BCIIT e-Journal 2026

Journal of Innovations in Computer Science and Trends in IT, (October 2026), Vol 3, Issue2, ISSN 3048-4707

Cloud Cost Management: A Practical Model for Cluster Cost Allocation

Manju Arora
Deptt of Information Technology
Jagan Institute of Management Studies, Rohini, Delhi
manju.arora@jimsindia.org

Amit Kumar Jha
Department of Information Technology
Jagannath University, Bahadurgarh, Haryana
amitjhadelhi16@gmail.com

Jestin N Daniel
Department of Information Technology
Jagannath University, Bahadurgarh, Haryana
jestindaniel0@gmail.com

Abstract: Cloud computing has fundamentally reshaped how organizations consume and pay for computational resources, shifting infrastructure from capital expenditure toward a flexible, usage-based model. However, as deployments grow—particularly in shared cluster environments— accurately attributing costs to individual users or workload becomes surprisingly difficult. Most existing billing strategies price resources at the virtual machine or reserved-instance level, which rarely reflects what tenants consume. This paper proposes a practical cluster-level cost allocation model that computes charges proportionally based on three resource dimensions: storage, compute (CPU), and memory. A normalized utilization metric drives a dynamic pod-generation mechanism that adjusts the billed workload in real time, while a scaling factor brings the result in line with market pricing. Experiments conducted on a synthetic but realistic cloud workload dataset show that the proposed model reduces billing inaccuracy and minimizes idle-resource waste compared to traditional VM-based pricing. The approach is lightweight, transparent, and straightforward enough to be integrated into existing cluster management platforms such as Kubernetes.

Keyword: Cloud Cost Management, Cluster Cost Allocation, Resource Utilization, Pod Scaling, Kubernetes, Multi-Tenant Billing, FinOps

I. INTRODUCTION

Cloud computing has been playing an important role in the modern digital infrastructure, making the organizations scalable, providing flexible computing resources without having the overhead of managing the physical hardware. Some of the examples are Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP), all of them operate on Pay-as-go pricing model. In this cost is connected to the actual usage theoretically. In practically there is cost difference between the usage and the cost paid by the organization.

Journal of Innovations in Computer Science and Trends in IT, (October 2026), Vol 3, Issue2, ISSN 3048-4707

Adopting the cluster-based deployment by the organization's best example is Kubernetes faces a common challenge: sharing the same underlying infrastructure or more than two users and platform doesn't provide the user's per usage cost breakdowns. As we know, Traditional costing mechanism are bases on the types of VM or reserved-instance hours, which leads to a huge-expensive billing which ignores how the CPU, storage and memory are being distributed among the users. This leads to some users subsidizing effectively other's and leads to little incentive avoiding over-provisions.

The impact on finance is significant. As per the surveys of Industries show that organization wastes 30-40 percent on clouds, when most of the time sitting idle or not utilized resource. A right or fair, allocation driven usage model doesn't only provide or improve the billing fairness but also impacts on the natural incentives for the user's workload as per the right size. This paper bridges the gap by implanting the cost allocation cluster level framework the decreases the overall cluster cost into one user contribution based on the normal utilization of resource. This model has been built on the traditional costing formula and has been extended with a dynamic pod generation component which shows the real-time demand. The contributions of work are threefold: (i) a mathematical clear formulation for both the traditional and proposed model cost; (ii) a based utilization scaling pod mechanism governed by an adjustable threshold; (iii) validation experimental on a cloud synthetic workload dataset which covers a realistic patterns usage of range.

The paper's remainder is organized as follows: Section II reviews the work related to cloud cost management. Section III depicts the gap of research targets on this paper. Section IV presents the proposed model in a detailed manner. Section V description on dataset. Section VI experimental methodology explained. Section VII analysis and results. Section VIII Describing the implications and limitations. Section IX Conclusion of the paper.

