



Compression Strategies for Nation-Scale Mobility Data Vaults

Bhasker Katta

Independent Researcher
kattaabhasker@gmail.com

Abstract

Most national organizations are establishing national platforms to collect, analyze, and share transportation data. Although data storage is a critical component, it often does not receive the same level of attention as other aspects. This paper addresses big data storage optimization techniques for national transportation data platforms. It examines the data ecosystem of a hypothetical national platform and details the types of anticipated data storage optimization. The research identifies relevant techniques, summarizes representative configurations, and discusses quantifiable effects in existing platforms.

National platforms enable data-driven insights for public and private sectors by offering a unified environment for transportation analytic applications based on data fusion—combining multiple data sources and types into a single data set for analysis. Big data technologies are essential for realizing these platforms, as they require significant computing and storage facilities to process large volumes of heterogeneous data that arrive at high velocities with the potential for high veracity. Even though storage systems are a key component, comparing large-scale storage architecture and optimization techniques remains largely unexplored. Such analysis is particularly important for national platforms because transport constitutes a basic national infrastructure, and disruption and degradation will incur heavy economic losses. Without cheap access costs, deployment design decisions cannot be met.

Keywords: National Transportation Data Platforms, Big Data Storage Optimization, Transportation Data Ecosystems, Data Fusion in Transportation Analytics, Large-Scale Data Storage Architectures, Heterogeneous Transportation Data Integration, High-Velocity Transportation Data Processing, Cloud-Native Transportation Data Systems, National Infrastructure Data Platforms, Scalable Big Data Storage Systems, Data-Driven Transportation Analytics, Distributed Data Storage Technologies, High-Volume Data Management Systems, Transportation Data Engineering Frameworks.

1. Introduction

Timely, reliable, and safe surface transportation services are vital for economic growth and the well-being of society. Government agencies and authorities are responsible for



maintaining transportation infrastructure, providing services, and ensuring public safety. Timely safety-related data are also crucial for industry operators, shippers, and the traveling public. To facilitate these needs, economies at all levels have developed national transportation data platforms. The platforms take advantage of emerging technologies, such as cloud computing, to collect, store, standardize, and provision a wide variety of transportation-related data to multiple stakeholders for hundreds of data use cases, including predictive safety analysis, risk assessment, monitoring, surveillance, and general research.

National transportation data platforms harbor unprecedented amounts of surface transportation data, including data from the Intelligent Transportation Systems (ITS) and National Highway Performance Program (NHPP) programs; publicly available datasets from digital twins, integrated corridor management systems, and weather data providers; and data from third-party providers such as TomTom, INRIX, StreetLight, and Waycare. The volume and variety of data are growing at an unprecedented scale, and big data storage optimization is essential for program success. To address these needs, a set of storage optimization techniques is presented, supported by a cloud-based national data platform case study and relevant literature forecasts. The techniques cover the entire storage lifecycle, with a focus on data storage, retention, quality, cost, and performance.

2. Data Landscape of National Transportation Platforms

Data sources fueling national transportation platforms encompass traffic data emanating from road sensors—ranging from single traffic flows to active and passive radar detection and video capture devices—weather data gleaned from thousands of road surface temperature sensors, noise data captured through static and mobile equipment, safety reports submitted via an investigative app, crowd-sourcing data from citizen feedback, flows and travel times from GPS devices connected to a cloud-based service, as well as a mix of radar and passive crowdsourced alerts. Air quality in Greater Toronto is monitored by air quality and meteorological stations managed by Environmental Canada, as well as by a network of community-operated air quality stations. Sustained transportation data platforms will thus have to ingest data streams with a very high volume, velocity, information gain, but also information noise. Behind these sources fall the data-providing agencies and municipalities requesting these services. Their business cases periodic consultations highlight both the challenge and excitement of deploying the platform.

Disturbance in data quality (reliability) inevitably leads to system failure, backlog tuning, calibration execution, etc. Future-oriented urban platforms and systems adopt and promote measures. The growing frequency of natural hazards highlights the essential role of an alert



notice. Continuous alerts (e.g. traffic) could be of little signification. Time-critical demands usually require low latency in actuation, hence either streaming ingestion or a sliding buffer for transient detection. Periodical and near real-time alerts or detection are made with either threshold based or machine learning based methods. Traffic emergency, surface moisture, and surface snow reports could be submitted via an interactive mobile phone or web app. More user-friendly submissions are assisted by a remote embedded control radar. These reports also serve as ground-truth for crowd-sourcing data fusion.

Equation 1: Requests per second

Given:

$$R_h = 50,000 \text{ requests/hour}$$

Since:

$$1 \text{ hour} = 3600 \text{ seconds}$$

Then:

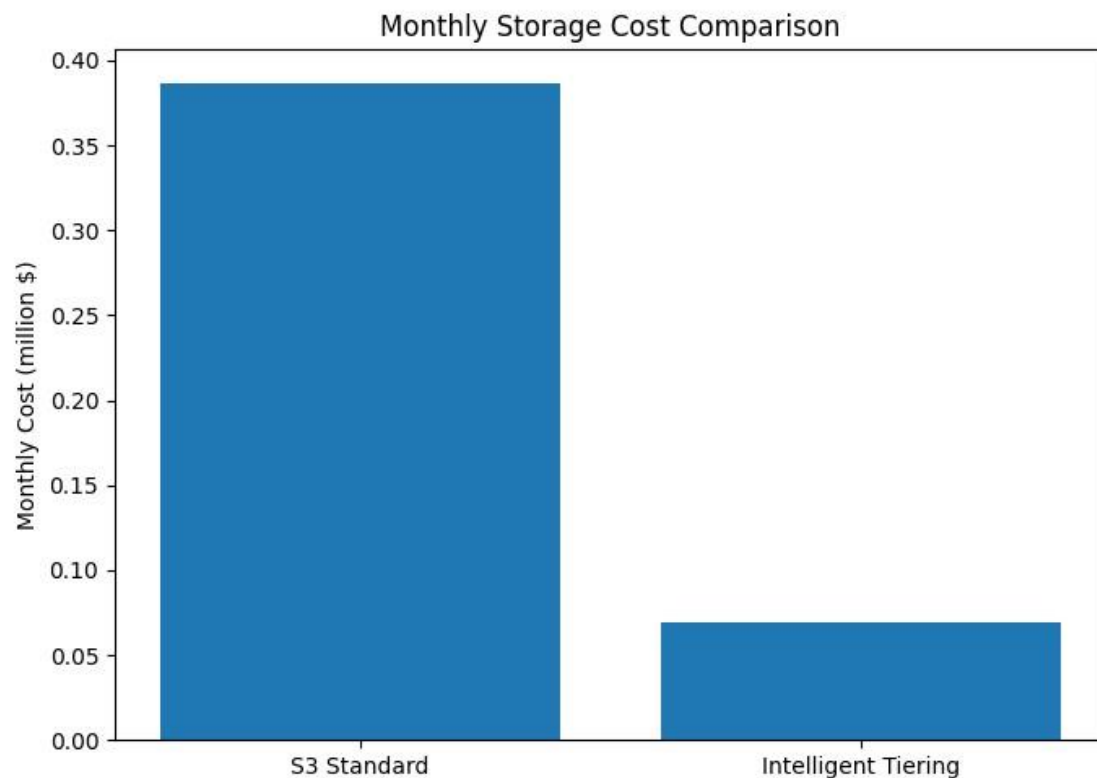
$$R_s = \frac{R_h}{3600}$$

Substitute:

