Towards Cloud Storage Tier Optimization with Rule-based Classification

Akif Quddus Khan[1], Nikolay Nikolov[2], Mihhail Matskin[3], Radu Prodan[4], Christoph Bussler[5], Dumitru Roman[2,6], and Ahmet Soylu[6]

[1] Norwegian University of Science and Technology – NTNU, Gjøvik, Norway
akif.q.khan@ntnu.no
[2] SINTEF AS, Oslo, Norway
[3] KTH Royal Institute of Technology, Stockholm, Sweden
[4] University of Klagenfurt, Klagefurt, Austria
[5] Robert Bosch LLC, CA, USA
[6] OsloMet – Oslo Metropolitan University, Oslo, Norway

Abstract. Cloud storage adoption has increased over the years as more and more data has been produced with particularly high demand for fast processing and low latency. To meet the users' demands and to provide a cost-effective solution, cloud service providers (CSPs) have offered tiered storage; however, keeping the data in one tier is not a cost-effective approach. Hence, several two-tiered approaches have been developed to classify storage objects into the most suitable tier. In this respect, this paper explores a rule-based classification approach to optimize cloud storage cost by migrating data between different storage tiers. Instead of two, four distinct storage tiers are considered, including premium, hot, cold, and archive. The viability and potential of the approach are demonstrated by comparing cost savings achieved when data was moved between tiers versus when it remained static. The results indicate that the proposed approach has the potential to significantly reduce cloud storage cost, thereby providing valuable insights for organizations seeking to optimize their cloud storage strategies. Finally, the limitations of the proposed approach are discussed along with the potential directions for future work, particularly the use of game theory to incorporate a feedback loop to extend and improve the proposed approach accordingly.

Keywords: Storage tiers, cloud, optimization, StaaS, cloud storage

1 Introduction

Cloud computing, in general, and cloud storage, in particular, have experienced exponential growth in recent years [21, 15, 20]. Organizations have increasingly embraced cloud services to meet their computing needs. According to Gartner, 85% of enterprises are expected to adopt a cloud-first approach by 2025 [19]. The use of cloud storage, i.e., Storage-as-a-Service (StaaS) [11], instead of local storage, has the potential to provide more flexibility in terms of scalability, fault tolerance, and availability. Cloud storage systems (e.g., Amazon S3, Azure Blob

Storage, Google Cloud Storage) offer very large storage with high fault tolerance, addressing several big data-related storage concerns [24]. When it comes to object storage services, leading cloud service providers (CSPs) such as Microsoft Azure, Google Cloud, and Amazon S3, offer four different storage tier options and pricing policies tailored to their specific data storage and access requirements. This presents an opportunity for users to optimize their StaaS cost. For example, Google Cloud Storage provides not only hot and cold storage tiers but also premium and archive tiers. The pricing structure varies across these tiers, with the hot tier offering lower access prices but higher storage cost, while the cold tier offers higher access prices but lower storage costs. This means that for data objects with infrequent access, storing them in the cold tier can result in lower expenses compared to the hot tier. As a result, StaaS users can strategically migrate their data, i.e., can do storage tier optimization from the hot tier to the cold tier when access demands decrease, reducing the overall cost.

Storage tier optimization is the process of organizing data into different tiers based on its usage and performance requirements [2]. This can help improve storage performance and efficiency by ensuring that the most frequently accessed data is stored on the fastest media, while less frequently accessed data can be stored on slower, less expensive media. There are many different ways to implement storage tiering. One common method is to use a storage array that has multiple tiers of storage media, such as high-performance flash storage, mid-range spinning disk drives, and low-cost nearline or offline storage. The storage array can then automatically move data between tiers based on its usage patterns [1]. Another common method of storage tiering is to use a software-defined storage solution [8]. These solutions typically provide a more flexible and scalable approach to storage tiering than traditional storage arrays. Software-defined storage solutions can also be used to tier data across multiple physical storage locations, such as on-premise and cloud storage.