II. LITERATURE REVIEW

A. Cloud Cost Monitoring and FinOps

A body of broad work focuses on the cost cloud visibility -helps the organization understand about the money where it goes. Some of the tools such as AWS, Cost Explorer and Azure Cost

Journal of Innovations in Computer Science and Trends in IT, (October 2026), Vol 3, Issue2, ISSN 3048-4707

management all provide some dashboards and detect anomalies. The FinOps emerges discipline formalize the process for the cross-functional accountability of cost [1]. These are valuable efforts, primarily an observation, they notify the team what was spent, explains not how to allocate the fairly within the shared environment

B. Resource Allocation and Auto-Scaling

Cost-efficient resource allocation-based research frequently employs the techniques on Machine learning to predict the workload demand and to adjust accordingly provisioning. Forecasting model bases on LSTM have been used to deduct overprovisioning by coming high traffics before they can occur [2]. Agents on Reinforcement learning agents have also been founded for the scaling of the dynamic decision which balances cost and service level objectives. All these methods help in the utilization of cluster level but don't affect the result of the cost distributed among the users.

C. Multi-Tenant and Cluster Level Billing

Cost allocation in some prior work has been examined in multi-tenant cloud environment. Sharing proportional schemes, got inspired by earlier work in utility computing, cluster cost has been divided according to each user's resource usage rather than the actual consumption [3]. Showback and Chargeback framework has been used in enterprise IT takes a same reservation-based technique. The limitation is resources that are reserved and resources that are used can diverge significantly, commonly in elastic and container-based deployment.

D. Container Orchestration and Kubernetes Economics

As we know, Kubernetes have become the standard factor for orchestration of containers, studies by researchers have begun about economic properties. Impact on pod scheduling policies have been examined in the studies which show cluster utilization and cost and several projects that are open source such as Kubercost attempt to provide per-namespace or cost on per team attribution. As most of the tools depend on the list of cost or simple equal splits and do not incorporate a utilization of formal based model of the kind proposed here [4].

Journal of Innovations in Computer Science and Trends in IT, (October 2026), Vol 3, Issue2, ISSN 3048-4707

E. Carbon Aware and Energy-Efficient Scheduling

A different but related flow of research puts light on reducing the footprint of environment clouds workloads by moving them towards the regions of low carbon or off-peak periods [5]. While one of the important goals is sustainability, carbon awareness approaches a different form of objectives from allocation of cost and does not provide the mechanism for distributing expenses among users.

III. RESEARCH GAP

The above literature review reveals several blind consistent spots which help and motivate the current ongoing work. Firstly, research on cloud computing majorly focuses on optimization or monitoring at the level of clusters, leaving behind the questions of inter cluster cost attribution is unsolved largely. Second is the existing billing model maybe based on VM hours, reversed instances or proportional splits namespace level- doesn't capture the multi-dimensional nature of consumption of resources (storage, CPU, memory). Third that is lack of transparent mathematical frameworks that are non-specialist can inspect, audit and adapt. Fourth demand dynamic fluctuation- where a resource of a user footprint varies significantly over time- are rarely mixed into allocation models in principled way.

Taken together, these gaps point to the need for a lightweight, formula-driven model that (a) considers multiple resource dimensions, (b) responds to real-time utilization, (c) is transparent enough to build user trust, and (d) can be implemented without specialized hardware or extensive infrastructure changes. The model proposed in this paper is designed to satisfy these criteria.

IV. PROPOSED MODEL

A. Traditional Pricing Baseline

Before introducing the proposed model, it is helpful to formalize the traditional approach. Under conventional VMbased pricing, the cost assigned to a user is a linear combination of raw resource consumption:

$$C_{trad} = C_s \cdot S_u + C_c \cdot P_u + C_m \cdot M_u \quad (1)$$

Journal of Innovations in Computer Science and Trends in IT, (October 2026), Vol 3, Issue2, ISSN 3048-4707

where Cs, Cc, and Cm are the unit prices for storage, compute (CPU), and memory, respectively, and Su, Pu, and Mu are the absolute quantities consumed by user u. The model is simple to implement but has a key weakness: it prices resources in isolation without any sense of how much of the cluster's total capacity a user is occupying. A user consuming 50 GB of storage on a 100 GB cluster is billed very differently from one consuming 50 GB on a 500 GB cluster, even if both users place a similar proportional burden on the infrastructure.

B. Resource Normalization

The proposed model starts by normalizing each dimension of resources against the total cluster's available capacity, converting absolute quantities into share of relative.:

$$S_{norm} = S_u / S_t \tag{2}$$

$$P_{norm} = P_u / P_t \tag{3}$$

$$M_{norm} = M_u / M_t \tag{4}$$

In this St, Pt and Mt denotes the cluster total capacity for storage, memory, CPU respectively. In this model normalization ensures that the model has invariant effect on the absolute scale of the cluster, which leads to making it equally applied to a small department cluster and production of large deployment.