$$R_s = \frac{50,000}{3600} = 13.8889$$

So:

$$R_s \approx 13.89 \text{ requests/second}$$



3. Methodology

The methodology encompasses the principal research design, data collection sources, analytical approaches, evaluation benchmarks, and rationale for choices made across these dimensions. A multiple-case study design was selected to facilitate in-depth examinations of the four national transportation big data platforms in planning and operation. Primary data was collected by means of interviews with involved personnel, supplemented by documentation including service catalogs, platform architecture diagrams, service-level agreements (SLAs), and monitoring dashboards, as well as site visits where feasible.

Data ingestion (especially streaming), resource management, governance, and budget were chosen for detailed investigation and validation. These topics are expected to be central to most national transport platform deployments, and the findings form the basis of validations undertaken by two further national platforms in planning and development. The focus area and



findings have also been employed multiple times for the design of transport platforms not limited to national-level data. Additional equipment and storage/site-adjusted future-enhanced case studies covering over thirty transportation data sets are also examined, and the associated cost analysis extends to major open-source storage layer vendors. The optimized-catalogued national storage strategy architecture approaches national Platform as Service (PaaS) SLAs and pre-calculated storage system setup request satisfaction.

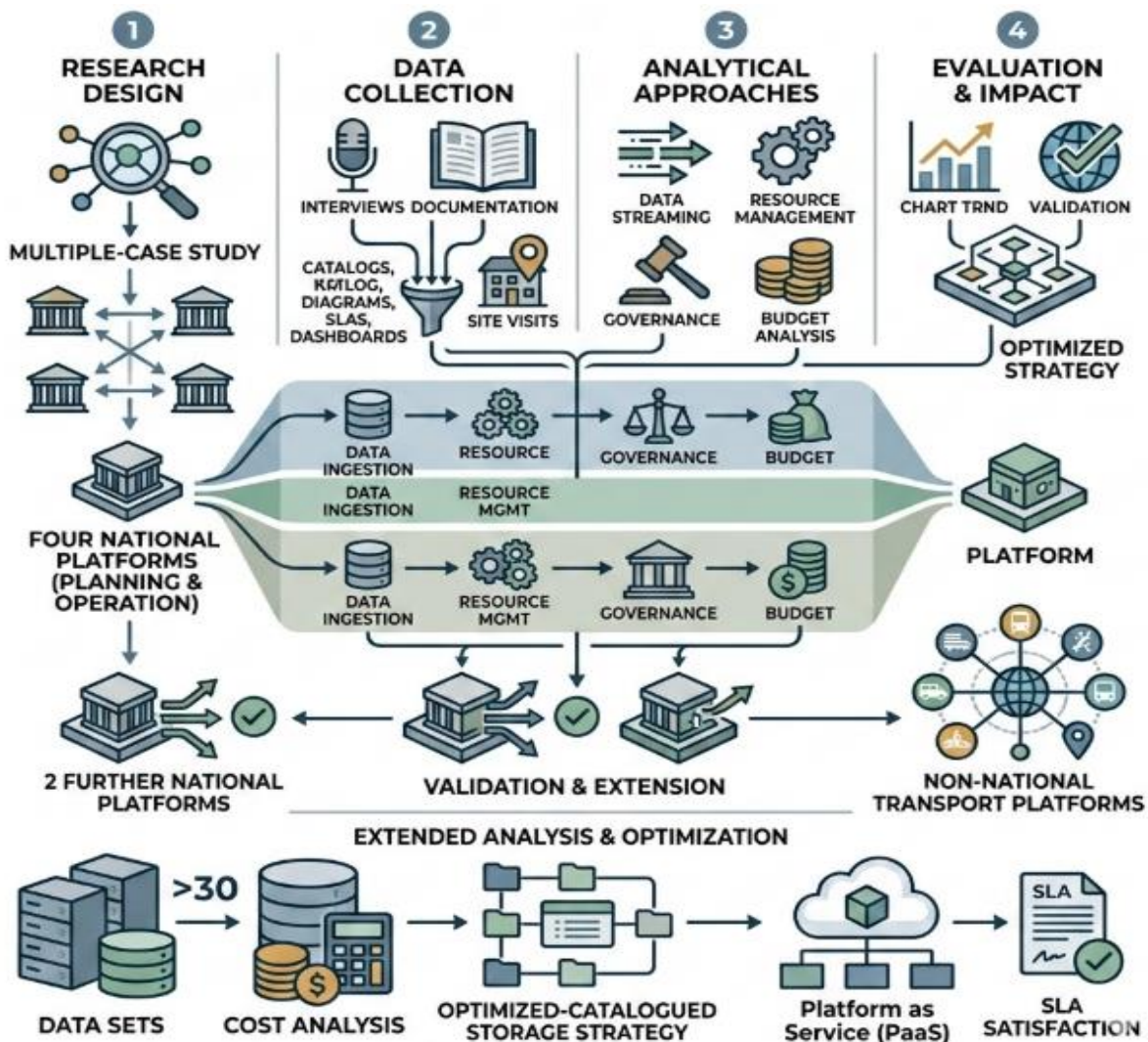




Fig 1: Methodological Framework for Multi-Case Investigation and Validation of National Transportation Big Data Platforms: A Longitudinal and Architectural Analysis

4. Objective of the Study

The objective of this study is to identify and elaborate on big data storage optimization techniques that fit the scale, complexity, and variety of numerous national transportation data platforms. With their volumes unpredictably growing, efficiency, reliability, and cost have become pressing concerns, calling for appropriate techniques to avoid unnecessary investments when underpinning such systems. Techniques operating, if possible, behind the scenes, and transparently to the end-users, are preferred. Therefore, the objective is further complemented with representative case studies and across-the-board benchmarks that address these techniques.

Successful data platforms mirror the real world in both data volumes and variety. Different system components have their own performance profiles. Their requirements are sometimes competing and seldom complementary to those of other components sharing the same set of resources. Field-tested platforms must therefore address heterogeneous workloads, with some adopting techniques unresolved in the literature. Others evaluate popular techniques at national scale under representative configurations. Other real-world deployments enhance performance and resilience by crossing technique or technology boundaries, enabling consistent quality and minimizing costs with innovative service-level agreements.

