoft Azure Public Dataset (2017-2019), Alibaba Cluster Trace Program Data (2018)**.
All cloud providers store data differently and we will address the heterogeneous nature of the three datasets and normalize them to a common format.

3.2 Utilization

- First, we will calculating per-VM resource utilization distribution, and then we will identify patterns in resource waste and over provisioning, followed by comparing the allocated vs actual resource usage for both CPU and Memory.
- We will analyse the variations in utilization across different time scales and also analyze peak-to-average ratios across providers. Moreover, we will identify the impact of seasonality and workload types on utilization.
- We will study how mixed workloads affect utilization and also look at workload characterizations by different providers.

3.3 Comparative Analysis

- Starting by comparing average utilization rates across different cloud providers, we will analyze allocated vs utilized resources and evaluate each provider's allocation effectiveness while also identifying common patterns and uniqueness.

- Then, we will cross validate findings across time periods and also understand how policies differ at rapid workload variations.

4 Schema

4.1 Google

Google's data for each cluster is divided into multiple cells from a to h - **(a, b, c, d, e, f, g, h)**. The data for each cell is in a folder with name - clusterdata_2019_{x}. Inside each cell, data is split into shards. For each shard, we have 5 tables. The 4 tables that we are interested in for our use case are: `Collection Events`, `Instance Events`, `Machine Events`, and `Instance Usage`.

Detailed in-depth schema for each table and relationship diagrams between entities can be found at [4]. From this document, we have pulled the schema for the most crucial/important tables that we will be utilizing for our project. The schema is as follows:

Schema of the `Collection Events` Table: **time**, **type**, **collection_id**, **scheduling_class**, **collection_type**, **priority**, **alloc_collection_id**, **user**, **collection_name**, **collection_logical_name**, **parent_collection_id**, **start_after_collection_ids**, **max_per_machine**, **max_per_switch**, **vertical_scaling**, **scheduler**

Schema of the `Machine Events` Table: **time**, **machine_id**, **type**, **switch_id**, **capacity**, **platform_id**, **missing_data_reason**

Schema of the `Instance Events` Table: **name**, **value**, **relation**

GitHub for Google Trace Data - [5]

4.2 Alibaba

This data has 6 tables: `Machine Meta`, `Machine Usage`, `Container Meta`, `Container Usage`, `Batch Task`, and `Batch Instance`. The schema for all the tables is available in the schema details link below. The main table for our use case would be `Batch Instance`, the schema for which we have pulled in the proposal from the same link in-order to provide clarity. The schema for `Batch Instance` is:

instance_name, **task_name**, **job_name**, **task_type**, **status**, **start_time**, **end_time**, **machine_id**, **seq_no**, **total_seq_no**, **cpu_avg**, **cpu_max**, **mem_avg**, **mem_max**

Alibaba Trace Data Schema: [6], Alibaba Cluster Data: [7]

4.3 Azure

The Azure dataset comprises **2.7M** VMs spanning **6.7K** subscriptions and **33.2K** deployments, with **162.5M** VM hours and **469.4M** core hours of utilization data. The schema for VM Table Data is as follows:

vmid, **subscriptionid**, **deploymentid**, **vmcreated**, **vmdeleted**, **maxcpu**, **avgcpu**, **p95maxcpu**, **vmcategory**, **vmcorecountbucket**, **vmmemorybucket**, **lifetimecorehour**

The schema is in the **schema.csv** file at [8]. The cumulative data is in the files - **vmtable.csv** and **deployments.csv**. The data is also divided into 195 shards with the csv format for the requisite files being:

$$\textbf{vm\_cpu\_readings-file-}x\textbf{-of-195.csv} \qquad \forall \mathbf{x} \in [1, 195]$$

Each row in each of these files contains the following data:

$$\textbf{vmid}, \textbf{min cpu}, \textbf{max cpu}, \textbf{avg cpu}$$

The entire dataset can be found at: [9]
Links to download the dataset are at: [8].

5 Planned Outcomes

We plan to study VM lifecycles, and resource utilization patterns, and identify similar, dissimilar approaches of cloud provider companies. We hope to gain insights that show whether techniques can be generalized across providers or not. We believe that a comprehensive comparative analysis will certainly contribute to a better understanding of cloud resource utilization patterns and the most effective strategies. This can guide future work in the area and be a key reference point for anyone who wants to look at a side-by-side comparative analysis of cloud providers.

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