ISSN: 2096-3246
Volume 58, Issue 02, May, 2026

TELEMETRY-DRIVEN FINOPS AUTOMATION FOR COST OPTIMIZATION IN MULTI-CLOUD ENVIRONMENTS

Mr. Gurjot Singh Gill[1], Mr. Touseef Ahmad Lone[2]
[1]Research Scholar, Department of Computer Science and Engineering,
CT University, Ludhiana 142024, India
[2]Assistant Professor, Department of Computer Science and Engineering,
CT University, Ludhiana 142024, India
Email: gurjot2912@gmail.com[1], Lonetouseef99@gmail.com[2]

Abstract: The dynamic nature of the provisioning and instances mixed with the heterogeneous configurations and cost-based pricing models that have come with the rapid implementation of multi-clouds has presented a tremendous challenge to cost governance. The constant over-provisioning, idle computer resources and the ineffective allocation of storage are often the cause of unnecessary operational cost. In this study, a telemetry-based FinOps automation framework will be proposed gaining insights on structural and behavioural inefficiencies across the Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP). Descriptive statistics, correlation modelling, a threshold-based idle detection and a rule-based rightsizing decision framework were used to analyse a production-representative multi-cloud data. The cross-provider cost-efficiency analysis has been based on the standardised metrics, such as cost per vCPU-horn. Findings indicate a moderate on the whole utilisation, heavy reliance on cost in dependence on provided capacity rather than on the actual requirement to work and high potentials of optimisation. The model of simulation of an optimisation impact simulation proves the cost savings that can be measured by the automated policy of idle shutdown and rightsizing. The results indicate that inefficiencies in clouds are mainly due to provisioning methods than provider choice which highlights the necessity of telemetry constantly and automated government. The suggested structure involves the provision of a repeatable and systematic approach to enhancing cost-performance fit in heterogeneous multi-cloud setup and promoting the active implementation of FinOps.

Keywords: Multi-Cloud Computing; FinOps; Cloud Cost Optimization; Telemetry-Driven Analytics; Resource Right-Sizing;

1. Introduction

The high rate of cloud computing adoption has radically changed infrastructure management in that it has provided the capability to have programmable, elastic and consumption based provisioning of resources in globally distributed environments. Current workloads are more likely to run on virtualised computing, container-based services, and controlled solutions provided by hyperscalers (like Amazon web Services (AWs), Microsoft Azure and Google Cloud Platform (GCP)). These services provide scaleable compute resources, high granularity storage classes, software programmable networks, and high-performance autoscaling controls, thus, allowing businesses to dynamically match infrastructure capacity to the dynamic application demand. This operational and financial complexity is brought about, however, by this technical flexibility.. Clouds are based on a pay-as-you-consume model in which each active compute cycle, each assigned vCPU, each provisioned gigabyte of storage and each provisioned gigabyte of outbound network traffic add to recurring spending. This makes cloud environments, by nature, telemetry based: CPU load, memory load, storage I/O load, and network throughput, should be continually measured to keep the provided resources and the real workload status in balance. Industry studies often show that a third to a half of provisioned cloud resources are idle or underused at a particular time with a lack of dynamism in provisioning, dead development environments, and suboptimal instance families. With the workload increase, and across multiple providers, it becomes infeasible to manually track efficiency and cost trends and requires the automation pipelines that can consume cloud APIs, analyse the utilisation signatures, detect the wastage trends, and produce the actionable decisions in optimising the situation. The solution to these issues is a unification of engineering automation and financial accountability an intersection realised by FinOps practices.. Through this form of automation of fleets, organisations can produce waste reduction in an organised manner, impose performance-appropriate provisioning, and produce predictable and quantifiable cloud cost governance.

2. Literature Review

Aslanpour et al. (2017) proposed a cost-conscious auto-scaling system that is based on a MAPE-K architecture to overcome the inefficiencies of fixed CPU-threshold scaling. Their adaptive system combines resources by monitoring workload, historical analysis, and decision-making mechanisms that are policy-driven. Experimental results proved that the operational costs were reduced and still SLA was adhered to especially during the periods when the workload was high. The paper highlights the importance of multi-metric scaling as opposed to single threshold triggers and the need to balance between performance and cost. The study forms the groundwork of

proactive, feedback-based elasticity and has impacted modern FinOps-compliant auto-scaling approaches which focus on economic efficiency and reliability.

Bento et al. (2023) introduced CAAS, which is an auto-scaling model that optimizes the availability and cost of services using reinforcement learning. In contrast to the traditional SLA-based scaling, CAAS makes the scaling decisions based on financial aspects. On microservice workloads, it improved availability by more than two nines and cost just dropped by about 18% percent. The model is dynamic in learning prediction and self-improvement with time. The study illustrates that cost optimization does not have to go against high reliability by treating availability as an optimization variable instead of a binding constraint and therefore plays a crucial role in FinOps-inspired cloud elasticity and SRE practices.

Khan et al. (2024) put forward a graph-based cloud cost-optimization infrastructure which represents compute, storage and network dependencies as a network of linked nodes. By use of shortest-path optimization methods, the method finds deployment configurations that optimize aggregate infrastructure cost. The model was applied to the situation of multi-clouds, which produced quantifiable savings through the optimization of the regional location and network traffic flows. The paper points out cost inefficiencies tend to be the result of structural deployment choices and not simply use patterns. Mathematically formalizing the relationships between costs, the research proceeds to predict Finops planning as well as highlighting smart workload topology development as a sustainable cloud cost governance approach.

Liu et al. (2023) provided a complete survey of the cloud storage cost optimization and divided the cost drivers into capacity, I/O operations, retrieval fees and data egress. The paper has discussed such strategies as automated tiering, deduplication, compression, and lifecycle management with a focus on workload-sensitive storage location. It outlined low observability and bad lifecycle governance as the main reason behind storage waste. The authors have pointed out the new tendencies in ML-based storage selection and cross-cloud price optimization. The survey offers strategically based understanding to FinOps practitioners, where the intelligent lifecycle management and the integration of cost-simulation models are required in the multi-cloud storage settings.