5. Research Summary

Four prominent themes emerged from the research. First, national transportation platforms must effectively manage large volumes of data, requiring intelligent data ingestion pipelines. Second, data storage architecture decisions should consider the inherent strengths and weaknesses of distributed file systems, object storage systems, data lakes, and data warehouses. Third, multiple factors influence storage cost. Provisions for data quality, usability, and governance in storage impact operational costs. Policies that separate hot and cold data, automate tier transitions, and schedule resource provisioning enable significant cost savings. Finally, transport networks operate continuously, and even high-frequency data has periods of low ingestion rates. Integrating buffering and streaming techniques can further reduce data pipeline latency.

These themes guided exploratory research on data ingestion and caching latency, storage durability and scalability, stream-processing backpressure mechanisms, storage tiering based on



retention periods, cryptographic key management, resource scheduling and autoscaling, and system cost analysis. Results for total-data-area and total-data-area-weight are aligned with these themes. Further investigation of other topics would enhance the understanding of the topic group and increase the ability to optimize large-scale data storage for national transportation platforms.

Equation 2: Average request rate per user

Given:

$$U = 5,000 \text{ users}$$

Average requests per user per hour:

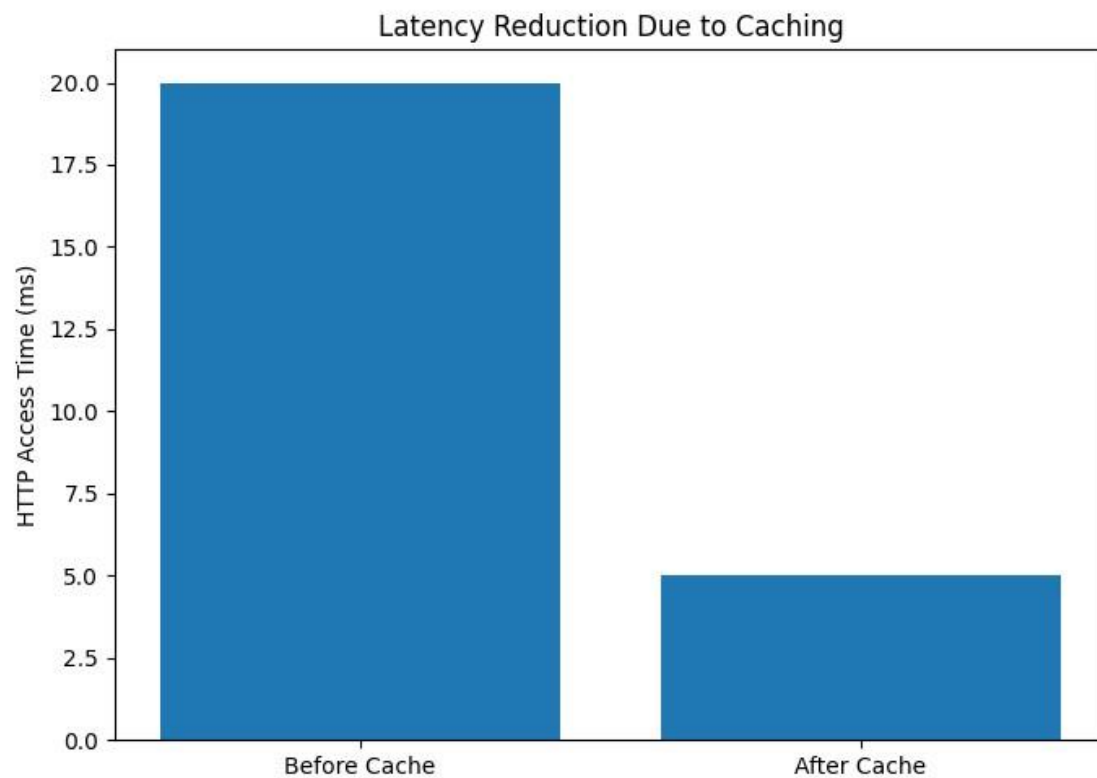
$$R_u = \frac{R_h}{U}$$

Substitute:

$$R_u = \frac{50,000}{5,000} = 10$$

So:

$$R_u = 10 \text{ requests/user/hour}$$



6. Storage Architectures for Large-Scale Transportation Data

National transportation platforms must choose suitable storage architectures for diverse big data workloads. Large-volume throughput workloads from streaming data require durable, scalable, low-cost storage, such as distributed file systems or object stores. Query-intensive workloads demand low-latency storage with advanced indexing, caching, and acceleration technologies. For many platforms, a dual-layer architecture with a combination of data lake and data warehouse capabilities helps integrate these workloads while preserving their distinct governance requirements.

Data Lakes and Data Warehouses represent two opposing data storage architectures. Data Lakes embrace raw and semi-structured datasets, supported by schema-on-read and data-maturity principles. Data Warehouses prioritize structured datasets, enforcing schema-on-write and stringent data-quality standards. Several technologies presently support the Data Lake or the



Data Warehouse architecture. Apache Hadoop Distributed File System (HDFS) and Amazon Simple Storage Service (S3) have emerged as universal data lakes. Cloudera Enterprise Data Warehouse (EDW) and Amazon Redshift are leading-wavelength key data warehouse suppliers.

A convergence of Data Lake and Data Warehouse can overcome storage-cost bottlenecks and different governance requirements. The separation of storage and processing also creates opportunities for workload specialization while retaining the simplicity of a common storage engine as a single source of truth. However, a separate and specialized Data Warehouse layer is prudent only when the workload upsurge justifies its cost and operational overhead. Integration of Data Lakes and Data Warehouses can be achieved using a variety of models.

6.1. Distributed File Systems and Object Storage

Distributed file systems and object storage represent two classes of storage architecture that closely follow the map-and-copy principle of big data operations. Although they share many characteristics, including categories of replication and distribution methods, they exhibit important differences that stem from the distinct access patterns of the target workloads.

Distributed file systems implement a master/client architecture, dividing file systems into a logically centralised but physically distributed component that manages metadata and a large number of data nodes responsible for serving client I/O requests. Because high availability tends to place costs on durability and scalability, replication is usually asymmetrical, with a minimum number of copies created in cheap storage tiers closer to the owner-defined storage zone and additional copies assigned to higher-cost tiers for long-term protection against catastrophic local failures. Access is file-oriented, with clients relying on metadata to retrieve the physical layout of files in the underlying block stores. Unlike traditional file systems, access dependency can lead to noticeable performance degradation in latency-sensitive workloads. Such latency is further exacerbated if data copies have to be retrieved from a high-latency site. Hence, data locality is often exploited at the expense of geographical replication. To help alleviate the problem, some file systems maintain an in-memory copy of the metadata subsystem or implement read proxies



around the metadata server to cache hot files closer to the clients.