C. Composite Utilization Score

U in single utilization computes as weighted sum of the sum of the normalized resource shares:

$$U = W_s \cdot S_{norm} + W_c \cdot P_{norm} + W_m \cdot M_{norm} \tag{5}$$

Ws, Wc, Wm which are weights should satisfy Ws+Wc+Wm=1. Its value should reflect the structure of cost of the specific cluster: in storage workloads that are heavy, for example data lakes. Ws will dominant whereas the intense compute workload for example Machine learning training would need a higher Wc. The experiment that is reported in this, 3 weight configurations are explored to assess the sensitivity, as per the description in Section VI.

Journal of Innovations in Computer Science and Trends in IT, (October 2026), Vol 3, Issue2, ISSN 3048-4707

D. Dynamic Pod Generation

In this model when the utilization surpasses the predefined threshold θ, model can generate the additional virtual bill units - normally referred as pods- to account for more. The number of additional pods is:

$$N_p = [\max (0, U - \theta) / \theta] \quad (6)$$

In this threshold θ acts like a baseline level utilization under the level in which no extra charges is applied. When user usage surpasses the θ, each additional increase of θ one more pod is generated. The θ involves choice of trade-Off below value (e.g 0.6) activates the pod generation earlier and makes the billing more sensitive according to the moderate usage, while large values (e.g 0.75) reserve the fees for the genuine heavy users. Based on the empirical testing on synthetic dataset, values in range $\theta \in [0.60, 0.75]$ able to produce stable and understandable results. In Cluster the admin should maintain θ to match their organization's acceptable utilization baseline.

E. Final Proposed Cost

In this model the final cost combines a scaled version of the traditional baseline including pod generation cost.

$$C_{prop} = \alpha \cdot C_{trad} + C_p \cdot N_p \quad (7)$$

Where $\alpha \in (0,1]$ scaling factor that adjust base cost which align actual cloud provider rates or internal charge policies and Cp is the per pod charge. The α serves 2 purposes: allows the model to generate against real billing data, and it stops double counting when pod already captures a high portion of high usage cost. Where the traditional formula already reflects contract pricing α can be set close to 1 also can be used to apply for a discount or markup.

Journal of Innovations in Computer Science and Trends in IT, (October 2026), Vol 3, Issue2, ISSN 3048-4707

F. Variable Summary

TABLE I. MODEL VARIABLE DEFINITIONS

Variable	Description
Cs, Cc, Cm	Unit prices for storage, CPU, and memory
Su, Pu, Mu	User consumption of storage, CPU, and memory
St, Pt, Mt	Total cluster capacity for each resource
Snorm, Pnorm, Mnorm	Normalized utilization fractions (0–1)
Ws, Wc, Wm	Dimension weights (must sum to 1)
U	Composite utilization score
θ	Utilization threshold (recommended 0.60–0.75)
Np	Number of dynamically generated billing pods
α	Base-cost scaling factor (≤ 1)
Cp	Per-pod billing charge
Ctrad	Traditional (baseline) cost estimate
Cprop	Final proposed model cost

V. DATASET DESCRIPTION

In this paper the experiment used on synthetic but realistic cloud workload dataset constructed to cover the data range of used patterns which are commonly observed in the production of Kubernetes clusters. Data that is synthetic chosen for 2 purposes First, real world clusters billing the data from the providers of the commercial which is rarely and proprietary made publicly available in the sufficient details for research. Second, generation in synthetic allows controls over workload – including hard cases such as near zero usage, sudden rise and sustained higher load time that may be presented in any single real deployment.

Journal of Innovations in Computer Science and Trends in IT, (October 2026), Vol 3, Issue2, ISSN 3048-4707

That dataset is available at the following link:

https://1drv.ms/x/c/82ba2223a8d71f64/IQD8TA7kw1JYR4fGcw3AZtV8Adc5ITdX2s4cBfm2xSjzegk?e=DCqF1u

This dataset contains records for the multiple used users operating within the shared cluster over a defined observation window. Each window includes the records of storage consumed (Su), CPU scores used (Pu), memory allocation (Mu), and the correspond cluster total figures the (St, Pt, Mt) from which the normalization can be performed in direct manner. Unit cost values (Cs, Cc, Cm) are set at those levels consistent with mid-tier pricing of AWS at the time of writing, the model itself is provider-agnostic.