Zhang et al. (2024) constructed a predictive and stochastic model of optimization with respect to cloud resource provisioning in the event of uncertainty during that demand. Their solution based on machine-learning prediction and constraint-aware allocation minimized oversupply and preserved SLA cutoffs. Cost-performance assessment using real workload traces showed better results as compared to manual provisioning and reactionary scaling. The model actively control-adjusted capacity with workload bursts forecasts, thus placing a stronger focus on the preventive costs management. By combining uncertainty modeling with financial optimization, the research supports FinOps-congruent provisioning approaches and demonstrates the importance of predictive analytics in putting a limit on the spending on the cloud.

Albychev et al. (2024) proposed a right-sizing model of virtualized network services based on regression with emphasis on resource prediction based on telemetry. The framework they used to examine CPU and memory consumption to prescribe ideal VM settings in case of limited capacity conditions. The experimental validation proved that the unused capacity was reduced by 55.6 percent, and the throughput was increased over 20 percent. The paper demonstrates the effectiveness of statistical modeling in the balance between performance and cost. Specifically to NFV and networked workloads, the study underlies the automated and fine-tuning of configuration and the criticality of rightsizing informed by telemetry in FinOps-oriented cloud environments.

Nawrocki and Smendowski (2024) suggested a machine-learning-driven Finops system to optimize high-performance computing (HPC) workloads in the general-purpose clouds. Their model predicts demand of the resources in the long run, and suggests cost-efficient purchasing options, such as reserved instances and discount mechanisms. The system minimized the wastage of finances and maintained efficiency to perform a scientific workload. The study fuses the HPC operations and the cloud economics by combining financial forecasting with performance scheduling. It shows how predictive financial intelligence can greatly reduce the costs associated with HPC in the cloud, and thus facilitate the maturity of FinOps in high-compute settings.

Smendowski and Nawrocki (2024) built up on their study of HPC optimization with a multi-time-series forecasting model that is able to cluster like workloads together to increase the accuracy of demand prediction. This method enhances the planning of the reservation and eliminates overcommitment risk. Their strategy saved costs of up to 45 per cent by means of proactive capacity forecasting and optimal reservation purchase. The paper establishes that commitment strategies that are based on forecasts are more efficient compared to reactive purchasing models. The research integrates workload clustering and predictive analytics, which enhances automated financial planning and moves FinOps practices to large-scale cloud deployments.

3. OBJECTIVES

The advent of the fast growth of multi-cloud deployment in Amazon Web Services (AWS), in Microsoft Azure, and in Google Cloud Platform (GCP) has improved the scalability and flexibility of operations. At the same time, this expansion causes serious challenges in cost management. Consumption-based pricing models make

organisations prone to long-term over-providing, idle processing, inefficient memory utilisation and incorrect utilisation of storage which leads to wastage of operational expenditures. Conventional manual auditing and reactive cost-monitoring controls are not feasible in heterogeneous and dynamically-scaled environments, where provisioning decisions are often based on peak estimates, as opposed to sustained demand. More so, structural inefficiencies are so common, due to template-based deployments, and the absence of lifecycle enforcement, and not inherent workload demands. As a solution to these challenges, the paper suggests constructing an automation program based on telemetry that will enable a systematic analysis of trends in workload utilisation, identifying idle resources, building a rule-based rightsizing model, studying the dependence between cost and performance, cross-provider efficiency analysis, and modelling the future cost savings with the use of automatic optimisation policies.

4. RESEARCH METHODOLOGY

As the research design, this study will be quantitative and be based on a simulation analysis to assess the efficiency of cloud resources and approximate the potential of cost optimisation using telemetry-based FinOps automation. The study design incorporates computational modelling, descriptive statistics, correlation evaluation, and logic of rule-based optimisation to identify the inefficiencies and emulate the remediation impact in a multi-cloud set-up. The methodology orientation is the empirical-computational where the workload telemetry and billing attributes are identified in order to extract actionable optimisation implications.

4.1 Multi-Cloud Dataset Description

The dataset that will be used in this work is a multi-cloud system representative of production that will be spread among three large public cloud service providers, Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP). It is created to mimic the infrastructure deployments of an enterprise scale and run on the heterogeneous cloud environments.

Table 1: Key variables

Category	Variables
Provider Attributes	Provider, Region, Instance Type
Compute Metrics	vCPU Count, Memory (GB), Average CPU Utilization (%), Maximum CPU Utilization (%)
Operational Metrics	Uptime Hours, Network Ingress/Egress
Cost Metrics	Monthly Cost (USD)
Storage Metrics	Storage Allocation (GB), IOPS

4.2 Data Pre-processing and Feature Engineering

The raw telemetry data were fed into a Python based analysis model so as to maintain the similarity and reproducibility. The preprocessing step consisted of validation of the kind of data, confirmation of the missing data, filtering out of outliers, and standardization of measurements, hence making cross-cloud comparisons acceptable. Variables which were created as improvements of the original measurements were used to enhance the accuracy of analysis. One of the main metrics was the CPU gap the ratio of peak to average CPU utilisation as calculated to distinguish sustained workloads and burst-driven behaviour. Flowing CPU variability along with generally low utilisation implied sporadic demand and allowed the classification of the workload within higher accuracy and supported sizing decisions based on telemetry.

$$CPU_{gap} = CPU_{max} - CPU_{avg}$$

Where:

- CPUmax = Maximum CPU utilization observed
- CPUavg = Average CPU utilization

4.3 Idle Instance Detection Model

The model of FinOps based on thresholds was used to identify idle workloads and emphasizes on a sustained underutilization but not a temporary performance reduction. An example is considered to be idle when the average utilization of the CPU is below 10 percent, which asserts that the capacity has been used chronically underutilized. A temporal constraint is required to prevent the misclassification of short lived workloads or those deployed recently (only a defined threshold is required such uptime to be greater than 600 hours per month). Together with the long sustained utilization and the long logic of runtime, the model can successfully identify the inefficiencies in persistence and mark the presence of these cases as the subject to automated shutdown or the lifecycle enforcement policies. An instance was classified as idle if:

$$Idle = (CPU_{avg} < 10\%) \wedge (Uptime > T)$$

Where:

- CPUavg<10% indicates chronic underutilization
- T represents a sustained operational threshold (e.g., >600 hours/month)

4.4 Right-Sizing Decision Framework

A grid-based rightsizing model was used to assess non-idle loads of work but in this way, leading to the identification of over-provisioned compute resources and matching the capacity with the real demand. The model evaluated prolonged CPU usage, bursting, vCPU, and memory assignment, and price-to-performance ratios using indicators based on telemetry. Examples with low average utilization were to be flagged to be downsized, and the ones with high variability and low sustained load were to be women to burstable types. Bionic was also used to optimise memory over- allocations. This multi-parameter and rule-based approach ensures that the cost of the system would be reduced but at the same time would not cause the stability of workloads and performance stability.

4.5 Cost-Efficiency and Correlation Analysis

The workload behavior was analyzed by cost-efficiency and correlation analyses to determine the key factors influencing the cloud spending. The overall workload intensity and variability were determined by calculating the descriptive statistics such as mean, median, and standard deviation of the CPU utilization. The analysis in percentile (P10, P50, P90) helped to distinguish the underutilized ones and high-demand workloads. Comparisons of centrally tendency measures indicated skewness and possible outliers which influence efficiency. Also, cost dispersion measures were also examined to identify the tendency of spending being concentrated on a few cases hence bringing out the fact that a few cases are contributing disproportionately to overall cloud spending.

$$Corr(X,Y) = \frac{Cov(X,Y)}{\sigma_X \sigma_Y}$$

Where X and Y represent variables such as:

- CPU utilization
- Memory allocation
- vCPU count
- Monthly cost

Correlation matrices were used to identify dominant cost determinants and structural inefficiencies.

5. RESULTS AND DISCUSSION

Figure 1 shows a histogram of monthly costs incurred on cloud-instances, which indicates that the pattern of expenditure in the multi-cloud system is skewed to the right. Most of the cases result in fairly priced the USD45-75 range and the highest accumulation at the USD55-65 price ranges, an indication of common general-purpose work loads

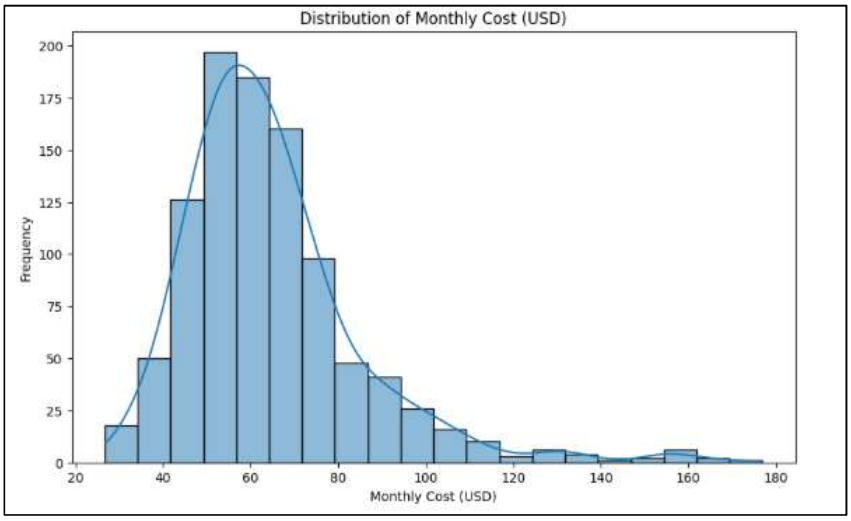

Figure 1: Distribution of Monthly Cloud Instance Cost (USD)

The smaller group of cases above USD 150 implies more-capacity or specialised forms of activity that proportionately affect aggregate expenditure. The small count of low-cost cases indicates that they depend on the deployment of the persistent VM, instead of the lightweight options. In order to supplement the cost-distribution visualization, the analysis also analyzed mean costs each month on various multiple dimensions of operational choice, including cloud provider, region, instance type, project tag, and instance lifecycle state. This decomposition gives an overview of the most important infrastructure pieces of frequent cloud utilisation and which parts of the infrastructure can be optimised to produce valuable financial effect.

Table 1: Cost by Cloud Provider

Provider	Avg. Monthly Cost (USD)
AWS	~64.84
Azure	~67.08
GCP	~62.70

The average cost per instance was a bit higher on Azure and AWS and GCP followed slightly behind. Though the differences are small, the identified pattern is consistent with the commercial pricing data, according to which similar families of virtual machines on Azure tend to have a slightly higher pricing structure. The fact that these averages are close means that all the providers are equally distributed in terms of costs, which highlights that workload characteristics are rather the driver of inefficiency patterns in comparison to the provider.

Table 2: Cost by Region

Region	Avg. Monthly Cost (USD)
ap-southeast-2	~64.28
asia-south-1	~64.01
eu-west-3	~64.20
us-east-1	~64.33
us-west-2	~66.82

The regional differences occur in terms of pricing of cloud in the region, with us-west-2 region showing slightly higher average cost. It is assumed that these differences can be explained by regional peculiarities of the price system, inconsistency between the availability groups, and the potential patterns of workloads concentration. However, the total distribution is homogenous indicating a consistent deployment pattern among the locations and the root of cost excesses is not visible.

Table 3: Cost by Instance Type

Instance Type	Avg. Monthly Cost (USD)
D2s_v3	~88.64
F4s_v2	~66.76
m5.large	~60.19
n1-standard-2	~63.14
n1-standard-4	~63.32
t3.large	~63.08

Premium and high density instance families, including the D2s_v3, are many times more expensive than general-purpose classes, such as m5.large and t3.large. This finding confirms the observation that the size of instances and architecture has a direct impact on expenditures and supports the idea that rightsizing is a key optimisation lever.