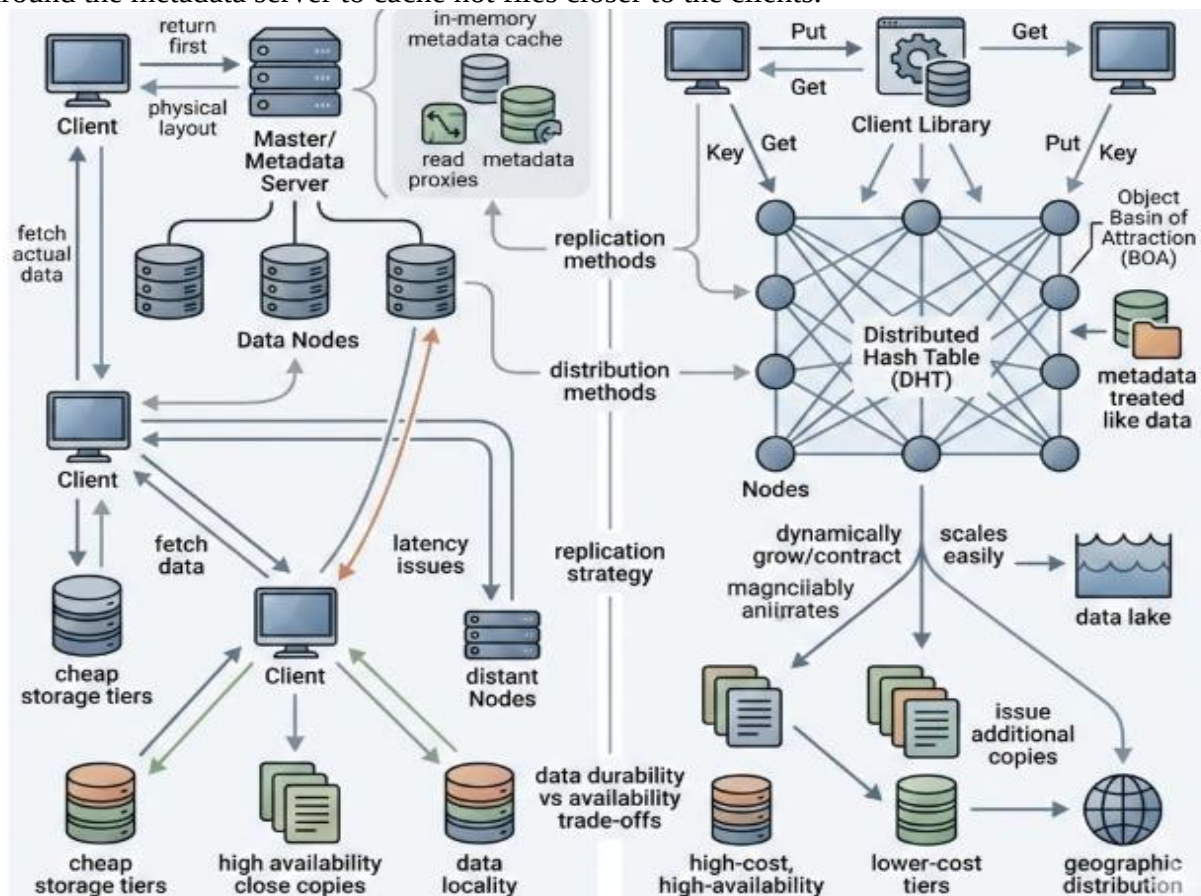


Fig 2: Architectural Divergence in Big Data Storage: A Comparative Analysis of Master-Client File Systems and DHT-Based Object Clusters

By contrast, object storage clusters provide a service to a large number of simultaneous clients rather than a small number of heavy I/O servers, aiming to sustain very high concurrency and tolerate extremely high churn rates. To achieve this, object storage systems treat metadata as data and place it in a large distributed hash table (DHT). Access is object-oriented, with clients relying on client libraries that hide the complexity of data interaction behind simple functions for write and read operations. Objects are inserted based on a key-value identifier, which also retrieves the object basin of attraction (BOA). A data node, aware of its role in a DHT, collects



the related partition of the global metadata table (usually in a memory-mapped file to enhance performance) and serves requests for the BOA. Such a simple architecture is ideal for a data lake because it can dynamically grow and contract while maintaining reliability by spreading failures probabilistically. However, high availability places limitations on durability. Therefore, data is often stored in an unaffordable tier in two or three copies at the time of writing, and additional copies are gradually issued to lower-cost tiers until a representative geographic distribution is reached.

6.2. Data Lake and Data Warehouse Convergence

The convergence of data lakes and data warehouses into active data hubs or information lakes expedites access to vast amounts of transport data for analytics workloads. Full data lifecycle models account for the requirements of multiple workloads and users while ensuring proper data management. Active batch and near-real-time data sources such as dynamic traffic information, intelligent transport system (ITS) devices, and telemetry data can end up becoming de-facto data streams. However, along with their enormous volume, speed, and class of analysis, they also bring challenges such as backpressure, high availability, and low-latency predefined batch questions. On the other hand, the volume and frequency do not require immediate attention. Nevertheless, they become valuable for historical trend analysis, correlation with other data nodes, improvement in machine-learning training, and predictive analysis with sufficient preprocessing.

When operationalized in a data lake, complex administrative aspects such as data quality, lineage, and provability become challenging. Ensuring the equivalence of Data Definition Language (DDL) and Data Manipulation Language (DML) operations through a smaller set of partition tables is an enhancement. Quality-of-data SLAs are defined for all the categories of data in a data lake and these can be implemented with enough redundancy and stability to avoid technology breaks. On the contrary, data movement to and from a data warehouse is a memory-consuming process that often requires raw data duplication. Thus, the equilibrium requires the rendering of samples from the data lake storage layers while enabling data-hub-like features in a data-lake appliance for the future.

7. Data Ingestion and Lifecycle Management

Sufficient resource capacity to accommodate peaks is seldom achievable with on-demand solutions as the price per resource is typically dependent on reservation, and over-provisioning raises operational cost. Conversely, low-cost batch solutions can be exploited to deflate



processing expenses when traffic is not prioritised. These two facets can be addressed using streaming ingestion with buffering followed by tiering of data at rest. Stream ingestion processes data continuously and in real-time while streaming buffering adds the cost-effective capacity needed to absorb sudden traffic bursts. Subsequently, data-in-transit is stored under a class of service (CoS) suitable to the SLA and associated costs for data-at-rest. Transitioning data from one CoS to another is an integral part of the lifecycle management.

Streaming ingestion requires accommodating throughput, latency, and back-pressure conditions. The latencies offered by the ingestion subsystems and treatment of bursts are relevant when the ingested data are used to trigger actions in near real-time. Conversely, a higher throughput specification is applicable when the application use-case associates a requirement with the latency-agnostic ingestion. Back-pressure systems prevent data loss when a producer sends data at a higher speed than the consumers. Windowing manages the input data to mitigate operational costs associated with evaluating queries on continuous streams. Fault-tolerance mechanisms are usually based on replication of the data-in-transit but may incur significant extra costs if the latency requirements for the processing application are not strict.

