FinOps in Cloud-Native Near Real-Time Serverless Streaming Solutions

Dimitar Mileski
Innovation Dooel
Skopje, North Macedonia

Marjan Gusev
Faculty of Computer Science and Engineering
Ss. Cyril and Methodius University in Skopje, North Macedonia

***Abstract*—FinOps is a novel discipline in cloud computing and technology management that aims to optimize an organization's cloud spending, enhance financial resource management, and promote collaboration between technology and finance teams for efficient cloud resource utilization and cost control. The adoption of cloud-native and serverless architectures has revolutionized the way organizations design and deploy their near-real-time streaming solutions. These solutions are vital for various applications, including data analytics, monitoring, and content delivery. However, understanding the cost implications of scaling these solutions to accommodate varying numbers of concurrent users remains a challenge. This paper presents a cost analysis for a cloud-native near real-time serverless streaming solution, considering both streaming and idle times of the system. The cost of cloud services is heavily influenced by efficient processing management and the selection of suitable services. This decision can result in substantial cost differences across various cloud providers, varying from expensive to cost-efficient options. This paper explores the critical factors impacting costs, highlighting differences in cost levels based on several factors.**

***Index Terms*—finops, pricing model, near real-time, cloud-native, serverless, streaming, public cloud, cloud computing**

I. Introduction

FinOps is a relatively new area [1] in cloud computing management to optimize and effectively manage cloud spending and financial resources. The primary goal is to ensure that cloud resources are utilized efficiently, controlling the costs while bridging the gap between technology and finance teams. FinOps encourages collaboration among IT, finance, and other departments, addressing the financial implications of technology decisions. Various tools and automation are being built for more efficient financial management, and it is an ongoing process of monitoring, optimizing, and improving cloud spending. Some organizations even offer FinOps certifications and training programs to empower professionals in this evolving discipline. Overall, cloud cost management is effectively addressed by FinOps in the context of a dynamic and rapidly evolving technology landscape, enabling organizations to maximize the benefits of cloud computing while maintaining cost control [2].

Near Real-Time systems operate with minimal delays, usually within a few seconds or fractions of a second. The acceptable delay varies depending on the specific context and application. For instance, a few seconds of delay in video streaming is not a burden, while in financial trading, even a few milliseconds can be significant. This implies that near real-time processing heavily depends on the situation and user needs.

Concurrent users in Near Real-Time systems refer to the number of individuals or processes that simultaneously interact with, utilizing the resources and services at any given moment. Cloud-native technologies enable organizations to build and run scalable applications in various cloud environments by leveraging the benefits of cloud computing, focusing on containers, service meshes, microservices, and immutable infrastructure elements [3], [4].

Serverless streaming in cloud computing enables developers to process Near Real-Time data streams, like sensor data and event streams, without the need to manage server infrastructure. Related benefits include simplified deployment, automatic management, and resource scaling, without the burden of server provisioning and system maintenance [5].

The pricing model and cost estimation in cloud computing are complex due to numerous variables. The industry remains divided on the cost efficiency of cloud computing, with debates occurring online [6] [7] [8] [9]. These variables include service and deployment models, resource usage, data transfer costs, geographical locations, instance types, scaling, licensing, data storage, compliance, support, etc. Businesses need to analyze these factors to manage cloud costs effectively.

The price of implementing and scaling these cloud-native streaming solutions is a pivotal determinant, often making a substantial difference between successful solution deployment into production and whether a business can attain profitability. In this paper, we address the cost estimation frameworks for small and large numbers of concurrent users and analyze resource usage scenarios in the cloud. To exemplify these principles, we implemented a cloud-native, near real-time streaming solution in the context of Google Cloud, leveraging Google Cloud Pricing Calculator for precise cost estimations.

The paper is structured as follows. The related work contextualizes our research in Section II. Section III-A explains our Pricing Model and Cost Estimation approach. The system architecture is described in Section III-B and the evaluation methodology in Section III-D to validate our cost estimation method. Results are presented in Section IV) and discussed in Section V) explaining what our research means. Finally, Section VI summarizes our conclusions, highlighting the importance of cost estimation in cloud-native near real-time streaming solutions.