In this paper, we focus on moving data between different tiers at a single location. Storage tier optimization can be a complex process, but it can offer significant benefits in terms of performance, efficiency, and cost savings. By planning and implementing a storage tiering strategy, organizations can improve the performance of their storage infrastructure and reduce their storage costs. To this end, in this paper, we explore storage tier optimization for cost-effective data storage using a rule-based classification approach that takes into account four storage tiers instead of just two, is lightweight, does not require intense computing resources, is platform-independent, and is fast. We propose a set of rules for calculating a score or priority score and define a threshold to classify each object stored in cloud storage into premium, hot, cold, or archive tiers. We demonstrate the viability and potential of the proposed approach against a synthetic dataset of 1TB by getting a significant reduction in storage cost. We discuss the limitations of the proposed approach and provide directions for improvement, particularly through expanding the proposal with the use of game theory, to incorporate a feedback loop in the process of storage object classification.

The rest of the paper is structured as follows. Section 2 provides an overview of cloud storage cost elements, while Section 3 presents the rule-based classification approach. Section 4 discusses the results and limitations and proposes the use of game theory for storage object classification. Section 5 provides a summary and discussion of related works. Finally, Section 6 concludes the paper and presents future work.

2 Cloud Storage Cost

The five major elements of cloud storage cost include: 1) data storage; 2) network usage; 3) transaction; 4) data retrieval; and 5) data replication/migration [5]. Table 1 shows the actual prices of different cost elements of cloud storage by using Google Cloud[1,2] as an example.

Table 1: Cost of data storage by Google Cloud in a single region, Europe - Warsaw (europe-central2) - data collected on 12 May 2023.

Cost Element	**Premium**	**Hot**	**Cold**	**Archive**
Official term	Standard	Nearline	Coldline	Archive
Storage cost ($\GB\month)	0.023	0.013	0.006	0.0025
GET Request ($ per 1,000)	0.0004	0.001	0.01	0.05
PUT Request ($ per 1,000)	0.005	0.01	0.02	0.05
Data Retrieval ($\GB)	0	0.01	0.02	0.05
Network Usage ($\GB)	0	0.01	0.02	0.05
Minimum Duration(days)	None	30	90	365
Latency		Low[a]		
Durability		99.999999999%[b]		
Availability	Multi-region: >99.99%	99.95%	99.95%	99.95%
	Dual-regions: >99.99%	99.95%	99.95%	99.95%
	Regions: 99.99%	99.9%	99.9%	99.9%

[a] Time to first byte typically tens of milliseconds.
[b] https://cloud.google.com/blog/products/storage-data-transfer/understanding-cloud-storage-11-9s-durability-target

2.1 Storage Cost

Storage cost refers to the cost of storing data in the cloud. It is charged on a per-GB-per-month basis. Each storage tier has different pricing. It also depends on the amount of data being stored. Some CSPs offer block-rate pricing, i.e., the larger the amount of data, the lower the unit costs are [14]. For example, there is a certain cost for data between 0 and 50 TB, and then for some tiers, it might

[1] https://cloud.google.com/storage/pricing
[2] https://cloud.google.com/storage/docs/storage-classes

be cheaper for over 50 TB of data. However, in this paper, we do not take that into account when calculating cost estimates.

2.2 Network Usage Cost

The quantity of data read from or sent between the buckets is known as network consumption or network usage. Data transmitted by cloud storage through egress is reflected in the HTTP response headers. Hence, the term network usage cost is defined as the cost of bandwidth out of the cloud storage server. It is charged on a per-GB basis. In addition to that, network cost also vary based on the amount of data transferred, as it offers different slabs for different amounts of data. The higher the amount of data transferred, the cheaper the cost will be.

