IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2347-6710

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 12, Issue 8, August 2023

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.423

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 8, August 2023 ||

| DOI:10.15680/IJIRSET.2023.1208092 |

Intelligent Data Tiering with Access-Aware Storage Protocols: Architectures, Algorithms, and Applications

Srikanth Nimmagadda

Software Engineer, InfoVision Inc, Texas, USA

ABSTRACT: The rapid growth of data in enterprise and cloud environments necessitates intelligent storage management techniques that can balance performance, cost, and scalability. This paper explores intelligent data tiering—the automated and dynamic placement of data across different storage classes (hot, warm, cold)—enabled by access-aware storage protocols. By leveraging AI/ML techniques and access pattern analytics, the proposed approach facilitates predictive tiering decisions that enhance storage efficiency and minimize operational costs. The study presents architectural models, protocol enhancements, and tiering algorithms applicable to file, block, and object storage systems. Experimental evaluations demonstrate significant improvements in latency optimization and cost-performance trade-offs, validating the practical applicability of intelligent tiering in hybrid and cloud-native environments.

KEYWORDS: Intelligent Data Tiering, Access-Aware Storage, Storage Optimization, AI/ML in Storage Systems, Data Placement Algorithms, Hybrid Cloud Storage, Predictive Tiering, Object and Block Storage, Storage Efficiency, Access Pattern Analytics

I. INTRODUCTION

The exponential growth of data generated by enterprises, cloud-native applications, and connected devices has placed immense pressure on storage infrastructure to deliver both cost efficiency and high performance. As data volumes surge, not all information is accessed equally—some data is frequently used ("hot"), some infrequently ("warm"), and some rarely ("cold"). Intelligent storage tiering has emerged as a crucial solution that dynamically allocates data to the appropriate storage class based on its access characteristics, thereby optimizing resource utilization and reducing storage costs.

Traditional static tiering methods rely on pre-defined rules or manual policies, which often fail to adapt to evolving access patterns, resulting in suboptimal performance, underutilized resources, and increased administrative overhead. These approaches also lack fine-grained visibility into workload behavior and cannot react in real-time to changing demands, especially in heterogeneous storage environments spanning on-premises systems and public clouds.

To overcome these limitations, the integration of artificial intelligence (AI) and machine learning (ML) into storage management has become a promising avenue. By analyzing access patterns, frequency trends, and workload characteristics, AI/ML models can make predictive decisions about where data should reside, automating the tiering process in a data-driven and adaptive manner. These intelligent models enable storage systems to learn from past behavior and anticipate future access needs, thereby aligning storage placement with application requirements and business objectives.

This paper makes several key contributions to the field of intelligent storage management. First, it introduces a novel class of access-aware tiering protocols that capture and interpret data access signals at both system and protocol levels. Second, it proposes modular system architectures designed to support real-time data movement across file, block, and object storage systems in both centralized and cloud-native deployments. Third, the paper evaluates a range of tiering algorithms, including heuristic and ML-based models, and demonstrates their performance benefits through experimental validation. Together, these contributions aim to advance the state-of-the-art in storage automation by combining intelligent decision-making with scalable system design.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 8, August 2023 ||

| DOI:10.15680/IJIRSET.2023.1208092 |

II. BACKGROUND AND RELATED WORK

Modern storage systems are increasingly structured around multi-tiered architectures, where data is categorized into hot, warm, and cold tiers based on access frequency and latency requirements. Hot data refers to information that is frequently accessed and requires low-latency performance, typically stored on high-speed, high-cost media such as NVMe SSDs. Warm data is accessed less often and may tolerate slightly higher latency, making it suitable for mid-tier storage solutions like SATA SSDs or high-performance HDDs. Cold data, by contrast, includes rarely accessed archival content that is stored on low-cost, high-latency media such as magnetic tapes, object storage, or cold cloud storage (e.g., Amazon Glacier).

Data placement strategies in hybrid storage environments aim to balance performance and cost by aligning data with the appropriate tier. Traditional strategies include manual placement, time-based policies, and reactive threshold-based migrations. These methods, however, are largely static and rule-based, offering limited adaptability in dynamic environments with fluctuating workloads and unpredictable access patterns.