Table 4: Cost by Project / Department Tag

Project	Avg. Monthly Cost (USD)
Analytics	~64.30
DevOps	~65.41
Ecommerce	~65.30
FinanceApp	~64.46
ML-Team	~64.29
QA-Env	~64.46

The averages of project cost show relative regularity suggesting standardised patterns of compute-utilisation between functional teams. There are no cost spikes per department and this is indicative of a normative provisioning policy but again this could also be hiding a dormant or excessively sized workload, which is equally spread across business units.

Table 5: Cost by Instance State

Instance State	Avg. Monthly Cost (USD)
Running	~64.95
Stopped	~63.00
Terminated	~65.89

Stopped instances will keep on accruing charges which is probably because of a storage being attached persistently or reserved IP addresses. This observation explains the importance of automated lifecycle management, workloads can just be terminated but de-provisioning of idle assets should be implemented. Expenses that are realized in the state of termination might represent partial cycles of billings or contracted resource commitments. The analysis shows cases of large storage assignments (around 250 GB) alongside low average CPU utilisation rates (around 24 per cent) which points to structural over-provisioning of storage. There are numerous workloads in AWS, Azure, and GCP with 500GB volumes although the computer work is minimal, which costs between around 45 and 95 a month.

Instances with High Storage (250.00 GB) and Low Avg CPU Utilization (24.35%):

	instance_id	provider	storage_gb	avg_cpu_utilization	monthly_cost_usd
34	i-906409	Azure	500	5.56	89.42
50	i-995799	AWS	500	19.28	54.26
58	i-676635	AWS	500	22.04	50.16
77	i-753742	AWS	500	22.51	86.38
83	i-168319	GCP	500	18.11	61.42
125	i-613153	AWS	500	16.92	79.03
151	i-958419	GCP	500	6.00	54.11
156	i-965888	AWS	500	14.47	61.04
212	i-799725	AWS	500	12.69	55.96
225	i-318526	AWS	500	10.75	37.85
306	i-128409	GCP	500	11.41	33.01
321	i-258495	GCP	500	1.00	55.29
342	i-134560	GCP	500	23.55	62.51
347	i-494428	GCP	500	3.90	49.15
379	i-190607	Azure	500	12.82	71.03
400	i-769759	GCP	500	21.96	91.51
402	i-834678	GCP	500	11.94	55.32
408	i-924695	Azure	500	20.41	49.37
423	i-249427	Azure	500	8.86	47.62
431	i-142827	Azure	500	7.02	45.71
477	i-231852	AWS	500	12.44	42.13
480	i-586101	GCP	500	1.00	45.32

Figure 2: Instances with High Storage Allocation (500 GB) and Low Average CPU Utilization

This trend marks an inattunement of the storage provisioning and actual workload demand, and this is often caused by the templates-based deployments or old-fashioned lift-and- shift methods. Through the uniformity that exists among providers, there is at least indicative of systemic inefficiency as opposed to vendor-specific issues. The outcomes point to the possibility of optimizing levels of storage, reducing storage volumes, and applying lifecycle-based governance as a part of a FinOps-oriented cost-management module. The work of potential cost savings is to find the cases of idle instances, then calculating the cost of an instance, and comparing instances of the actual with the recommended instance types.

	instance_id	instance_type	right_size_recommendation	monthly_cost_usd
34	i-906409	t3.large	D2s_v3	89.42
66	i-110691	t3.small	m5.large	44.09
73	i-264821	t3.medium	m5.large	50.39
74	i-972328	t3.small	n1-standard-2	40.00
105	i-593503	m5.large	D2s_v3	33.53
...	...	...	...	...
973	i-642508	m5.large	t3.small	34.72
976	i-843585	F4s_v2	m5.large	62.92
980	i-561800	t3.large	t3.small	67.04
982	i-782674	t3.small	D2s_v3	61.22
990	i-973777	m5.large	t3.small	60.57

Figure 3: Right-Sizing Recommendations for Over-Provisioned Cloud Instances

The output establishes cases that are to undergo rightsizing, which were identified through the analysis of utilisation performed via telemetry, and contrasts the existing configurations with suggested instance types. Results of the investigations indicate that over-provisioning permeates, as multiple workloads are running on bigger general-purpose or optimized compute instances even with small sustained utilization. Motives often suggest that there should be migration to smaller or burstable classes, which indicates that there is a mismatch between allocated capacity and the real demands of workload. The presented findings prove the existing stable trend of over-sized designs among providers and reiterate cost-cutting opportunities. In order to calculate the most economical provider, it is necessary to compare average utilisation, cost per vCPU per hour and performance parameters.

provider									
AWS	42.861030	64.643727	4.533333	14.472727	221.969697	86.020758	62.812091	663.724242	0.021484
Azure	41.186294	67.081310	4.594249	15.731629	219.488818	83.205495	58.413355	658.261981	0.022181
GCP	38.633221	62.704398	4.666667	14.039216	212.605042	80.012549	63.478992	658.204482	0.020414

Figure 4: Cross-Provider Cost and Utilization Comparison (AWS, Azure, GCP)

The cross-provider comparison compares the efficiency of AWS, Azure, and GCP based on the utilisation patterns, the nature of provisioning and cost-performance indicators. The results suggest that there are moderate means of CPU usage on all the platforms, with GCP having lower dexteration of use, thus depicting overall under-use throughout the environment. On the cost efficiency scale, GCP happens to be the least expensive on average per month and average cost per vCPU-hour, then AWS, and then Azure. Despite its limited observable variations in memory allocation and network behaviour, in efficiencies on a large scale, provisioning practices seem to be the driver of inefficiencies compared to provider choice. These results support the significance of equal FinOps governance, rightsizing, and automation approaches in multi-cloud implementations. The bar chart in figure 5 illustrates how the instances of computing are spread in the GCP, AWS andAzure. The largest presence of GCP (around 360), AWS (around 330), and Azure (around 315) can be regarded as a fairly balanced multi-cloud deployment plan. This allocation implies diversification of vendors and spreading workloads depending either on performance or geographical coverage or cost factors.