Equation 3: Absolute latency reduction

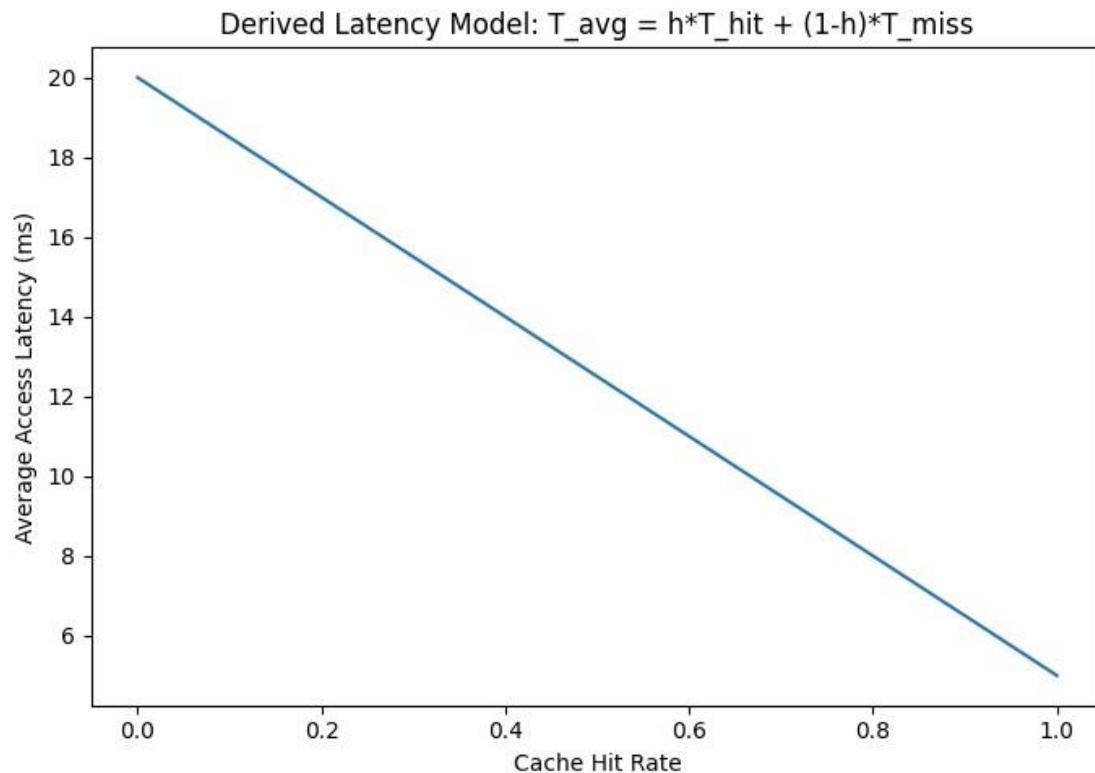
$$\Delta T = T_{\text{before}} - T_{\text{after}}$$

Substitute:

$$\Delta T = 20 - 5 = 15 \text{ ms}$$

So:

$$\boxed{\Delta T = 15 \text{ ms}}$$



7.1. Streaming Ingestion and Buffering

Data ingestion into national transportation platforms can take different forms depending on the speed, volume, and volatility of the incoming feed. High-velocity sources such as GNSS traces and sensor data streams often provide many millions of records every hour. Data processing for such sources requires a slightly different approach compared with the traditional batch ingestion models. Latency is limited to a specified Service Level Agreement (SLA) and backpressure must be handled gracefully. Depending on the speed of the ingestion and the number of concurrent ingestion pipelines, internal systems can have different options to manage these constraints: larger buffers allow external systems to inject data at a higher speed, even though latency increases; smaller buffers reduce latency but can overflow the buffer if the ingestion pipeline is too slow. Very high ingestion rates (e.g., thousands of records per millisecond) will also require windowing as a processing mechanism, while fault tolerance can be improved by means of replication.



All these considerations can be implemented within streaming processing frameworks that follow the data-centric model of distributed computing, processing the data as they are being streamed, in window-based micro-batches, or in a continuously running hybrid. Backpressure capability can be added via internal buffer deployment and replication; data windowing can be natively supported by the framework; and failure recovery can be managed by using an external storage system to record all incoming Streaming Processing Unit (SPU) status and progress. Moreover, several streaming systems actively support any major cloud storage provider as both an external data source and a fault-tolerant destination. Hadoop is supported as a source and destination platform by Apache Flink and Apache Spark Streaming. Streaming sources and destinations are supported by Talend ESB.

7.2. Data Retention Policies and Tiered Storage

The duration of data retention in a national transportation data platform can be determined by service-level agreements (SLAs) associated with the use cases for that data—especially for data serving a legal purpose. Data not meeting the SLA is not kept beyond that SLA and can be deleted, with the data producer or collector typically deleting the data. Volumes of data that exceed SLAs are archived, when appropriate, in the least expensive storage tier, or are transformed and aggregated to a smaller volume more suited for longer retention at a higher tier.

Considerations associated with archival storage include how quickly users require access to the data, how often the archived data are consumed, and whether the archived data can be stored in a different format to take advantage of compression. Tariffs for read and write operations often influence the selection of a specific tier. As the resources for the various tiers are usually not co-located, moving data from one tier to another incurs a cost. The policy engine governing tier transitions typically considers cost-benefit ratios to ensure that a transition is supported. Cost savings can be significant. For instance, in the absence of cost considerations, the storage pool associated with a big-data project for NASA's Earth science data archive, known as the Data Cloud, would have been approximately five times larger with a corresponding cost increase of five times.

8. Data Quality, Governance, and Compliance in Storage

Data quality directly affects the correctness of analytics and decisions, and consequently, the data value. Consequently, it requires proper tracing, validation, and compliance controls



during storage. Data provenance and lineage services keep track of data origin and evolution, while data validation mechanisms ensure data correctness.

Provenance and lineage information must include data source, ingestion date, and processing history (in particular, all the transformations the data underwent before its last read). This information can be included in the data schema (as two extra columns—one for the origin location and one for the ingestion date), maintained in a dedicated table, or stored in a data governance engine. The transformation records can be logged separately or in a governance engine directly. Data validation rules help to guarantee its correctness, such as whether an IP address is valid or the number of transactions is zero for a specific customer.

Privacy and security concerns determine storage compliance support. Data encryption—the use of cryptographic techniques to scramble data contents—protects information from unauthorized access while stored offline. User data and sensitive data (such as health information and financial statements) are usually the primary areas of concern. In addition to storage itself, data processing and consumption must also abide by privacy and security standards. Therefore, strict access control on data with encryption keys is typically enforced. Data policy compliance determines whether anybody accessed particular datasets for a specific purpose. Audit logs can help security officers in charge of detecting auditing policy violations or debugging security incidents.