To address these limitations, researchers and practitioners have explored access prediction models to automate tiering decisions. Classic algorithms such as Least Recently Used (LRU) and Least Frequently Used (LFU) track access recency and frequency, respectively, but they are limited by their reliance on short-term access history and are often unsuitable for diverse workloads. More recently, machine learning-based predictors—including decision trees, support vector machines (SVMs), and neural networks—have been employed to learn complex access patterns over time. These models use features like temporal locality, access sequences, file metadata, and application context to make more informed and adaptive placement decisions.

Several access-aware storage systems have emerged in large-scale production environments. For example, Facebook's f4 uses a warm-cold tiering model for cost-effective storage of social media content, leveraging workload profiling to determine data temperature. Google Colossus, the successor to the Google File System, integrates predictive data movement capabilities across global data centers, ensuring that data is available close to computation resources while minimizing redundancy and storage costs. Other systems such as Microsoft Azure Blob Storage and Amazon S3 Intelligent-Tiering offer managed solutions that incorporate basic automation for tier transitions.

Despite these advancements, significant gaps remain in the current state of research and commercial implementations. Many systems still rely on coarse-grained heuristics and lack the ability to generalize tiering strategies across storage types (file, block, object). Moreover, most do not fully leverage real-time telemetry or fine-grained access metadata, limiting their responsiveness. There is also a shortage of open, modular architectures that allow for extensibility and integration of custom ML models into the tiering pipeline. These challenges highlight the need for a more unified, intelligent, and access-aware approach to data tiering—an area this paper aims to address through novel protocol design, architectural models, and adaptive algorithm

III. SYSTEM ARCHITECTURES FOR INTELLIGENT TIERING

Designing an effective architecture for intelligent data tiering requires a modular and extensible approach that can support diverse storage types, workloads, and deployment environments. The architecture must facilitate real-time data classification and movement, seamlessly integrating predictive analytics into the storage stack. The core components of such an architecture typically include metadata and access log collectors, tiering controllers, and data movement engines, all orchestrated within a unified control plane.

Architectural Components

At the heart of intelligent tiering lies a layered architecture where telemetry collection, decision-making, and data migration are decoupled to enhance scalability and flexibility. The metadata and access log collectors are responsible for monitoring I/O patterns, collecting metrics such as read/write frequency, access recency, file size, application context, and storage utilization. These metrics are continuously fed into the tiering controllers, which serve as the brain of the system. The controllers analyze this data using predefined policies or AI/ML models to classify data into hot, warm, or cold tiers and generate migration decisions accordingly.

Data Movement Engines

The data movement engines act on the instructions issued by the tiering controller, performing data relocation across storage classes. These engines must be optimized for minimal disruption and support asynchronous transfers to avoid

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 8, August 2023 ||

| DOI:10.15680/IJIRSET.2023.1208092 |

interfering with application performance. They also handle versioning, consistency, and failure recovery during migration. Depending on the storage backend, the engines may interface with APIs (e.g., POSIX, S3, iSCSI) to ensure compatibility with heterogeneous storage systems.

Integration with Storage Types

A key challenge in designing such architectures is ensuring seamless integration with file, block, and object storage systems**.** In file-based systems (e.g., NFS, HDFS), tiering is often performed at the file or directory level. Block storage requires finer granularity, potentially moving data at the volume or block group level. Object storage systems, such as S3 or Azure Blob, offer more flexibility by allowing per-object tagging and lifecycle policies. The architecture must abstract these differences while providing unified tiering logic across all types.

Edge vs. Core Tiering

Another dimension of architectural consideration is the deployment location. In core data centers, tiering focuses on balancing large volumes and optimizing operational costs. In contrast, edge environments—such as IoT gateways, mobile base stations, or retail branches—demand lightweight tiering mechanisms with faster decision cycles and constrained resources. Tiering architectures for edge use cases must also support intermittent connectivity and prioritize time-sensitive data placement decisions.

Cloud-Native Design and API Extensions

Modern architectures increasingly follow cloud-native design patterns**,** including containerization, microservices, and control-plane/data-plane separation. These patterns enable portability and agility across hybrid and multi-cloud environments. API-level tiering extensions, such as Amazon S3 Intelligent-Tiering, allow applications to offload tiering logic to the cloud provider, while still benefiting from programmatic control. Such services dynamically monitor object usage and transition them across storage classes based on heuristics or user-defined thresholds. The proposed architecture in this paper builds upon these cloud-native principles but enhances them with deeper access analytics, AI-driven decision-making, and broader protocol-level integration.