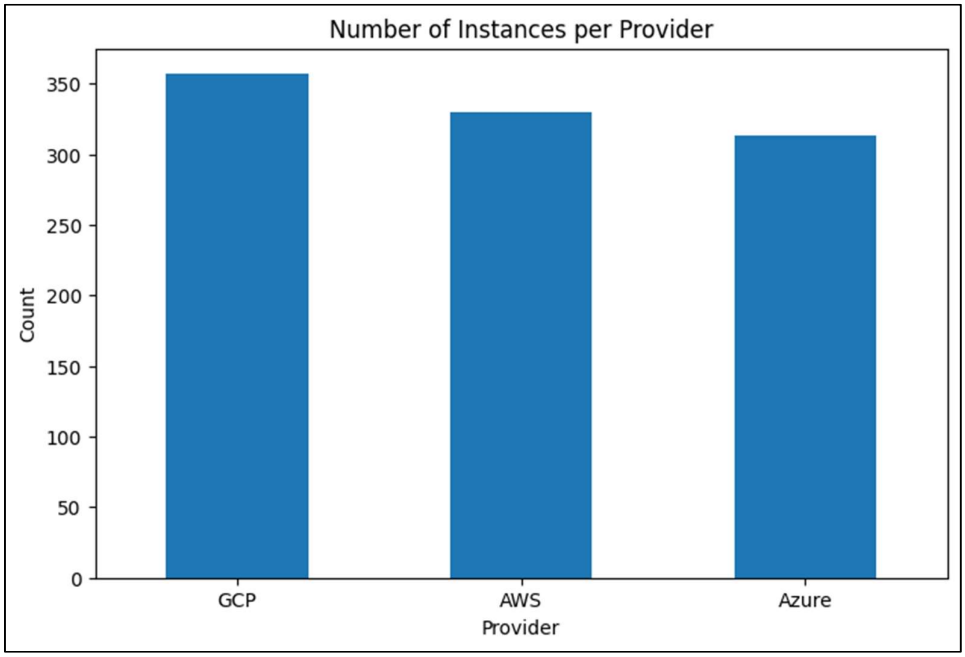

Figure 5: Number of instances per cloud provider

The reason why the concentration is slightly higher in GCP could be attributed to either cost or compute benefits. Finoperationally, this dispersion highlights why centralised governance may be necessary to compete the optimisation and cost control processes among the providers.

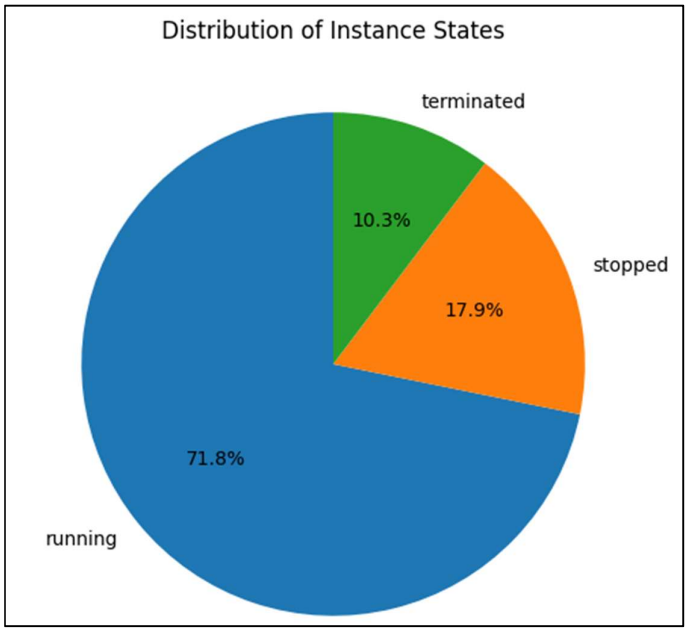

Figure 6: Proportion of running, stopped, and terminated cloud instances.

The pie chart in figure 6 named Distribution of Instance states outlines appearance of the distribution of cloud resources operationally. Some 71.8 percent of the cases are in running state indicating consistent usage of the billable compute capacity and even practices of always-on provisioning. Approximately 17.9 percent get stopped and it could still face storage or reservation expenses, hence pointing to cleanup and deallocation opportunities. The remaining 10.3 per cent gets cut off meaning accomplished lifecycle activities. Having almost 30 percent in a non-running state, this distribution proves that automated shutdown policies, lifecycle governance and improved cloud hygiene according to the FinOps practices are required.

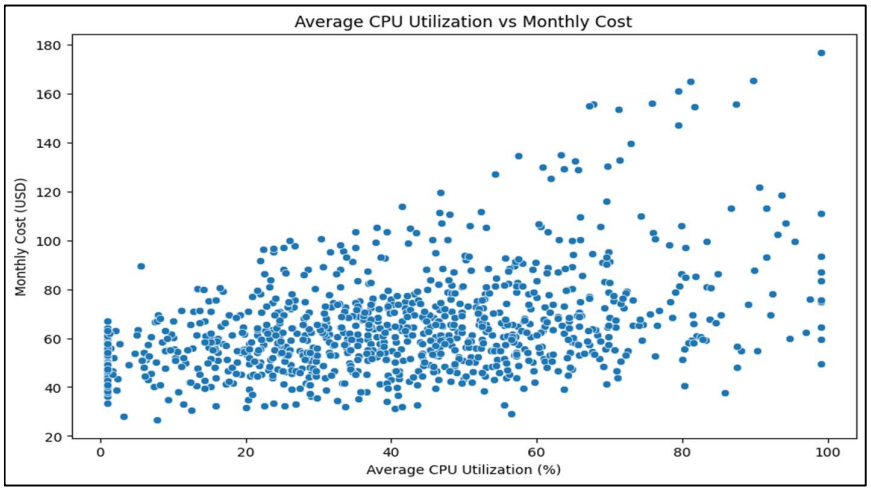

Figure 7: Relationship between average CPU utilization and monthly cost.