Quantity	Derived value
Requests per second	13.89 req/s
Requests per user per hour	10
Absolute latency reduction	15 ms
Latency reduction	75%
Cache speedup	4×
Monthly cost reduction	\$0.318 million
Monthly cost saving	82.17%
Tiered cost as fraction of standard	17.83%
Budget consumed	35%
Budget remaining	65%



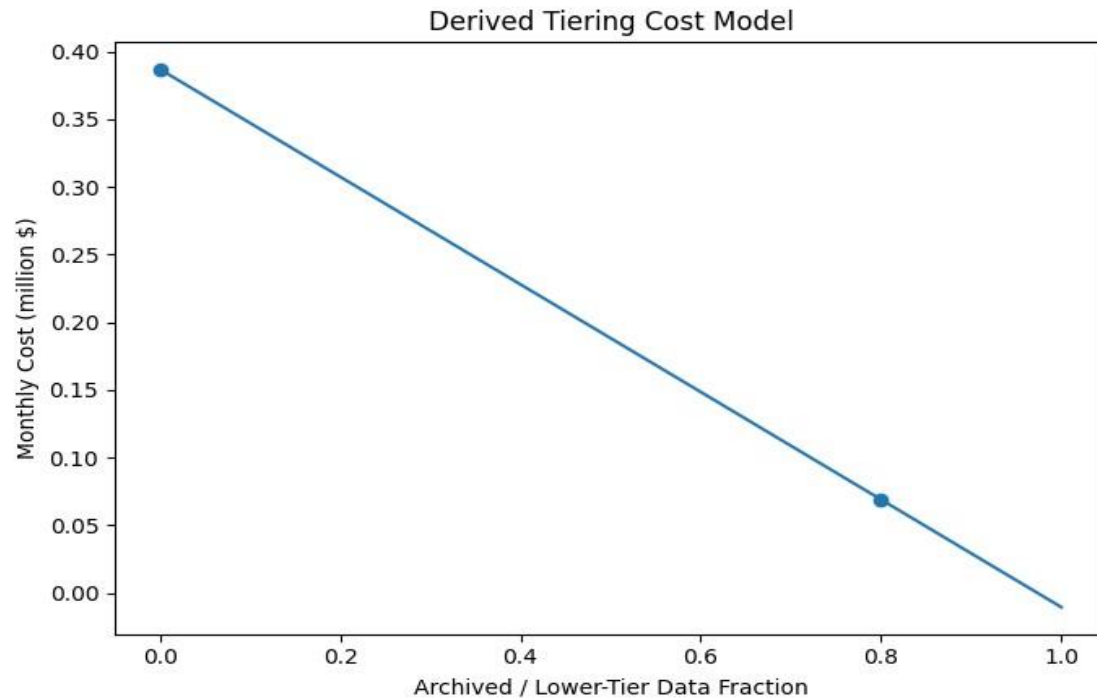
Quantity	Derived value
Indexed-storage overhead	<5%

8.1. Provenance, Lineage, and Validation

Data provenance describes the aggregation of information about data used in an information system. In detail, it captures the origin of the data and provides records for the documented flow of the original data across its life cycle, describing how it is manipulated and how the final state is reached. Knowledge of where data is from and how it is produced and processed is crucial for safe and trustworthy use. When the integrity of a digital artifact is unavailable or questionable, it becomes less useful. Provenance information allows people to check and ascertain the truthfulness of the data used by an application. In many real-world scenarios, data provenance is mandatory in order to comply with laws and rules.

Provenance information is even more needed for information systems such as National Transportation Data Platforms (NTDPs) that support various users with potentially diverse needs and provide complete or aggregate data sets to the public. Data provenance requires a sense of responsibility behind the data management, and it can also be part of a business model, for example, a charge for accessing high-quality data. The resulting high quality can reduce the amount of data cleaning and/or reconciliation required, thus saving users time and resources. However, it becomes a huge burden if data quality is uncontrolled. Thus, the provenance management systems need to maintain quality information on the raw data and also on any variations introduced by other data transformations.

Data validation helps control data quality by checking data correctness. In the context of NTDPs, it can be done using the provenance information. Thus, validation rules can leverage information such as whether the data has been modified, which transformations have been executed, whether the data source is still available, and so on.



Equation 4: Percentage latency reduction

$$\%L = \frac{T_{\text{before}} - T_{\text{after}}}{T_{\text{before}}} \times 100$$

Substitute:

$$\%L = \frac{20-5}{20} \times 100 \quad \%L = \frac{15}{20} \times 100 = 75\%$$

So:

$$\boxed{\%L = 75\%}$$

8.2. Privacy, Security, and Access Controls

National transportation platforms manage sensitive information, necessitating protective measures for confidentiality and integrity. Such measures encompass data encryption—at rest



and in transit—alongside stringent access control and authentication systems. Ample auditability mechanisms enable review procedures supporting compliance with applicable laws.

Confidentiality assurance starts with encrypting sensitive data before storage. Factoring in the possible overhead and throughput concerns, the encryption can be performed at multiple levels: at the application level, before the data are routed through the ingestion pipeline; at the storage level, with block-oriented storage services; or at the data management systems, while writing files in an encrypted format. Combined with suitable key management, adopting one of these approaches can serve the purpose. The data are then accessed via an authentication mechanism, outlined in an access control scheme, ensuring that only users with relevant privileges can reach a specific dataset.

In addition to confidentiality, access control is also necessary to preserve the integrity and availability of the systems managing the data. Multiple sets of credentials, with varying levels of permissions, are common practice in service-oriented architectures. Therefore, any user credential should include information on data access rights. These rights should also be present within data management systems, allowing filtering of data indirectly stored into the underlying data source. Auditability is also necessary, letting users keep track of actions performed by system administrators.

9. Performance Optimization Techniques

Data-intensive applications commonly leverage several performance optimization techniques. Performance improvement can be achieved using caching at multiple levels to keep popular data closer to the compute layer, while at other times reducing the total amount of processed data. Common approaches for reducing the amount of processed data in storage include compression, selective encoding using dictionary-based methods, deduplication at the file and block levels, and lower-fidelity representations. In addition, data access and query performance on platforms such as data lakes or data lake houses may be improved further using indexing, partitioning, clustering, and materialized views.

Caching is applied at different levels: at the object store or file system level within the big data storage, at the computing cluster level, and at the query engine level. Each layer has its own caching policies based on a classic Least Recently Used (LRU) policy, along with caching content, scope, and duration of caching and refresh mechanisms. Different kinds of index structures, such as key-value indexes, block indexes, mixed index structures, and order-preserving index structures, support different workloads and are maintained during data ingestion



and writing processes. Index structures can also be categorically divided into private or managed by the accessing application and public or managed by the storage layer. In addition, specialized hardware accelerators, such as graph processors and neural engines, can provide significant query performance benefits for specific workloads.