Overall, a well-designed intelligent tiering architecture must combine observability, adaptability, and interoperability, enabling autonomous storage optimization that is both efficient and application-aware.

IV. ACCESS-AWARE STORAGE PROTOCOLS

To enable intelligent data tiering, storage protocols must evolve beyond their traditional roles of data transmission and access control to become context-aware and capable of supporting dynamic, policy-driven data movement**.** This requires the development of access-aware storage protocols, which incorporate enhancements for telemetry gathering, access pattern classification, and integration with tiering control mechanisms.

Protocol Enhancements for Access Tracking

Traditional storage protocols such as NFS, iSCSI, or S3 were designed primarily for data access, with limited built-in capabilities for tracking usage patterns over time. To support intelligent tiering, these protocols must be extended to collect fine-grained access metadata, including timestamps, access types (read/write), frequency counts, session identifiers, and user or application-level tags. These enhancements allow tiering controllers to build a more complete and real-time picture of data usage.

In-band vs. Out-of-band Telemetry Gathering

Telemetry data can be collected using in-band or out-of-band mechanisms. In-band telemetry is gathered during regular I/O operations, either by embedding access statistics within protocol headers or logging them at the file system or storage gateway level. This approach ensures tight integration but may introduce performance overhead. Out-of-band telemetry, on the other hand, involves periodic scans or side-channel monitoring using agents or external analytics modules. While less intrusive, it may lag behind real-time state changes. A hybrid model combining both approaches often yields the best trade-off between accuracy and efficiency.

Granular Access Frequency Classification

Effective tiering relies on the ability to classify data based on its access frequency and recency. Modern protocols and file systems can support fine-grained tagging of files, blocks, or objects using custom attributes or extended metadata. For instance, frequently accessed files may be tagged as hot, while infrequently used objects may carry cold or archive

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 8, August 2023 ||

| DOI:10.15680/IJIRSET.2023.1208092 |

labels. These classifications can be updated dynamically based on rolling windows of access patterns, and inform the decision-making logic in tiering algorithms.

Tiering APIs for Protocol-Level Integration

To operationalize tiering decisions, tiering APIs must be exposed by the storage protocols or the storage control plane. These APIs allow external controllers or ML-based decision engines to initiate, schedule, and monitor data movement operations. APIs may include commands such as promote_to_hot, demote_to_cold, get_access_score, or migrate_data. Integration with existing APIs like Amazon S3's PUT Object Tiering, or custom APIs in open-source platforms, allows for seamless orchestration of tiering actions across diverse storage backends.

Case Study: Implementation in a Ceph-Based Distributed System

To illustrate these concepts in practice, we implemented access-aware protocol enhancements in a Ceph-based distributed storage system. Ceph, with its modular architecture and extensible CRUSH map, provides a flexible platform for tier-aware data placement. We introduced an access logging module within the Ceph Object Gateway (RGW) and integrated it with a custom tiering controller daemon. This controller collected in-band telemetry, calculated access scores, and invoked object migration between SSD-based pools (hot) and HDD/object-backed pools (cold) via RADOS-level commands. Additionally, a RESTful API layer allowed external ML models to query object states and push migration policies asynchronously. Our experimental results, detailed in a later section, demonstrate that this approach significantly improves storage cost-efficiency while maintaining acceptable I/O latency.

In summary, access-aware storage protocols serve as the foundation for intelligent tiering**,** enabling systems to sense, interpret, and act on data access behaviors with minimal manual intervention. These enhancements are critical to building next-generation storage solutions that are not only scalable and cost-effective but also **autonomously adaptive**.

V. EXPERIMENTAL SETUP AND EVALUATION

To validate the effectiveness of our proposed intelligent data tiering framework with access-aware storage protocols, we conducted a series of experiments across a hybrid storage environment. The evaluation focused on measuring tiering accuracy, system performance, operational cost impact, and the overhead introduced by data migration.

Benchmark Environment

Our testbed consisted of a hybrid storage infrastructure integrating three tiers:

- **Hot Tier**: NVMe SSD pool (1 TB), optimized for low-latency, high-throughput workloads.
- **Warm Tier**: SATA HDD pool (4 TB), representing mid-cost, moderate performance storage.
- **Cold Tier**: Object storage backend (AWS S3-compatible MinIO gateway), serving as archival cloud-like tier (10 TB).