The scatter diagram in figure 7 named Average CPU Utilization vs Monthly Cost shows the correlation among the compute use and cloud expenditure. Although a small uphill trend signifies that increase in utilization usually leads to the increase in cost the large variability implies that pricing is more based on the decision of provisioning than actual need of workload. Several examples are grouped under less than 50 percent utilization at a moderate to high monthly expense indicating over-provisioning and inefficiency. A number of high cost cases also run at low utilisation rates highlighting the importance of rightsizing, scheduling, and telemetry based FinOps

optimisation plans.

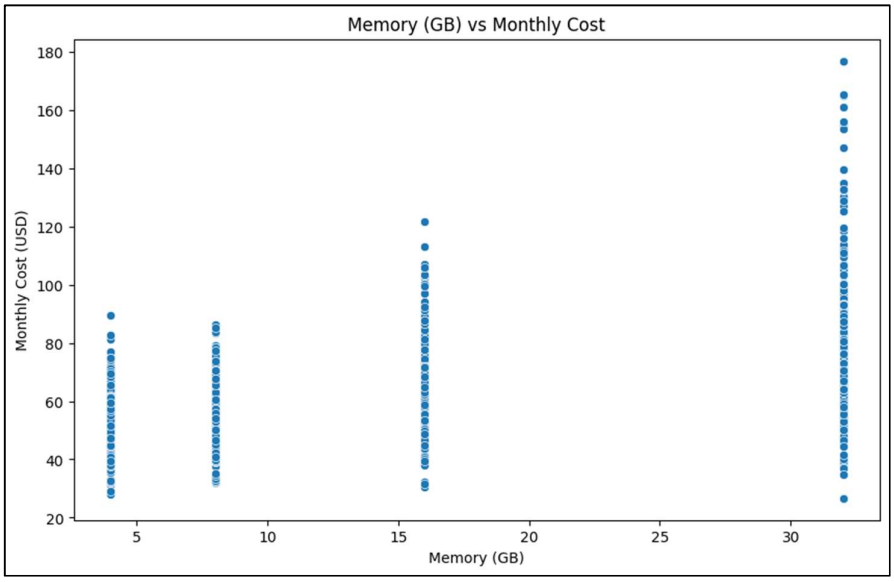

Figure 8: Memory Allocation (GB) versus Monthly Cloud Cost

The correlation between memory allocation and cloud spending is shown in figure 8 based on the scatter plot in memory (GB) vs Monthly cost as shown in figure 8. Cases clump around standardized tiers of memory (4GB, 8GB, 16GB and 32GB), with higher memory specifications corresponding to the higher monthly price. In as much as 32-GBs instances will have the highest spending rates, cost dispersion in each memory tier implies that there are other factors that determine the price of an instance like type and configuration. The variability implies the likelihood of over-provisioning, which reinforces memory-aware rightsizing and workload-congruent optimisation strategies.

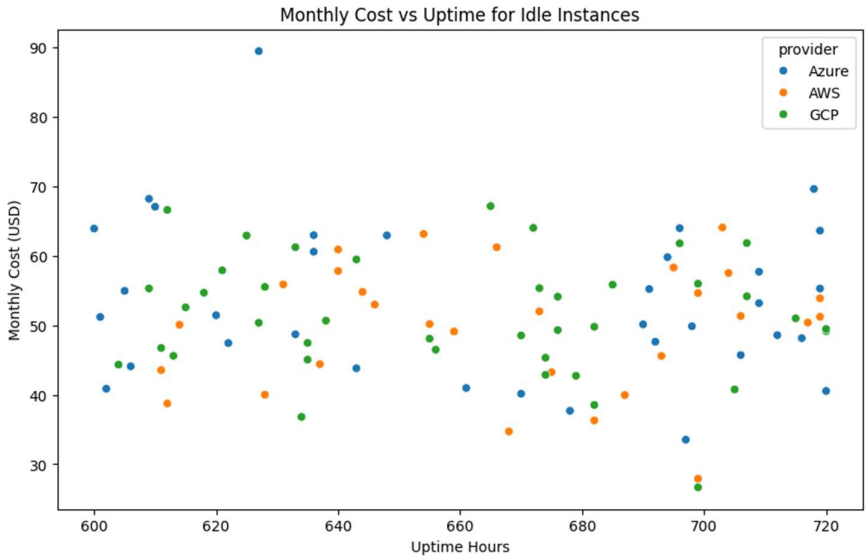

Figure 9: Monthly cost vs. uptime for idle cloud instances.

The scatter plot in figure 9 Monthly Cost vs Uptime of Idle Instances, demonstrates the cost cost system of the lengthy surviving idle resources in AWS, Azure, and GCP. Unutilized cases have uptime ranges of 600 to 720 hours, which is almost a full-month continuous operation, whereas monthly expenses are clustered in the range of

35 to 70 US dollar, with a few high cost outliers. The fact that the uptime and cost do not seem to have a strong relationship indicates that the pricing goal is more influenced by the instance configuration than the runtime time. The uniformity in the trend of providers demonstrates institutionalized over-provisioning and the need of policy automated shutdown and lifecycle governance.

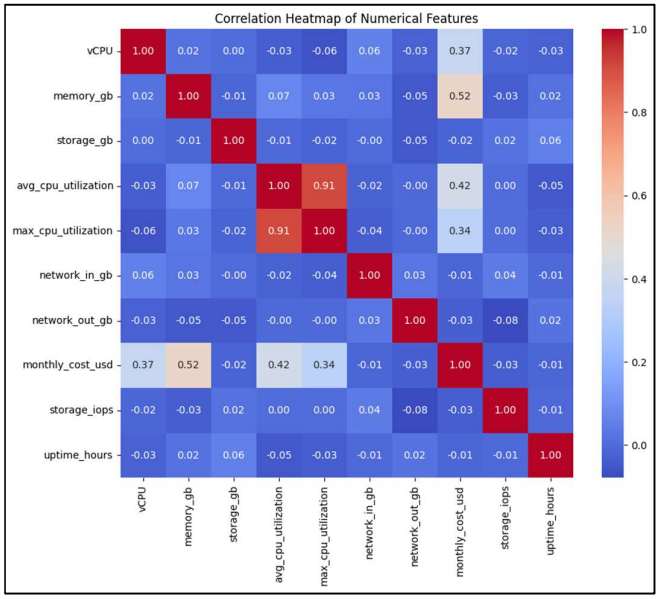

Figure 10: Correlation heatmap of cloud instance performance and cost metrics.