This is accepted version. Published version DOI: 10.1109/TELFOR59449.2023.10372626

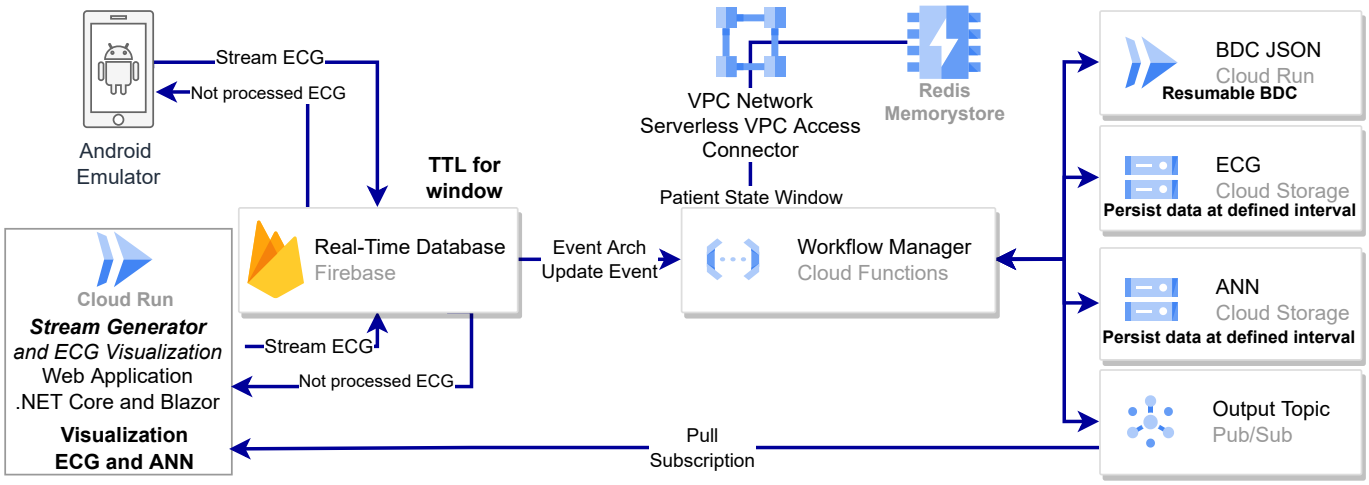

Fig. 1: System Architecture

II. Related Work

The concept of Financial Operations (FinOps) [1] introduces principles around enabling teams to make informed decisions about cloud spending, balancing cost, quality, and speed while measuring cloud spending against relevant business metrics (unit economics), reflecting the growing importance of efficient cloud cost management and collaborative decision-making in today's cloud-centric organizations. Integrating cloud monitoring technologies with FinOps methodologies optimizes cloud costs and emphasizes the importance of data management, tool integration, and effective team communication in achieving cost optimization [10].

More details on suitable tools include cost-related monitoring in specific environments, the definition of user requirements for cost monitoring and analysis, the development of custom dashboards, graphs, KPIs, and metrics, and identification of primary cost contributors and cost-generating factors [11]. A consistent focus on cost optimization remains imperative, and regardless of the chosen approach, to ensure efficient utilization of cloud resources and cost-effectiveness [12] adopting FinOPs practices and integrating multi-cloud billing monitoring tools.

Third-party tools and APIs significantly contribute to the field of cost management in the Google Cloud Platform (GCP), such as Alerteam API for cost monitoring, Neoxia's client anomaly detection, Stime, with a powerful tool to detect high costs and take prompt action. This will empower stakeholders to make informed decisions, proactively manage their budgets, and avoid potential budget overruns [13]. SmartCMP platform specializes in cloud cost analysis and optimization by offering financial insights, multidimensional cost analysis, customizable operational strategies, real-time risk monitoring, and security enhancements [14].