2.3 Transaction Cost

Transaction cost refers to the costs for managing, monitoring, and controlling a transaction when reading or writing data to cloud storage [16]. When it comes to data storage, cloud storage providers charge for the amount of data transferred over the network and the number of operations it takes. Transaction costs deal with the number of operations. These costs are associated with managing, monitoring, and controlling a transaction when reading or writing data to cloud storage.

2.4 Data Retrieval

Data retrieval fees refer to the charges incurred when retrieving or accessing data from a storage system or service. In various cloud storage or object storage platforms, data retrieval fees may apply when retrieving stored files or information. These fees are typically associated with the data transfer or bandwidth used during the retrieval process. Data access frequency in this context is of importance when considering the impact of data retrieval on cost.

2.5 Migration Cost

Different CSPs provide the capability to migrate data objects between tiers throughout their lifecycles, presenting a valuable opportunity for cost optimization. The migration process involves retrieving the complete object from the source tier and subsequently submitting a PUT request to the destination tier to inform it of the impending object. As such, the data migration operation is subject to expenses associated with data retrieval, calculated based on the object size in the source tier, as well as expenses associated with the PUT request in the destination tier.

3 Rule-based Classification

The term rule-based classification can be used to refer to any classification scheme that makes use of IF-THEN rules for class prediction [23]. In this method, we define rules that assign each object to a storage tier based on specific criteria, such as the frequency of access, the size of the data, and the age of the stored object. For example, we define a rule that assigns objects that are accessed frequently to a high-performance storage tier and those that are accessed less frequently to a lower-performance storage tier. The following are some general rules that are used to determine which characteristics are appropriate for each tier:

1. **Premium tier:** This tier should be used for data with the highest frequency of access, such as data that is accessed continuously or near-continuously and requires the highest levels of performance and durability. For example, mission-critical databases or high-performance computing workloads.
2. **Hot tier:** This tier should be used for data with frequent access patterns, such as data that is accessed daily or weekly and requires fast access times. For example, this might include frequently accessed files, frequently used application data, or logs that require analysis on a regular basis.
3. **Cold tier:** This tier should be used for data with infrequent or irregular access patterns, such as data that is accessed monthly, quarterly, etc. For example, backups, archives, or historical data that is rarely accessed but needs to be kept for long periods of time for compliance or other reasons.
4. **Archive tier:** The archive tier is designed for data that is rarely accessed and has minimal retrieval requirements. It is typically used for long-term storage and compliance purposes. This tier is suitable for data with very infrequent access patterns, such as annually or even less frequently.

3.1 Solution Approach

We first define the weights (W) for each factor (size (Z), access frequency (F), and age (A)) as W_z, W_f, and W_a, respectively. Then the data is defined as a list of dictionaries, where each dictionary represents an object and contains its size, access frequency, and age. Afterwards, the priority score for each object is calculated using the defined weightings using Eq. 1 for size score(α), Eq. 2 for access frequency score (β), Eq. 3 for age score (γ) and Eq. 4 for calculating priority score (λ). The weight (W) of data indicates its priority or significance, allowing for varied importance levels across data objects, often determined by business criteria; for instance, vital data could bear a greater weight, directing it to higher-tier storage. Data size, access frequency, and age serve as pivotal determinants in storage choices, where larger values may entail increased storage costs. Applying the logarithm of these values, such as $\log_{10}(X)$, facilitates data normalization and mitigates the potential dominance of extreme values in classification. This logarithmic transformation ensures a balanced scale for storage tiers, effectively accommodating a wide range of data sizes.

$$\alpha = W_z \times \log_{10}(Z) \tag{1}$$

$$\beta = W_f \times \log_{10}(F) \tag{2}$$

$$\gamma = W_a \times \frac{A}{365} \tag{3}$$

$$\lambda = \alpha + \beta + \gamma \tag{4}$$

In this context,

- Z represents the size of data in Gigabytes (GB);
- F denotes the total number of R/W operations for an object in a specified period of time; and,
- A represents the age of the data in months.