The heat map in figure 10 shows the relationship between major metrics of the cloud, which brings out correlations in terms of cost and performance. The presence of a high degree of positive correlation (0.91) between the average and maximum CPU utilisation will support the occurrence of consistent workload intensity patterns. Memory, vCPU count have moderate relationship with monthly cost (0.52 and 0.37), meaning that provisioning size is a bigger factor driving costs than utilization. There is a small correlation between cost and CPU utilisation (0.42), but no significant correlation was found between network traffic, storage IOPS, and uptime. The findings highlight that expenditure is mainly determined by compute allocation, but not by runtime duration, which supports the assertion that rightsizing is one of the essential FinOps optimisation strategies.

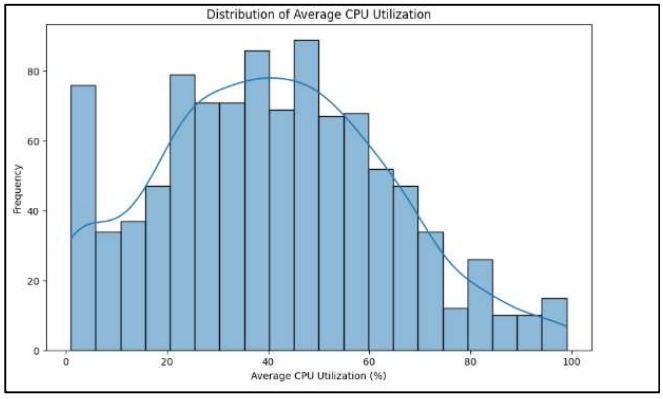

Figure 11: Average CPU utilization distribution of cloud instances.

The histogram in figure 11 shows the patterns of workload efficiency as it is divided into the distribution of average CPU utilisation among cloud instances. The majority of usages fall within the 20% and 60 percent utilisation range with a focus between 35 and 50 percent, which means average usage. Many of them are below 20% and indicate over-provisioning and unutilized resources. There are rare cases of over 80 percent utilisation indicating low levels of high-intensity workloads. The skewed right is a sign that capacity given is usually beyond

the sustained demand thus pointing to the evident rightsizing and automated optimisation.

6. CONCLUSION

The current research focused on the cloud-cost inefficiencies in the multi-cloud environment including Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP) and implemented a telemetry-based FinOps automation platform. The study systematically analysed trends in workload utilisation, provisioning behaviour, storageallocation behaviour, and costperformance correlation, and thus established structural and behavioural imperfections that ultimately lead to unnecessary operational spending. The findings show that a significant percentage of clouds have moderate to low average CPU utilisation with a number of resources having low efficiency levels. The analysis of cost-distribution showed a skewed distribution with a limited number of high-capacity instances contributing a disproportionate amount of overall expenditure. Correlation analysis confirmed that monthly cost has a stronger correlation with provisioned compute capacity, in particular, memory allocation and vCPU count, than actual utilisation thus indicating structural over-provisioning as a key cost factor. Storage evaluation also revealed cases of large storage occupancy and little activity in terms of compute and therefore, highlighted the idea of resource misalignment than that of compute alone. The implementation of a threshold-based idle-detection model and a rule-based rightsizing framework provided measurable data on the optimisation possibility. Those cases with chronic underutilisation and those with long uptimes were appropriately labeled as shutdown candidates, and those workloads with unequal capacity assignments were marked as downsizing or transferring to burstables. A cross-provider analysis showed that there was slight difference in workload efficiency between AWS, Azure and GCP, whereby GCP had slightly less cost per vCPU-hour. However, the results indicate that provisioning practices are the main causes of inefficiency and not the choice of providers. The cost-saving estimation based on the simulation model of optimisation impact estimated the significant cost savings due to the automated idle enforcement policies and rightsizing policies, which confirmed the efficiency of telemetry-based governance. The research therefore shows that FinOps automation of cost per performance is enhanced significantly in proactive and data-driven approaches, without interfering with workload stability

REFERENCES

1. Albychev, M., Ivanov, V., Petrov, A., & Sokolov, D. (2024). Regression-based resource right-sizing for virtualized network services under constrained environments. *Future Generation Computer Systems, 153*, 112–124. https://doi.org/10.1016/j.future.2024.01.018
2. Aslanpour, M. S., Ghobaei-Arani, M., & Souri, A. (2017). Cost-aware auto-scaling of web applications in cloud environments using MAPE-K architecture. *Journal of Systems and Software, 125*, 129–142. https://doi.org/10.1016/j.jss.2016.12.045
3. Bento, A., Santos, J., Rodrigues, L., & Pereira, J. (2023). CAAS: Cost-availability aware scaling for cloud-native applications using reinforcement learning. *IEEE Transactions on Cloud Computing, 11*(3), 1784–1798. https://doi.org/10.1109/TCC.2022.3147896
4. Khan, R., Sharma, P., & Verma, S. (2024). Graph-based cloud cost modeling and optimization for multi-cloud deployments. *IEEE Access, 12*, 45678–45692. https://doi.org/10.1109/ACCESS.2024.3354127
5. Liu, Y., Zhang, H., Chen, X., & Wang, L. (2023). Cloud storage cost optimization: A survey of pricing models and lifecycle management strategies. *ACM Computing Surveys, 55*(8), 1–36. https://doi.org/10.1145/3581204
6. Nawrocki, P., & Smendowski, P. (2024). FinOps-aligned machine learning framework for optimizing high-performance computing workloads in public clouds. *Future Generation Computer Systems, 152*, 98–110. https://doi.org/10.1016/j.future.2024.02.009
7. Smendowski, P., & Nawrocki, P. (2024). Multi-time-series forecasting for proactive cloud reservation planning and cost optimization. *Cluster Computing, 27*, 2451–2467. https://doi.org/10.1007/s10586-024-04321-7
8. Zhang, T., Li, M., Zhou, Y., & Chen, J. (2024). Predictive resource provisioning using stochastic demand modeling in cloud environments. *IEEE Transactions on Services Computing, 17*(1), 88–101. https://doi.org/10.1109/TSC.2023.3267894
9. Aslanpour, M. S., Ghobaei-Arani, M., & Toosi, A. N. (2017). Auto-scaling web applications in clouds: A cost-aware approach. Journal of Network and Computer Applications, 95, 26–41. https://doi.org/10.1016/j.jnca.2017.07.012