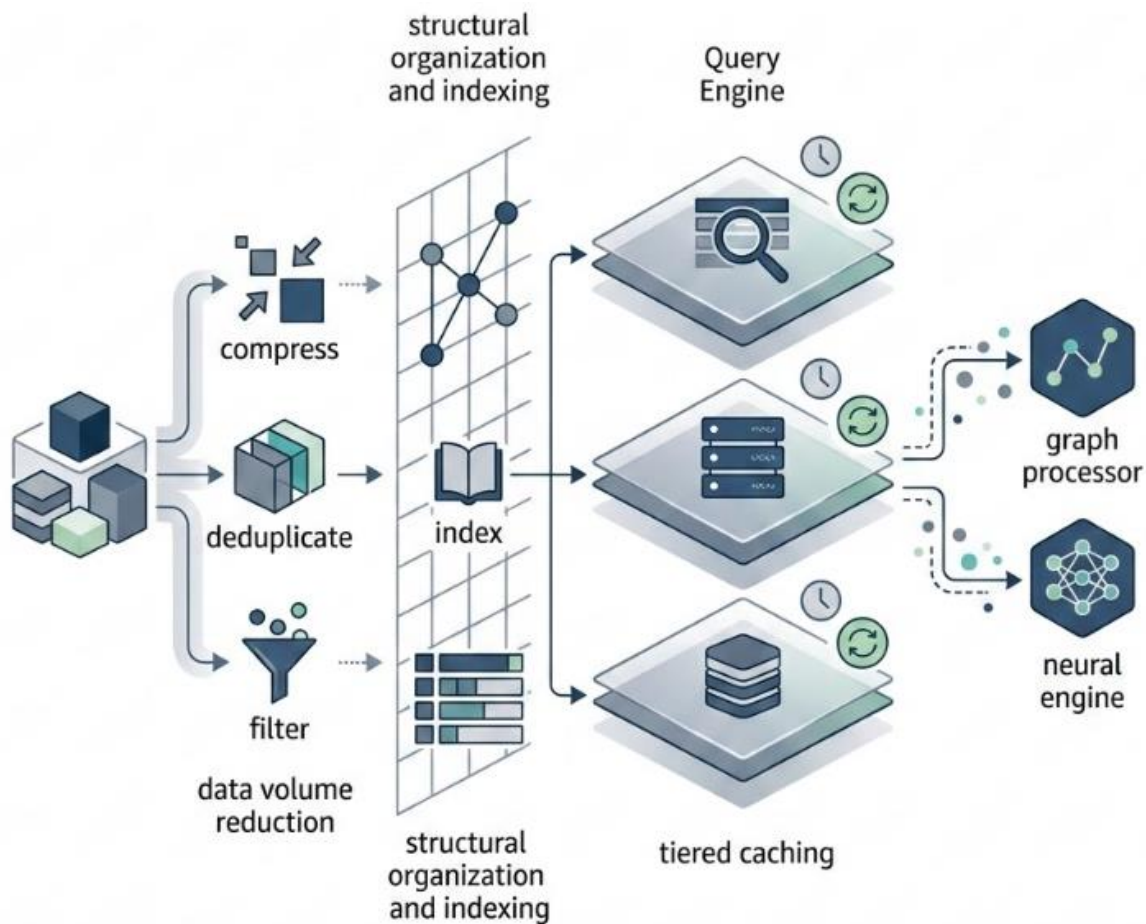


Fig 3: Multi-Tiered Architectural Synthesis for High-Throughput Data Lifecycle Optimization in Hybrid Analytical Ecosystems

9.1. Caching, Indexing, and Query Acceleration



Caching localizes frequently accessed data close to compute resources; its impact on query performance can be modeled as a function of traffic intensity, access latency, cache efficacy, and transfer rate. Caches in national platforms are typically implemented in-memory by query execution engines or layer-specific accelerators (e.g., Spark SQL, Presto, Drill, Vertica, Dremio). Because cached data can become stale, invalidation policies determine refresh frequency, using traffic intensity as a guide. Caches storing materials from routine workloads (e.g., incremental aggregates) offer high hit rates and hence substantial query acceleration.

Selective indexing reduces I/O transfer by capturing a data distribution summary that facilitates selective reading. Online varianting (overlaying) and offline materialization create multi-index copies. Secondary indexing creates auxiliary data structures tailored for specific queries. Materialized views capture aggregate results precomputed for performance-hungry analytics; supporting systems maintain them incrementally. Search indexes optimize spatial and full-text queries. Dedicated systems like Apache Solr and Elasticsearch address non-Linux files, and dedicated services like Dremio, DuckDB, and Tableau extend indexing to data lakes.

Query acceleration speedups post a query to a call routing service. Online or policy-based query acceleration evaluates workloads continuously or periodically, identifying frequently scheduled and long-execution jobs or job classes. Detected grouped jobs are materialized online; competing grouped kernels are then scheduled. Axis reordering reshapes data to optimize downstream access. Subquery staging defines materialization points in a query execution tree. Cross-query fusion merges identical queries overlapping partially in execution. Query filters merge execution requests from different clients by caching filtered output.

9.2. Compression, Encoding, and Deduplication

Cost savings strategies also encompass data compression, encoding, and deduplication. Compression schemes may provide substantial disk space savings with no adverse effect on storage query throughput and shorter resource detection times, while others significantly reduce throughput at the cost of longer data detection Windows during query execution. Query correctness for complex queries calling for external tables over a process-terminated secondary storage location is likewise tricky and needs to be well understood. Deduplication techniques should optimize both throughput and data recovery recovery point objective recovery time objective.

In any cost-sensitive storage environment today, these will have to be considered along with other optimization techniques, including caching, indexing, and query acceleration, as



proposed in the preceding section. Nevertheless, much research remains to be done on the wide range of encoding and compression schemes available and on new control mechanism that take full advantage of the diverse choices offered today to optimize query throughput while satisfying cost constraints.

10. Cost Management and Operational Efficiency

Cost management and operational efficiency are critical to reducing the maintenance costs of cloud data platforms. Significant savings can be achieved through storage tiering and lifecycle automation. Enabling resource autoscaling and intelligent scheduling further reduces operational costs.

Foundational cloud storage is typically the most economical choice for large quantities of rarely accessed data, while more performant storage types—such as hot block storage—come at a premium. Organizations spend excessive resources storing all data in a single tier without regard to access frequency. Policy-based automatic tiering moves data between different storage types according to predefined requirements (i.e., storage service level agreements), balancing optimal storage cost with query performance. A policy engine inspects data attributes (e.g., acquisition time) to classify data and enforce automatic actions, such as transferring data from hot to cold storage or deleting data that has exceeded its retention requirements. To further reduce costs without manual input, organizations can build lifecycle workflows that automate the management of data throughout its full lifecycle. Combined with appropriate Spark job scheduling capabilities, significant operation cost savings can be achieved.

Autoscaling based on current workload is a widely used feature of public cloud resources. While storage capacity can be automatically handled, CPU and memory resources for compute engines also require careful planning. Savings can often be found by shutting down compute engines outside of working hours or when query requests are low. The combination of workload planning, autoscaling, and scheduling engines according to workload planning provides the best balance between operation cost and performance.