Finally, the objects are divided into groups based on the available storage tiers by iterating over each object and checking if its priority score is greater than or equal to the threshold for each tier. If so, it is added to the corresponding group. Regarding the access frequency, the followings are the nineteen possible windows: hourly, every 2 hours, every 3 hours, every 4 hours, every 6 hours, every 8 hours, every 12 hours, daily, every other day, every 3 days, every 4 days, every week, every 2 weeks, every month, every 2 months, every 3 months, every 4 months, every 6 months, and yearly.

Priority Score Threshold. We set the following priority scores to classify each object into premium, hot, cold, or archive tiers.

- Premium: 1.0
- Hot: 0.7
- Cold: 0.4
- Archive: 0.1

The selection of priority score thresholds for each storage tier aims to balance the trade-offs between data size, access frequency, and age. The premium tier, with a threshold of 1.0, represents the highest priority for critical and frequently accessed data. This tier ensures fast and reliable access to the most valuable information. The hot tier, set at 0.7, accommodates data with slightly lower priority but still significant access requirements. It provides a balance between performance and cost for frequently accessed data. The cold tier, with a threshold of 0.4, caters to less frequently accessed data, offering cost-effective storage without compromising data availability. Lastly, the archive tier, at 0.1, serves as a long-term storage solution for rarely accessed data, providing cost optimization while preserving data retention. These thresholds enable the effective allocation of data to the appropriate storage tiers based on their priority scores, ensuring optimal cost management, while meeting the needs of data access and availability.

3.2 Evaluation

To evaluate the viability and potential of the proposed approach, a software tool has been developed to provide cost estimations based on the values obtained from Google Cloud storage when data is migrated according to the classification performed by the proposed approach.

Dataset Information. Due to limitations in acquiring a real dataset, synthetic data was generated based on publicly available data on Kaggle. It is an access log of a software application deployed on the cloud for a period of almost 2.5 years. Figure 1 shows the access pattern based on the average number of accesses of all objects over the whole storage time period. Additionally, some of the key features of the dataset are as follows:

- Total time of data storage: $A = 871$ days.
- Total number of objects $T_{obj} = 14321$
- Total number of GET Requests: $g(t) = 2906097$
- Total number of PUT Requests: $p(t) = 14321$
- Object size range: 50MB to 100MB
- Total data size $Z = 1052.45$ GB

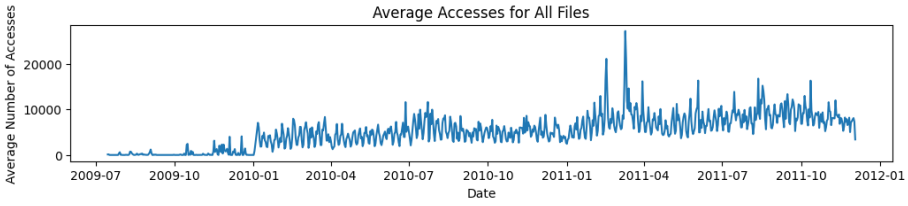

Fig. 1: Data access pattern over the whole time period. Number of accesses on the y-axis and date on the x-axis.

Figure 2 compares the data storage cost if it remained static in one storage tier. The calculation is done keeping in view that the access pattern that objects will follow for the next 871 days will be similar to the first 871 days.

Weights. If 30% weight is set for size, 20% for access frequency, and 50% for the age of the data, the combination of weights would be (0.3, 0.2, 0.5). Generally, the sum of the total weights should be equal to 1. In that case, there are a total of 36 possible combinations of weights. By removing the condition of the sum being equal to 1, we created a total of 286 combinations. Then the priority score was calculated for each combination of weights, and subsequently, the cost was calculated. Out of 286, the cost calculation script returned 169 unique values for the cost. The comparison of cost with those combinations is shown in Figure 3.