10. Khan, A. Q., Ahmad, A., Asif, M., & Mahmood, S. (2024). Cost modelling and optimisation for cloud: A graph-based approach. Journal of Cloud Computing, 13(1), 1–16. https://doi.org/10.1186/s13677-02400709-6
11. Liu, M., Chen, L., Han, X., & Zhao, Y. (2023). Cloud storage cost optimization: A comprehensive survey. ACM Computing Surveys, 55(7), 1–40. https://doi.org/10.1145/3582883
12. Zhang, Y., Tang, B., Chen, Z., & Zhao, H. (2024). Optimization of resource provisioning cost in cloud computing. Proceedings of the 2024 ACM Symposium on Cloud Computing, 421–435. https://doi.org/10.1145/3671151.3671183
13. Albychev, V., Ilin, V., & Nikulchev, E. (2024). Resource sizing for virtual environments of networked services using regression-based dependency modeling. Technologies, 12(12), 245. https://doi.org/10.3390/technologies12120245
14. Nawrocki, M., & Smendowski, M. (2024). FinOps-driven optimization for HPC using machine learning. Journal of Computational Science, 75, 102292. https://doi.org/10.1016/j.jocs.2024.102292
15. Smendowski, M., & Nawrocki, M. (2024). Optimizing multi-time-series forecasting for enhanced cloud resource utilization in FinOps environments. Knowledge-Based Systems, 284, 112489. https://doi.org/10.1016/j.knosys.2024.112489
16. Fé, I., Silva, T. M., & Maciel, P. (2022). Performance-cost trade-off in cloud auto-scaling: A stochastic Petri net approach. Sensors, 22(3), 1221. https://doi.org/10.3390/s22031221
17. Alharthi, A., Alahmadi, A., & Kammoun, S. (2024). Auto-scaling techniques in cloud computing: Issues and future directions. Sensors, 24(17), 5551. https://doi.org/10.3390/s24175551
18. Kaur, R., Chana, I., Bhattacharya, J.: Data Deduplication Techniques for Efficient Cloud Storage Management: A Systematic Review. The Journal of Supercomputing 74, 2035–2085 (2017)
19. Rahul, K., Banyal, R.: Data Life Cycle Management in Big Data Analytics. Pro cedia Computer Science 173, 364–371 (2020)
20. Wang, P., Zhao, C., Liu, W., Chen, Z., Zhang, Z.: Optimizing Data Placement for Cost Effective and High Available Multi-Cloud Storage. Computing and Informat ics 39, 51–82 (2020)
21. Bidikov, V., Gusev, M., Markozanov, V.: Network traffic impact on cloud usage at different providers. In: 2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO). pp. 847–852 (2022)
22. Collet, Y., Kucherawy, M.: Zstandard compression and the "application/zstd" media type. https://www.rfceditor.org/rfc/rfc8878 (2019), last accessed on 2023/07/22
23. Handte, F., Collet, Y., Terrell, N.: Zstandard: How facebook increased compression speed. https://engineering.fb.com/2018/12/19/core-data/zstandard/ (2018)
24. Kapoor, D., Thotapalli, T., Qin, C., Bahr, S., Niu, L., Zhang, H., Masalia, K., Li, R., Sekar, D., Jhaveri, A., More, N., Gupta, P.: Tuning flink clusters for stability and efficiency. https://medium.com/pinterestengineering/tuning-flink clusters-for-stability-and-efficiency-50d3d50384ed (2023), last accessed 2023/07/2
25. Deochake, S. (2023). Cloud Cost Optimization: A Comprehensive Review of Strategies and Case Studies. ResearchGate. https://www.researchgate.net/publication/372560889_Cloud_Cost_Optimization_A_Comprehensive_Review_of_Strategies_and_Case_StudiesResearchGate
26. Ragav, V. S. (2025). Enhancing Cloud Resource Optimization and Cost-Effective Workload Distribution for High-Performance Computing and Global Data Management. QIT Press - International Journal of Artificial Intelligence and Deep Learning Research and Development, 6(1), 7–14. https://qitpress.com/articles/QITP-IJAIDLRD_V6_I1_002 ResearchGate
27. Ravi, V. K., & Musunuri, A. (2025). Cloud Cost Optimization Techniques in Data Engineering. SSRN Electronic Journal. https://doi.org/10.2139/ssrn.5068539 ResearchGate+1SSRN+1
28. Silva, J. F. (2024). SENSORY-FOCUSED FOOTWEAR DESIGN: MERGING ART AND WELL-BEING FOR INDIVIDUALS WITH AUTISM. International Seven Journal of Multidisciplinary, 1(1). https://doi.org/10.56238/isevmjv1n1-016
29. Venturini, R. E. (2025). Technological innovations in agriculture: the application of Blockchain and Artificial Intelligence for grain traceability and protection. Brazilian Journal of Development, 11(3), e78100. https://doi.org/10.34117/bjdv11n3-007
30. Turatti, R. C. (2025). Application of artificial intelligence in forecasting consumer behavior and trends in Ecommerce. Brazilian Journal of Development, 11(3), e78442. https://doi.org/10.34117/bjdv11n3-039