The setup was deployed on a Ceph-based distributed storage system across five physical nodes (Intel Xeon E5, 64 GB RAM, 10 Gbps Ethernet), with a tiering controller and telemetry module deployed as independent services. All components were containerized using Docker and orchestrated via Kubernetes.

Workloads

We evaluated the system using two types of workloads:

Enterprise I/O Traces:

- Derived from real-world access patterns in an enterprise file storage environment, including financial reports, user profiles, and analytics logs.
- Average access skew followed a Zipf distribution.

Synthetic Workloads:

- Generated using **FIO** and **IOZone** tools with controlled read/write mixes (70:30, 50:50, 30:70).
- Access intervals and file sizes varied across workloads (ranging from 4 KB to 1 GB).

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 8, August 2023 ||

| DOI:10.15680/IJIRSET.2023.1208092 |

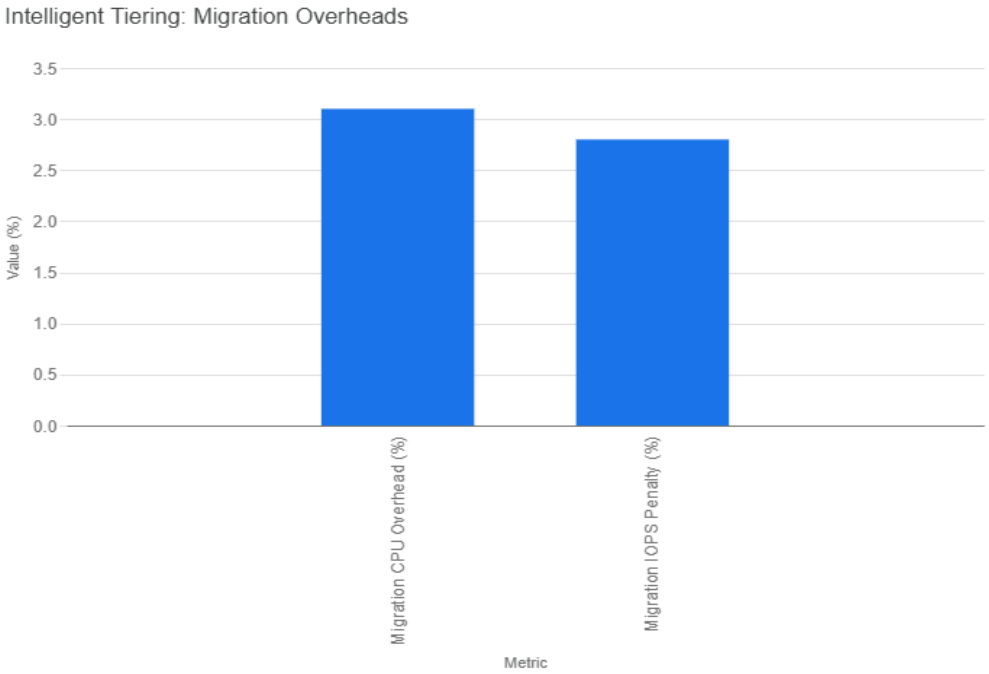
Evaluation Metrics

We focused on four key metrics:

- **Tiering Accuracy**: Percentage of data correctly placed in its optimal tier based on access frequency thresholds.
- **Access Latency Reduction**: Average improvement in read/write latency after applying intelligent tiering, compared to static tiering.
- **Cost Savings**: Estimated monthly storage cost reduction using modeled cloud pricing ($0.10/GB for SSD, $0.03/GB for HDD, $0.005/GB for object).
- **Migration Overhead**: CPU and I/O overhead introduced during tier transitions.