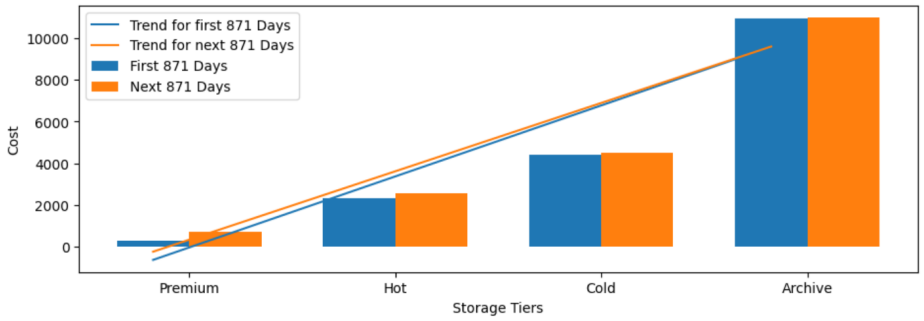

Fig. 2: Comparison of the cost of data storage for the first 871 days with each object having variable age vs. the cost of data storage if it is not moved between tiers for the next 871 days.

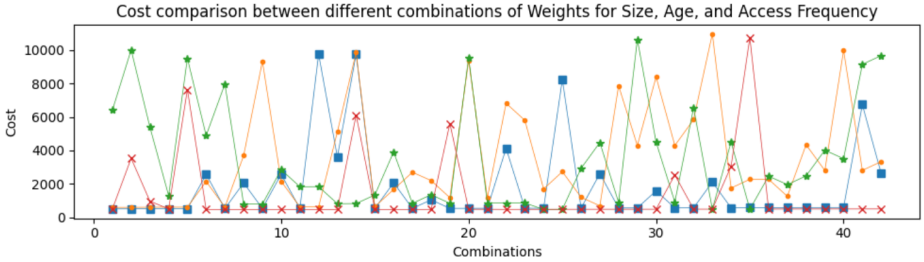

Fig. 3: Cost comparison between different combinations of weights for size, age, and access frequency. Cost in US Dollars is specified on the y-axis, whereas the combination number (#) is shown on the x-axis.

Results. Due to a high number of data access requests and free data retrieval for the premium tier, the cost of the data stored in the premium tier is the cheapest, as shown in Figure 2. Although when calculating the cost of data storage for the next 871 days, the premium tier shows the highest difference in the cost and is still cheaper than the rest of the tiers because of the low cost of data retrieval in the premium tier. Different costs are calculated using the proposed rule-based classification, and a comparison is presented in Figure 4. The effectiveness of the weights can vary according to the characteristics of the dataset, hence, for this dataset, the best suitable combination turned out to be size: 20%, access frequency: 80%, and age: 0%. It can be seen that with the proposed rule-based classification technique, the cost of data storage is $473.39. In contrast, if the data is stored in the premium tier for the whole time, the total cost is $694.35 (the cost of data migration is not included in this calculation).

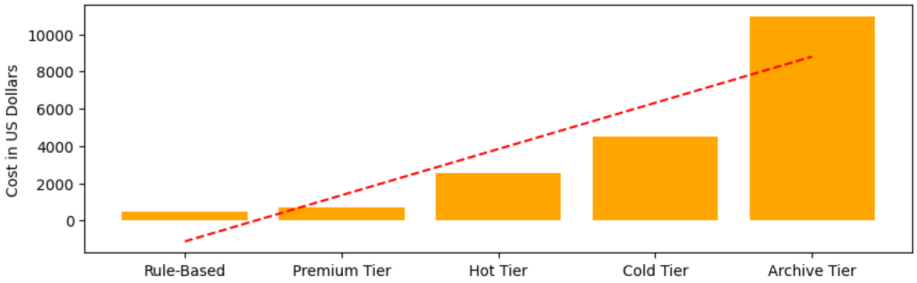

Fig. 4: Cost comparison between rule-based classification and single tiers.