TABLE I: Service Pricing Model

Service	Pricing Model and Included Factors
Firebase Realtime Database	Data Storage, Data Transfer, Number of Connections
Cloud Memorystore for Redis	Memory Capacity
Cloud Pub/Sub	Published Messages, Subscribed Messages
Cloud Storage	Data Storage, Data Transfer Out
Cloud Functions	Function Invocations, Execution Duration
Compute Engine	Instance Type, Usage Duration, Network Usage
Cloud Run	Request Count, CPU/Memory Allocation
Networking	Data Transfer, Load Balancing, VPN Usage
Artifact Registry	Storage, Data Transfer
Cloud Logging	Log Ingestion, Data Retention
Cloud Build	Build Minutes, Storage

III. Methods

A. Pricing Model and Cost Estimation

Major cloud providers offer similar tools, such as the AWS Pricing Calculator by Amazon Web Services and the Azure Pricing Calculator by Microsoft Azure. In this paper, we use the Google Cloud pricing model to analyze various components, including computing, storage, networking, and specialized services, each with its pricing structure. The Google Cloud Pricing Calculator allows specifying usage requirements with real-time cost estimations for resource allocation. Table I provides a detailed breakdown of the service pricing model and the specific factors influencing the costs of Google Cloud services utilized within the system architecture depicted in Figure 1, aiding organizations in effective cost estimation and management.

B. System Architecture

Fig. 1 represents system architecture for the cloud-native, near real-time serverless streaming solution designed to process and visualize electrocardiogram (ECG) data. The central

components include the Stream Generator for data generation, and the ECG Visualization Web Application built using .NET Core and Blazor for user-friendly data visualization. Firebase Real-Time Database supports real-time data streaming, Google Cloud Pub/Sub "Pull Subscription" retrieves data, and "Output Topic" handles data publication. The Workflow Manager orchestrates tasks, while Cloud Functions execute specific functions. The "Patient State Window" manages patient-related data. Event management is facilitated by "Event Arch" and "Update Event," and unprocessed data is queued. Redis Memorystore ensures data availability, and a Virtual Private Cloud (VPC) network maintains security. Data flows through various stages, with "BDC" representing the beat detection and classification. Google Cloud Run offers scalability for specific functions, and data storage is ensured through "ECG Cloud Storage" and "ANN Cloud Storage."

C. Experiments

We conduct two experiments:

- **E1** an application where a single user continuously transfers one second of ECG data, streaming this data every second and
- **E2** non-streaming solution without any execution (an idle day)

The goal is to compare cost implications and resource usage in these experiments and gain insights into the financial and resource dynamics specific to the high-frequency transmission of short-duration ECG data, facilitating an assessment of the cost-effectiveness and resource efficiency.

Distinguishing between "days with streaming" and "idle days" with no user activity will be realized by analyzing the system performance under different usage scenarios.

D. Evaluation Methodology

The evaluation is performed by calculating the "cost ($) per concurrent user per day" metric to provide how efficiently our streaming solution manages expenses for each user actively using our platform daily.

Besides evaluating the cost-effectiveness based on user engagement, we account for the impact of idle times on cloud resource usage, ensuring that our streaming solution remains financially efficient and responsive to user demands.

IV. Results

Results for the E1 experiment are presented in Fig. 2, where a single user streams content continuously for 24 hours, displaying the cost ($) incurred by different cloud services and showing the cumulative cost for the entire day. Additionally, it provides the cost portion that each cloud service contributes to the total daily expense, offering a comprehensive view of cost distribution in our streaming solution.

Fig. 3 illustrates data for the E2 experiment for renting resources on an idle day lasting 24 hours without streaming activity. Besides the cost ($) incurred by various cloud services during this idle period, it shows the cumulative cost.

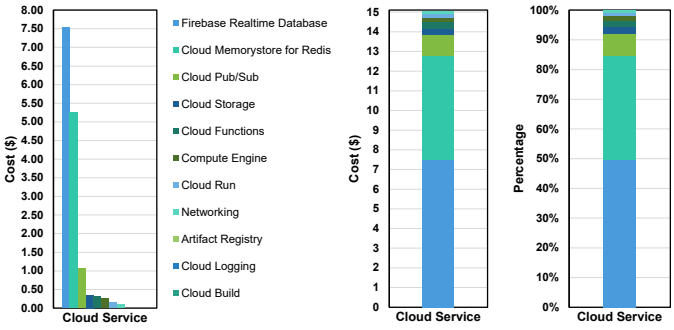

Fig. 2: Cost analysis of the E1 experiment, one user streaming continuously for 24 hours