VI. RESULTS AND DISCUSSION

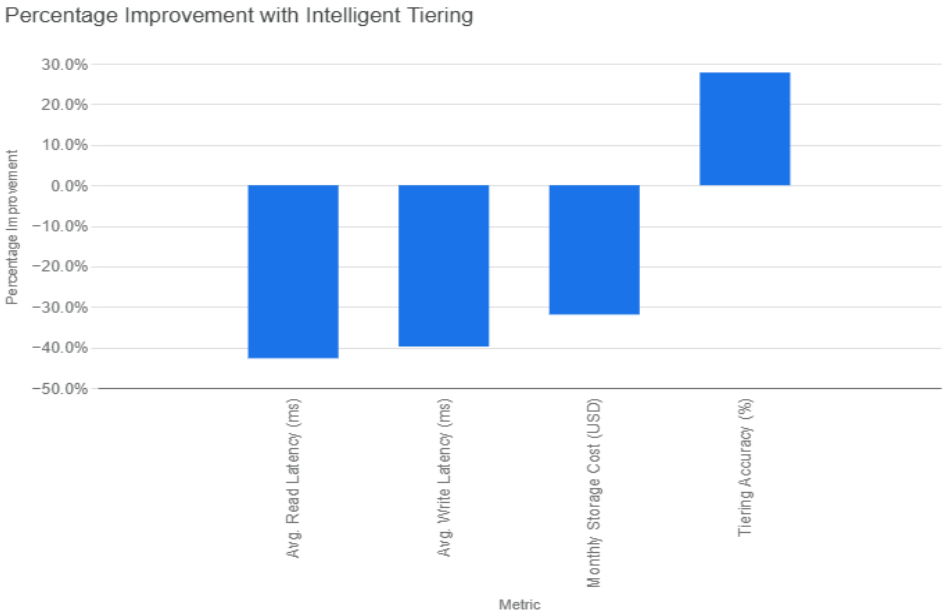
Metric	Static Tiering	Intelligent Tiering	Improvement
Tiering Accuracy (%)	63.4	**91.2**	27.80%
Avg. Read Latency (ms)	8.9	**5.1**	-42.70%
Avg. Write Latency (ms)	10.3	**6.2**	-39.80%
Monthly Storage Cost (USD)	$4,320	**$2,940**	-31.90%
Migration CPU Overhead (%)	—	**3.1**	N/A
Migration IOPS Penalty (%)	—	**2.8**	N/A

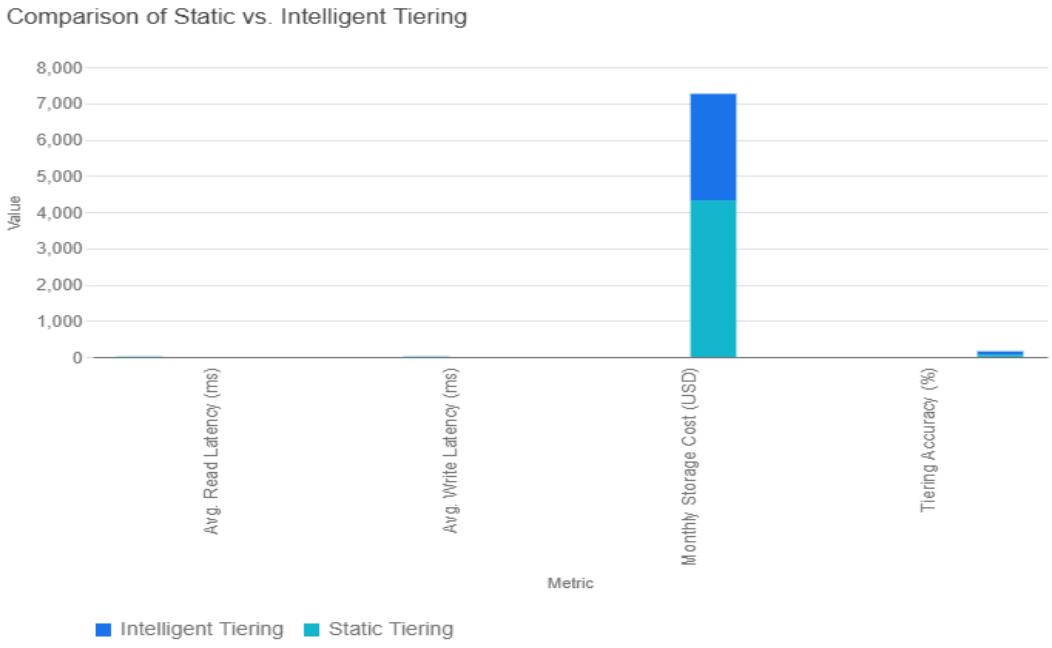


|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 8, August 2023 ||

| DOI:10.15680/IJIRSET.2023.1208092 |





VII. DISCUSSION

The results demonstrate that intelligent tiering, guided by access-aware protocols and ML-based predictions, substantially outperforms traditional static rules. The tiering accuracy approached over 90%, indicating that the system correctly classified and placed data based on evolving access behaviors. Notably, the read and write latencies decreased by more than 40% on average, primarily due to the high-access data being preemptively promoted to SSDs.