4 Discussion

The proposed rule-based classification approach is lightweight, industry- and platform-independent, and has shown promising results. According to the evaluation, the cost reduction is nearly 32%; even when factoring in the cost of data migration, the difference would be significant. However, it lacks the ability to consider feedback regarding each classification. There is a chance that the classification of a storage object may not be cost-effective, and to enhance the algorithm's performance, it is crucial to incorporate that information as feedback. To tackle this challenge, we suggest utilizing game theory for the classification of storage objects into different tiers.

Game theory is a mathematical framework used to analyze the interactions and decision-making strategies of individuals or agents within a group or system [13]. We propose the use of game theory to optimize the storage tier selection in a multi-agent system, where each agent is responsible for storing and retrieving data. One approach is to use a variant of the multi-armed bandit problem, where the agents are the arms, and the storage tiers are the bandits. One possible implementation of this approach could use the Thompson Sampling algorithm [22], which is a Bayesian approach to the multi-armed bandit problem. In this algorithm, each agent maintains a beta distribution over the storage tiers, where the parameters of the distribution represent the number of successes and failures in selecting a storage tier. The agent selects a storage tier based on the highest sampled value from the distribution. The update of the distribution is done after the storage operation is completed, based on the feedback from the system. Specifically, if the storage cost is lower than the expected cost from the distribution, the parameters of the beta distribution are updated to reflect success.

5 Related Work

Existing related work primarily focuses on two tiers: hot and cold. Hot data is frequently accessed and requires high-performance storage. Cold data is accessed infrequently and can be stored on lower-cost storage.

Liu et al. [7] proposed RLTiering, an auto-tiering system that uses deep reinforcement learning to place data in the most cost-effective tier in cloud storage. They also proposed a randomized online migration algorithm [6] for cost optimization. Similarly, Erradi et al. [4] proposed two online cost optimization algorithms for tiered cloud storage services. They are designed to minimize the overall cost of storage, while meeting the Quality of Service (QoS) requirements of users. The first algorithm is a greedy algorithm that places data in the cheapest tier that meets the QoS requirements of users. The second algorithm is a reinforcement learning algorithm that learns to place data in the most cost-effective tier over time. Alshawabkeh et al. [3] developed an automated and adaptive framework using efficient Markov chain correlation-based clustering to move active data to high-performance storage tiers and inactive data to low-cost/high-capacity storage tiers. This framework can predict workload changes and group similar storage units, enhancing performance, reliability, and availability and reducing cost. On the contrary, we propose an approach to storage tiering that considers four storage tiers: premium, hot, cold, and archive.

Mansouri and Erradi [9], as well as Erradi and Mansouri [4], introduced a series of deterministic online algorithms to address cost reduction in this particular problem. However, the aspect of access frequency, specifically the number of access requests, was not taken into account during their decision-making process. Our approach, however, takes into account three main factors when determining which tier to store an object size, age, and access frequency. Moreover, Zhang et al. [25] investigated how cloud providers can maximize their profits by using hot and cold storage tiers, but our research focuses on how cloud users can minimize their costs by using hot and cold storage tiers. The multi-cloud setting is also introduced by some scientific studies that consider migrating data among multiple clouds for achieving cost-effective geo-distributed workloads [10][18][17]. In [12], Facebook developed a storage tier optimization approach and targeted two storage tiers. In addition to that, their proposed approach made decisions based on the characteristics of the whole bucket. Our algorithm makes decisions on objects rather than buckets, hence proposing a more flexible approach. In addition to that, it is generic, platform- and industry-independent.