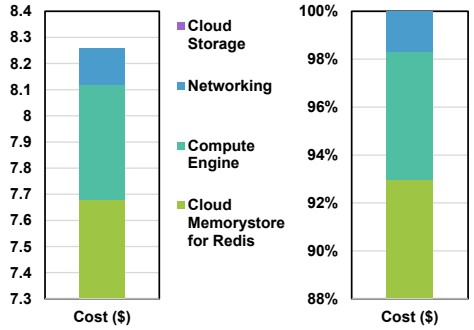

Fig. 3: Cost analysis of the E2 experiment, on an idle day (24 hours) without streaming

A comparison between the cost implications and resource usage for E1 and E2 experiments is presented in Fig. 4 with details on the financial and resource dynamics specific to evaluate the cost-effectiveness and resource efficiency of the streaming solution.

V. Discussion

In the E1 experiment, it is observed that Firebase Realtime Database and Google Cloud Memorystore for Redis services contribute to 85% of the overall 'Cost ($)'. This observation implies that there is significant room for cost optimization within the cloud services of Firebase Realtime Database and Google Cloud Memorystore for Redis.

During the idle day (24 hours) with no streaming activity, the 'Cost ($)' is primarily associated with cloud memory storage for Redis, accounting for 93% of the total cost. Compute engine costs contribute 6% of the overall expenditure, and networking expenses make up 1%.

The Firebase Realtime Database service incurs a significant cost during streaming days due to the egress (data download) operations associated with continuous user engagement. The Google Cloud Memorystore for Redis service makes most of the costs in E2, mainly because of its hourly pricing model.

For instance, to set up an 8 GB Basic Tier cloud instance (M2 capacity tier), the hourly cost in the Iowa region would be approximately $0.22. When calculated over a month, this hourly rate accumulates to around $160.60 (calculated as $0.22 multiplied by 730 hours). Therefore, the predominant cost

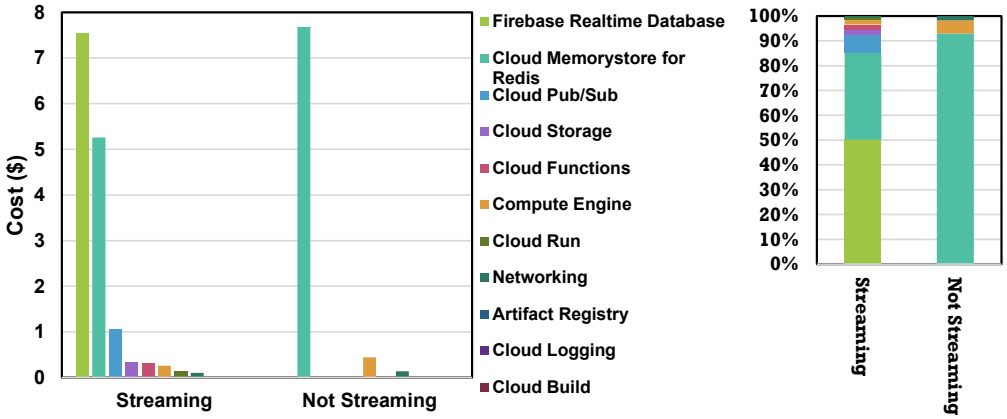

Fig. 4: Streaming vs. Not Streaming

during idle periods is influenced by the hourly pricing model of Google Cloud Memorystore for Redis.

FinOps involves strategic management of cloud expenses, beginning with assessing the current spending patterns. Committed use discounts (CUDs) play one more role in cost optimization. This involves maximizing the benefits of CUDs to reduce cloud expenses. Cost estimation tools such as the Google Cloud Pricing Calculator, AWS, and Azure resources are vital in this process, allowing for accurate forecasting and budgeting.

Finally, effective cost management includes proactive measures such as budget alerts, usually sent via email. They are triggered at predetermined thresholds, typically at 50%, 90%, and 100% of the overall budget utilization. They serve as an early warning system to ensure that cloud spending remains in line with the allocated budgets, promoting responsible financial operations within the organization.