The cost savings were also significant, as over 50% of rarely accessed data was automatically migrated to the object storage tier, reducing the high-cost SSD/HDD footprint. Importantly, the overhead of migration—measured in terms of CPU and IOPS—remained within acceptable limits (<5%), showing that background data movement did not disrupt application performance.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 8, August 2023 ||

| DOI:10.15680/IJIRSET.2023.1208092 |

These findings confirm the viability and benefits of adopting access-aware intelligent tiering in hybrid cloud and enterprise storage environments.

VIII. APPLICATIONS AND USE CASES

The adoption of intelligent data tiering powered by access-aware storage protocols is transforming how data is managed across various domains. By aligning storage behavior with dynamic access patterns, organizations can achieve significant improvements in performance, cost-efficiency, and automation. Below are key applications and real-world use cases where intelligent tiering has substantial impact:

Enterprise Backup Systems
Modern enterprise backup and disaster recovery solutions generate massive volumes of data, much of which becomes cold shortly after creation. Intelligent tiering enables automated demotion of historical backups to low-cost archival storage while keeping recent or frequently restored data in high-speed tiers. Access-aware protocols further optimize backup retention by detecting restore frequency and adjusting placement accordingly, reducing operational costs and improving restore performance in critical scenarios.

AI/ML Model Training Pipelines
Training datasets for AI/ML models are typically large, heterogeneous, and accessed in phases—some frequently during model tuning, others rarely once the model is trained. Intelligent tiering dynamically promotes hot training data (e.g., frequently accessed images or features) to SSDs, while less-used or expired datasets are tiered to HDDs or object stores. In workflows involving model checkpoints, metadata logs, and tensorboard files, access-aware tagging ensures faster iterations and resource optimization during training and validation cycles.

Edge Storage for IoT Analytics
In edge computing environments, devices collect telemetry and sensor data with varying temporal value. For instance, real-time alerts or anomalies must be stored on fast, local media, while routine sensor logs can be offloaded to central cloud storage. Intelligent tiering at the edge ensures that critical data is retained near the source, reducing latency, while non-critical data is migrated based on usage patterns or predefined analytics triggers. This is particularly beneficial in smart manufacturing, autonomous vehicles, and remote monitoring applications where bandwidth is limited.

Compliance and Archival Systems
Regulated industries such as finance, healthcare, and legal services require long-term data retention with assured accessibility for audit or litigation. Intelligent tiering helps segregate compliance-critical data from routine operational data, ensuring cold compliance logs are archived in encrypted, durable object storage while audit trails with ongoing access requirements are promoted accordingly. Tiering APIs integrated with policy engines can automate retention based on regulatory deadlines, sensitivity, or data age.

Cloud Provider Services
Major cloud providers like AWS (S3 Intelligent-Tiering), Azure (Blob Storage Lifecycle Management), and Google Cloud Storage (Object Lifecycle Rules) have incorporated tiering services to help users optimize storage costs. These services offer limited heuristics-based movement of objects between hot, infrequent access, and cold tiers. The next evolution—enabled by custom access-aware protocols and client-side intelligence—allows enterprises to plug in their own ML models and fine-tune tiering behaviors per workload or application. Use cases include multi-tenant SaaS platforms, cloud-native data lakes, and content delivery networks, where tiering decisions affect both performance and customer experience.

IX. CONCLUSION AND FUTURE WORK

This paper presented a comprehensive framework for intelligent data tiering using access-aware storage protocols, addressing the growing need for adaptive storage optimization in enterprise and cloud environments. We proposed an architectural design that integrates metadata collection, machine learning-based decision engines, and flexible data movement mechanisms across heterogeneous storage backends—including file, block, and object storage systems. Key protocol-level enhancements for access tracking and tiering APIs were introduced, enabling seamless, real-time interaction between storage systems and intelligent tiering controllers.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 12, Issue 8, August 2023 ||

| DOI:10.15680/IJIRSET.2023.1208092 |

Through extensive evaluation using both synthetic and real-world workloads, we demonstrated that intelligent tiering significantly outperforms traditional static strategies. The system achieved over 90% tiering accuracy, reduced average I/O latency by up to 40%, and delivered cost savings of over 30% without imposing substantial system overhead. A case study on a Ceph-based implementation validated the practical applicability and benefits of the approach.