6 Conclusion and Future Work

Maintaining data in a single tier continuously is ineffective and expensive. We explored a rule-based approach that examines object metadata and access patterns for storage tier optimization. The rule-based classification was demonstrated to be successful on a synthetic data set and is straightforward and simple to use using $\lambda = \alpha + \beta + \gamma$ for priority score calculations. We also proposed using game theory, which is more complex, to improve the accuracy of the proposed approach. The suggested approach is not platform- or industry-specific and is also not very resource-intensive in terms of computation. It can, therefore, be considered appropriate for a variety of applications. The findings indicate that while developing such an algorithm, it is crucial to take into account the access

patterns and metadata of storage items. Additionally, it was demonstrated that by utilizing the suggested approach, storage cost can be decreased.

In the future we aim to extend the proposed approach using game theory to improve the accuracy of our predictions. By using game theory, we can model the interactions between different entities in our system and develop an algorithm that can anticipate their behaviour and make more accurate predictions. In addition, to make the estimations and comparisons more accurate and concise, there is a need for comprehensive mathematical modelling that not only correctly calculates the costs, but also takes into account the followings: 1) network usage cost based on block pricing; 2) data migration costs; and 3) penalty fees if an object is removed before the minimum time period specified for that particular tier. These should be used to generate accurate and concise estimates and comparisons of the cost for different storage options.

Acknowledgments. The first author is a Ph.D. Candidate. This work received partial funding from the projects DataCloud (H2020 101016835), enRichMyData (HE 101070284), Graph-Massivizer (HE 101093202), UPCAST (HE 101093216), and BigDataMine (NFR 309691).

References

1. What is a storage device hierarchy? `https://www.ibm.com/docs/en/zos/2.2.0?topic=dfsmshsm-what-is-storage-device-hierarchy` (2021), accessed: 2023-05-20
2. Tier definitions and volume placement optimization. `https://www.ibm.com/docs/en/storage-insights?topic=SSQRB8/com.ibm.spectrum.si.doc/tpch_saas_r_volume_optimization_process.htm` (2022), accessed: 2023-05-20
3. Alshawabkeh, M., Riska, A., Sahin, A., Awwad, M.: Automated storage tiering using markov chain correlation based clustering. In: Proceedings of the 11th International Conference on Machine Learning and Applications (ICMLA 2012). vol. 1, pp. 392–397. IEEE (2012). https://doi.org/10.1109/ICMLA.2012.71
4. Erradi, A., Mansouri, Y.: Online cost optimization algorithms for tiered cloud storage services. Journal of Systems and Software **160**, 110457 (2020). https://doi.org/10.1016/j.jss.2019.110457
5. Khan, A.Q., Nikolov, N., Matskin, M., Prodan, R., Song, H., Roman, D., Soylu, A.: A taxonomy for cloud storage cost. In: Proceedings of the 15th International Conference on Management of Digital Ecosystems. Springer (2023)
6. Liu, M., Pan, L., Liu, S.: Keep hot or go cold: A randomized online migration algorithm for cost optimization in staas clouds. IEEE Transactions on Network and Service Management **18**(4), 4563–4575 (2021). https://doi.org/10.1109/TNSM.2021.3096533
7. Liu, M., Pan, L., Liu, S.: RLTiering: A Cost-Driven Auto-Tiering System for Two-Tier Cloud Storage Using Deep Reinforcement Learning. IEEE Transactions on Parallel and Distributed Systems **34**(2), 73–90 (2022). https://doi.org/10.1109/TPDS.2022.3224865
8. Macedo, R., Paulo, J.a., Pereira, J., Bessani, A.: A Survey and Classification of Software-Defined Storage Systems. ACM Computing Surveys **53**(3) (2020). https://doi.org/10.1145/3385896