TABLE II. DATASET SUMMARY STATISTICS

Parameter	Value
Total user records	500
Observation window	30 days
Storage range (S_u)	5 GB – 480 GB
CPU range (P_u)	0.5 – 48 vCPUs
Memory range (M_u)	1 GB – 192 GB
Cluster storage capacity (S_t)	500 GB
Cluster CPU capacity (P_t)	64 vCPUs
Cluster memory capacity (M_t)	256 GB
Users exceeding $\theta = 0.70$	~27%

Journal of Innovations in Computer Science and Trends in IT, (October 2026), Vol 3, Issue2, ISSN 3048-4707

VI. METHODOLOGY

A. Experimental Setup

All experiments were implemented using MS-Excel for mathematical computation. No specialized online server is needed. It will work on a normal laptop also with full dataset.

B. Baseline Configuration

The Proposed cost was set as Cs=$0.023 for every 1GB of data, Cc=$0.048 for 1 virtual CPU for one hour and Cm = $0.006 for 1 GB per hour. The per pod-charge Cp was $0.15 for every pod running during the observation time as the scaling factor is (0.85) 85% capacity, (0.90) 90% capacity, (1.00) 100% capacity.

C. Weight Configurations

Three different methodologies used to represent different cluster types:

- Storage-dominant: Ws = 0.6, Wc = 0.2, Wm = 0.2
- Balanced: Ws = 0.33, Wc = 0.33, Wm = 0.34
- Compute-dominant: Ws = 0.2, Wc = 0.6, Wm = 0.2

D. Threshold Sensitivity Analysis

For each weight category, four values were examined: $\theta = \{0.55, 0.60, 0.70, 0.75$. For each (weight, threshold) pair, Np was examined for every record of the user and then validated to produce total number of pods count and cost distributions.

E. Evaluation Metrics

The proposed model was analyzed over the traditional base model on three things: billing, accuracy and efficiency as the new model is more accurate which uses fewer resources as it helps in reducing billing errors and wasted cost.

Journal of Innovations in Computer Science and Trends in IT, (October 2026), Vol 3, Issue2, ISSN 3048-4707

VII. RESULTS AND ANALYSIS

A. Cost Comparison: Traditional vs. Proposed

We tested 500 records. The result shows the new model is much better. On average, it is more accurate in calculating cost than old traditional model. The old model had a big problem; it consistently overcharges the user which doesn't use much of the service. This happened because old system uses a flat rate that doesn't consider how small users use this.

TABLE III. MODEL PERFORMANCE COMPARISON

Metric	Traditional Model	Proposed Model
Billing accuracy (MAPE)	23.6%	5.2%
Idle-resource penalty	31.1%	12.4%
Cost variance (std dev)	$14.72	$6.38
Users billed above actual share	61%	18%

B. Threshold Sensitivity

This figure generally illustrates how many users generating at least additional pod changes with Θ . The user should pay extra fees based on their usage. like Θ = 0.55 shows that 42% of users exceeds threshold, which is aggressive for environments and Θ = 0.75 only 19% users pay extra which shows a balance and it is most recommendable.

TABLE IV. POD GENERATION RATE BY THRESHOLD

Threshold (θ)	Users with Np > 0	Avg. Np (when > 0)	Total Cluster Pods
0.55	42%	2.1	441
0.60	34%	1.8	306
0.70	27%	1.5	203
0.75	19%	1.3	124

Journal of Innovations in Computer Science and Trends in IT, (October 2026), Vol 3, Issue2, ISSN 3048-4707

C. Weight Configuration Impact

Computational dominance weight categorization produces the billing variance, which reflects the CPU consumption more than storage and memory. The balanced approach gives the most balanced and stable cost distribution across the users and therefore recommended for the general clusters. Organizations such as Big Data/ML companies which benefit a lot from tuning weights.