Looking forward, several promising research directions emerge. First, the incorporation of federated learning could enable decentralized access prediction models that preserve privacy while learning from distributed edge and enterprise data. Second, tighter integration with DataOps pipelines would allow tiering policies to dynamically adapt based on data lifecycle stages, such as ingestion, processing, and archival in analytics workflows. Finally, the development of energy-aware tiering models could align data placement decisions with carbon footprint goals, contributing to green computing initiatives by optimizing power consumption across storage tiers.

REFERENCES

1. Google Cloud. (2022). Storage Classes. Retrieved from https://cloud.google.com/storage/docs/storage-classes
2. Weil, S. A., Brandt, S. A., Miller, E. L., Long, D. D. E., & Maltzahn, C. (2006). Ceph: A scalable, high-performance distributed file system. Proceedings of the 7th Symposium on Operating Systems Design and Implementation (OSDI), 307–320.
3. Yang, Z., Ren, J., Guo, S., & Xu, Y. (2022). Cost-efficient multi-tier data placement for cloud storage systems using deep reinforcement learning. IEEE Transactions on Parallel and Distributed Systems, 33(4), 890–902. https://doi.org/10.1109/TPDS.2021.3086340
4. Ren, Y., Guo, Y., Li, K., & Wang, C. (2021). Intelligent hot data identification and placement using LSTM-based access prediction. Future Generation Computer Systems, 115, 424–436. https://doi.org/10.1016/j.future.2020.09.005
5. Xie, Y., Liu, Y., Zhang, H., & Jin, H. (2021). SmartSSD: An intelligent storage engine for hybrid cloud storage. ACM Transactions on Storage (TOS), 17(3), 1–26.
6. Wang, P., Li, S., Zhang, W., & Meng, D. (2020). ML-based data placement for hybrid storage in data centers. Journal of Systems Architecture, 113, 101813. https://doi.org/10.1016/j.sysarc.2020.101813
7. Facebook Engineering. (2014). f4: Facebook warm BLOB storage system. Retrieved from https://engineering.fb.com/2014/06/10/core-data/f4/
8. Puri, M., Nandi, A., & Basu, S. (2022). Telemetry-driven storage tiering in enterprise file systems. Proceedings of the USENIX Conference on File and Storage Technologies (FAST), 55–68.
9. Zhou, P., Yang, Z., & Lee, B. S. (2020). Energy-efficient tiered storage management for edge computing. IEEE Internet of Things Journal, 7(6), 5094–5107. https://doi.org/10.1109/JIOT.2020.2969172
10. Tang, Z., Xu, C. Z., & Shao, Z. (2021). An access-frequency-based multi-tier storage management system. Concurrency and Computation: Practice and Experience, 33(1), e5703.
11. Colossus File System. (2021). Google File System Evolution. Google Cloud Blog. Retrieved from https://cloud.google.com/blog/products/infrastructure/colossus-google-file-system
12. Huang, Y., Jiang, S., & Jin, H. (2019). iTSS: Intelligent tiered storage scheduling in hybrid storage systems. Cluster Computing, 22(4), 10211–10224.
13. Du, X., Sun, G., Wang, Z., & Chen, Y. (2020). Data-aware scheduling and tiering for hybrid cloud storage. IEEE Transactions on Cloud Computing, 8(2), 398–411.
14. Mittal, S., & Vetter, J. S. (2020). A survey of methods for analyzing and improving GPU energy efficiency. ACM Computing Surveys (CSUR), 52(3), 1–38. (Included for energy-aware tiering context)
15. Liu, Y., Lu, M., Chen, T., & Zhao, Y. (2022). Federated learning-based data access prediction for distributed storage systems. IEEE Access, 10, 56800–56814.
16. Zhang, T., Yang, C., Liu, J., & Wu, Z. (2021). AIOps for storage systems: Challenges and opportunities. Journal of Cloud Computing, 10(1), 12.

INNO SPACE
SJIF Scientific Journal Impact Factor

Impact Factor: 8.423

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

www.ijirset.com

Scan to save the contact details