9. Mansouri, Y., Erradi, A.: Cost optimization algorithms for hot and cool tiers cloud storage services. In: Proceedings of the 11th International Conference on Cloud Computing (CLOUD 2018). pp. 622–629. IEEE (2018). https://doi.org/10.1109/CLOUD.2018.00086
10. Mansouri, Y., Toosi, A.N., Buyya, R.: Cost optimization for dynamic replication and migration of data in cloud data centers. IEEE Transactions on Cloud Computing **7**(3), 705–718 (2017). https://doi.org/10.1109/TCC.2017.2659728
11. Mansouri, Y., Toosi, A.N., Buyya, R.: Data Storage Management in Cloud Environments: Taxonomy, Survey, and Future Directions. ACM Computing Surveys **50**(6) (2017). https://doi.org/10.1145/3136623
12. Muralidhar, S., Lloyd, W., Roy, S., Hill, C., Lin, E., Liu, W., Pan, S., Shankar, S., Sivakumar, V., Tang, L., et al.: f4: Facebook's warm BLOB storage system. In: Proceedings of the 11th USENIX Symposium on Operating Systems Design and Implementation. pp. 383–398. USENIX Association (2014)
13. Myerson, R.B.: Game theory: analysis of conflict. Harvard university press (1997)
14. Naldi, M., Mastroeni, L.: Cloud storage pricing: A comparison of current practices. In: Proceedings of the International Workshop on Hot Topics in Cloud Services (HotTopiCS 2013). pp. 27–34. ACM (2013). https://doi.org/10.1145/2462307.2462315
15. Nikolov, N., Dessalk, Y.D., Khan, A.Q., Soylu, A., Matskin, M., Payberah, A.H., Roman, D.: Conceptualization and scalable execution of big data workflows using domain-specific languages and software containers. Internet Things **16**, 100440 (2021). https://doi.org/10.1016/j.iot.2021.100440
16. Nuseibeh, H.: Adoption of cloud computing in organizations. In: Proceedings of the Americas Conference on Information Systems (AMCIS 2011). AISeL (2011)
17. Oh, K., Chandra, A., Weissman, J.: Wiera: Towards flexible multi-tiered geo-distributed cloud storage instances. In: Proceedings of the 25th ACM International Symposium on High-Performance Parallel and Distributed Computing (HPDC 2016). pp. 165–176. ACM (2016). https://doi.org/10.1145/2907294.2907322
18. Qiu, X., Li, H., Wu, C., Li, Z., Lau, F.C.: Cost-minimizing dynamic migration of content distribution services into hybrid clouds. IEEE Transactions on Parallel and Distributed Systems **26**(12), 3330–3345 (2014). https://doi.org/10.1109/INFCOM.2012.6195655
19. Robinson, K.: Why companies are flocking to the cloud more than ever. `https://www.businessinsider.com/cloud-technology-trend-software-enterprise-2021-2` (2021), accessed: 2023-02-20
20. Roman, D., Prodan, R., Nikolov, N., Soylu, A., Matskin, M., Marrella, A., Kimovski, D., Elvesæter, B., Simonet-Boulogne, A., Ledakis, G., Song, H., Leotta, F., Kharlamov, E.: Big Data Pipelines on the Computing Continuum: Tapping the Dark Data. Computer **55**(11), 74–84 (2022). https://doi.org/10.1109/MC.2022.3154148
21. Rydning, D.R.J.G.J., Reinsel, J., Gantz, J.: The digitization of the world from edge to core. Tech. rep., Framingham: International Data Corporation (2018)
22. Thompson, W.R.: On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. Biometrika **25**(3-4), 285–294 (1933)
23. Tung, A.K.H.: Rule-based Classification, pp. 2459–2462. Springer US, Boston, MA (2009)
24. Yang, C., Xu, Y., Nebert, D.: Redefining the possibility of digital Earth and geosciences with spatial cloud computing. International Journal of Digital Earth **6**(4), 297–312 (2013). https://doi.org/10.1080/17538947.2013.769783
25. Zhang, Y., Ghosh, A., Aggarwal, V., Lan, T.: Tiered cloud storage via two-stage, latency-aware bidding. IEEE Transactions on Network and Service Management **16**(1), 176–191 (2018). https://doi.org/10.1109/TNSM.2018.2875475