VI. Conclusion

Our focus in this study was the cost analysis of cloud-native near real-time serverless streaming solutions, which have become indispensable for various applications. Our experiment, comparing the cost implications and resource usage between continuous ECG data streaming and idle periods, has shed light on the specific financial and resource dynamics associated with high-frequency, short-duration data transmission.

One of the key takeaways from our study is the significant role played by Firebase Realtime Database and Google Cloud Memorystore for Redis services in driving costs. On streaming days, Firebase Realtime Database costs are primarily attributed to data download operations due to continuous user engagement. In contrast, on idle days, Google Cloud Memorystore for Redis becomes the dominant cost factor, mainly due to its hourly pricing model.

Our paper also emphasizes the importance of FinOps in managing cloud expenses, starting to evaluate current spending patterns and proceeding to optimization recommendations. Key recommendations include FinOps Score, Peer Benchmarking, Committed Use Discounts (CUDs), Cost Estimation Tools, and Proactive Budget Alerts.

Future research involves real workloads, concurrency, and statistical distributions to enhance the accuracy of cost estimation methods and explore alternatives among cloud-native services that contribute to higher overall expenses. Implementing FinOps Score, Peer Benchmarking, Committed Use Discounts (CUDs), Cost Estimation Tools, and Proactive Budget Alerts will evaluate other similar Cloud-Native solutions.

References

[1] J. Storment and M. Fuller, *Cloud FinOps*. " O'Reilly Media, Inc.", 2023.

[2] J. Bryant, "Driving into the cloud: What is finops?" *ITNOW*, vol. 64, no. 3, pp. 54–55, 2022.

[3] "Cloud Native Technology — glossary.cncf.io," https://glossary.cncf.io/cloud-native-tech/, [Accessed 30-09-2023].

[4] "Cloud Native Apps — glossary.cncf.io," https://glossary.cncf.io/cloud-native-apps/, [Accessed 30-09-2023].

[5] K. Konstantoudakis, D. Breitgand, A. Doumanoglou, N. Zioulis, A. Weit, K. Christaki, P. Drakoulis, E. Christakis, D. Zarpalas, and P. Daras, "Serverless streaming for emerging media: towards 5g network-driven cost optimization: A real-time adaptive streaming faas service for small-session-oriented immersive media," *Multimedia Tools and Applications*, pp. 1–40, 2022.

[6] "Why we're leaving the cloud — world.hey.com," https://world.hey.com/dhh/why-we-re-leaving-the-cloud-654b47e0, [Accessed 01-10-2023].

[7] "We have left the cloud — world.hey.com," https://world.hey.com/dhh/we-have-left-the-cloud-251760fb, [Accessed 01-10-2023].

[8] "DHH Left the Cloud. You Shouldn't. — cloudfix.com," https://cloudfix.com/podcast/dhh-left-the-cloud-you-shouldnt/, [Accessed 01-10-2023].

[9] D. Linthicum, "Even with repatriation cost savings, the value of cloud computing is still strong — infoworld.com," https://www.infoworld.com/article/3707248/even-with-repatriation-cost-savings-the-value-of-cloud-computing-is-still-strong.html, [Accessed 01-10-2023].

[10] P. Singh and V. Khatri, "A study on the finops framework for sustainable cloud engineering," *Elementary Education Online*, vol. 20, no. 3, pp. 4874–4874, 2023.

[11] P. Suppala, "Finops in saas platform within hybrid, multi-cloud, multi-tenant, multi-region environments," 2022.

[12] L. Mei, "Cost optimization in cloud costs with finops and multi-cloud billing monitoring tool," 2023.

[13] S. Ait Chikh, "Finops: Monitoring and controlling gcp costs," 2023.

[14] F. Li, G. Wu, J. Lu, M. Jin, H. An, and J. Lin, "Smartcmp: A cloud cost optimization governance practice of smart cloud management platform," in *2022 IEEE 7th International Conference on Smart Cloud (SmartCloud)*. IEEE, 2022, pp. 171–176.