TABLE V. COST VARIANCE BY WEIGHT CONFIGURATION

Weight Scheme	Mean Cost ($)	Std Dev ($)	MAPE (%)
Storage-dominant	8.14	5.91	7.3
Balanced	8.22	6.38	5.2
Compute-dominant	8.09	8.17	9.1

D. Scaling Factor Effect

Reducing the scaling factor (α) from 1 to 0.85 decreases the total revenue of the cluster by approximately 7% while maintaining the cost of the user, which is heavy users still billed more as compared to low resource driven users. This suggests that α can be used as a Leveler to give discounted benefits without disrupting the fairness of the allocation.

VIII. DISCUSSION

A. Practical Implications

The results indicate that the proposed model can meaningfully reduce billing inaccuracy in shared cluster environments with a very modest implementation burden. The formulas are simple enough to be evaluated in a spreadsheet, and the only inputs required—per-user resource consumption and cluster-total capacity—are already collected by standard monitoring tools such as Prometheus or the Kubernetes Metrics Server. Organizations using Kubecost or similar tooling could, in principle,

replace the default proportional-split logic with the proposed model at the reporting layer without touching cluster configuration.

B. Limitations

Several limitations are acknowledged. First, the model was calculated on the synthetic data. While the dataset was designed to be realistic, it may capture pattern seen in production - such highly irregular profiles or workloads. Validations on real cluster billing from different organizations would strengths these validations especially.

Second, the model visualizes all the time periods within the observation window. In peak hours, resources often cost more and time aware extension of the model would be more accurate for environment with variable pricing.

Third, the choice of weights (Wc, Ws,Wm) and threshold require domain knowledge and more information for administrators which are new for the cluster environments. While the default problem is tested across different configurations, a data-driven procedure using historical billing - records would reduce the load of the practitioners. Lastly, the pod-generation mechanism provides the step function in cost rather than a smooth curve. This means that users near the threshold boundary may notice reasonable bills according to the usage levels, which is unfair

C. Comparing with Existing models and approaches

Compared to reservation-based systems, our model tracks actual consumption of a user and prevents people from cheating the system. It basically gives a fair idea as it uses a specific “threshold” to encourage a better resource use. While some more complex AI models’ approach is more advanced, our model is basically easier to understand and use, which is usually better for daily work.

IX. CONCLUSION

This paper basically gives a practical cluster-level cost allocation model for their shared cloud environments by observing the traditional VM-based pricing and cost allocation models and it cuds the ideal resource optimization in a very effective manner. The model has also some limitations -

mostly it works on a synthetic dataset and uses step function for increment and rest in future we will surely address these things also and make it like a real - world deployment model and more efficient cost allocation model. In future, it will also work with certain parameters like network bandwidth, GPU allocation, and pricing in natural direction.

Overall, this model provides a transparent and easy way for companies to allocate cloud cost accurately.

REFERENCES

[1] J. R. Storment and M. Fuller, Cloud FinOps: Collaborative, Real-Time Cloud Financial Management. O'Reilly Media, 2019.

[2] H. Guo et al., "LSTM-based workload forecasting for cloud auto-scaling," Journal of Cloud Computing, vol. 10, no. 3, pp. 1-18, 2021.

[3] A. Gulati et al., "Cloud-scale resource management: Challenges and techniques," in Proc. USENIX HotCloud, 2012.

[4] S. Burns, Kubernetes: Up and Running, 3rd ed. O'Reilly Media, 2022.

[5] L. Shi et al., "Carbon-aware task scheduling for cloud computing data centers," IEEE Transactions on Sustainable Computing, vol. 7, no. 2, pp. 345-358, 2022.

[6] A. Beloglazov and R. Buyya, "Energy-efficient resource management in virtualized cloud data centers," in Proc. IEEE/ACM CCGrid, 2010.

[7] R. N. Calheiros et al., "CloudSim: A toolkit for modeling and simulation of cloud computing environments," Software: Practice and Experience, vol. 41, no. 1, pp. 23-50, 2011.

[8] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: State-of-the-art and research challenges," Journal of Internet Services and Applications, vol. 1, pp. 7-18, 2010.

[9] T. Mastelic et al., "Cloud computing: Survey on energy efficiency," ACM Computing Surveys, vol. 47, no. 2, pp. 1-36, 2015.

[10] M. Armbrust et al., "A view of cloud computing," Communications of the ACM, vol. 53, no. 4, pp. 50-58, 2010.