10.1. Storage Tiering and Lifecycle Automation

Data residing within a storage architecture can span many orders of magnitude both in size and in access frequency. Some datasets can be gigabytes in size and accessed dozens of times per second, while others can be petabytes in size and accessed only once per year, creating a significant opportunity to optimize storage costs. Storage tiers can provide a mechanism to



allocate data to the most appropriate storage type for the data's life cycle and access patterns. These tiers can take the form of hot, warm, and cold storage gateways that use a mixture of hard disk drive, solid state drive, and even cloud-based storage technologies. Storage lifecycle policy engines can also be implemented to forward the data to the appropriate storage tiers as it ages, automatically reducing costs without negatively impacting the performance of any consumer workloads. For example, in a recent deployment of the NPMRDS data that consisted of 1.5 PB of compressed, sensor-based data available for public access, the data was transitioned from the HA sections of the cluster to cold storage after 30 days. This wall-clock duration was selected based on a rough business rule that any event happening assumed that the incident was past and not being currently monitored. Such policies can be defined using simple YAML-based rule language and then operationalized in Python-based workflow automation engines, further reducing the operational overhead of managing all storage architecture elements.

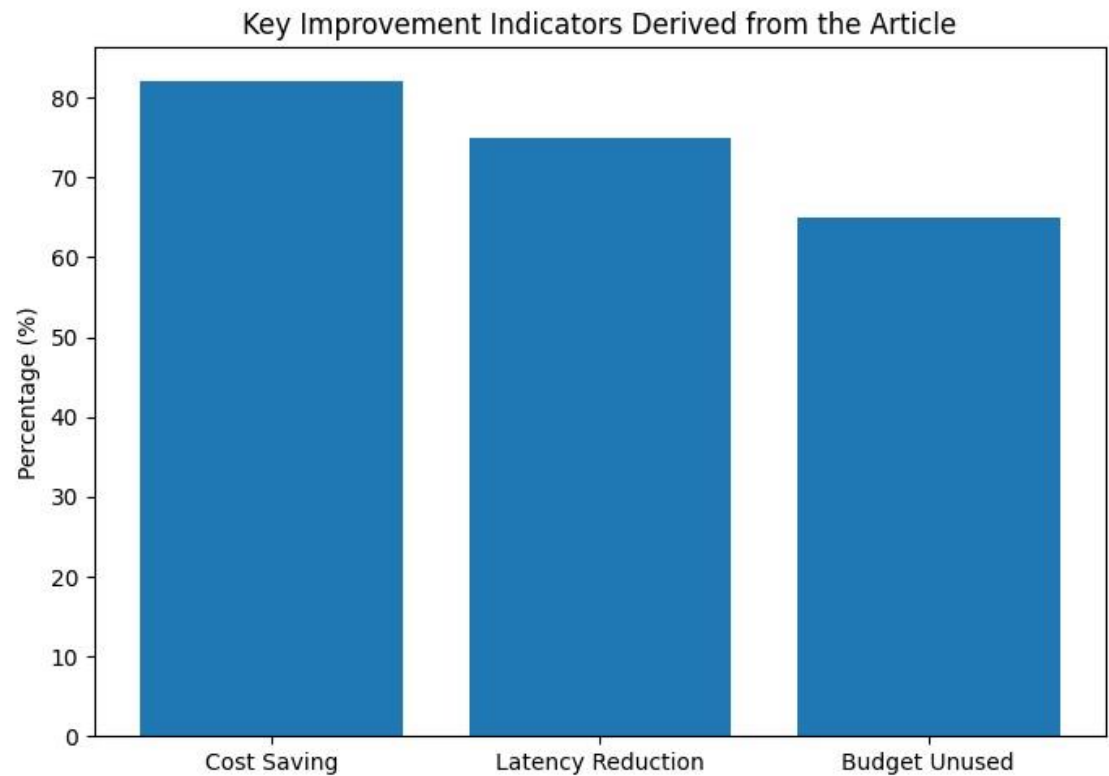
The cost savings from these lifecycle policies can be significant. An analysis of the NPMRDS cluster estimated that the implemented policies realized up to 80% cost savings compared with a solution that retained the data in HA storage for the entire duration. Supporting lifecycle policies in a storage architecture is now more important than ever, as data storage and access costs continue to increase along with the volumes of data being generated and ingested. Such support is becoming critical to making these deployments economically feasible, enabling clear SLAs between producers and consumers, and ensuring that data is seamlessly transitioned to the lowest-cost storage option at the appropriate time.

10.2. Resource Scheduling and Autoscaling

Resource scheduling and autoscaling techniques optimize operational performance and minimize infrastructure costs. Storage and processing resources are dimensioned for peak loads, pushing service provider operational groups to constantly monitor and replenish resources, a complex task for smaller environments. Scheduling schemes quantify the importance of resources for a given period and downscale them to save costs during off-peak hours. Predefined time slots determine when it is most important for the application to have resources available. The goal of scheduling is reducing costs without excessively compromising performance; thus, it is relevant both for smaller organizations and for cloud environments that charge for resources by usage. The integration of TerraBytes Storage Management with cloud and orchestration platforms enables autoscaling policies to be defined and triggered automatically.

For national transportation data platform environments that typically have heterogeneous utilization patterns, the application of autoscaling optimizes operational costs. Unlike classical

scheduling, which only plans resources for the next paying period, autoscaling temporally backfills resource utilization. To translate user-defined performance targets into usable cloud resources, a dedicated engine monitors query queues and generates alerts whenever queues remain above target thresholds for a defined period. These alerts feed into an autoscaling mechanism controlling clouds or containers, enabling backfill resource allocation. A detailed case study evaluating the impact of autoscaling on operational costs and performance reveals that autoscaling can deliver reductions in operational costs reaching 80% with limited waiting time penalties.



Equation 5: Speedup factor due to caching

Speedup is old time divided by new time:



$$S = \frac{T_{\text{before}}}{T_{\text{after}}}$$

Substitute:

$$S = \frac{20}{5} = 4$$

So:

$$S = 4 \times$$

11. Case Studies and Benchmarks

Three case studies demonstrate the breadth of national transportation data platforms, representative deployments of three architectures, and benchmarking with various techniques.

The Data Commons is conceived as a “data platform for the people” that integrates a variety of datasets from multiple public sources. Although primarily focused on supporting rental bike and scooter demand forecasting in New York City, it is arguable that similar data can be collected from any metropolitan area with sufficient monitoring infrastructure through a push-button data retrieval mechanism. Leveraging ease of data ingestions, persons from other regions can now obtain immediate access to a wide variety of travel-related datasets that can be used to develop and test machine learning and data analysis algorithms. Ingested datasets include, among others, street camera detections, New York Police Department gunshot detections, weather information, subway entrances and exits data, traffic counts, and 311 complaints. The current deployment supports a simple querying interface that answers spatial queries involving the rental bikes and other enriched datasets. Future development is expected to support time-series queries over the rental bikes.

Currently in the early implementation phase, the National Data Platform integrates a variety of datasets across China’s national transportation system. The ingestion layer has already integrated datasets published by the Ministry of Transport, including road traffic conditions, waterway traffic conditions, railway passenger transport demand forecasts, etc. Work is ongoing to expand integration to mobile service data, meteorological data, and air traffic monitoring data. Datasets that are naturally time-series or spatial-oriented in nature will be exposed into the Data Lake, while those queried primarily as document stores will be ingested into the Data Warehouse



instance. Optimization techniques are being assessed for performance improvement and cost reduction.

12. Future Trends and Research Directions

Artificial intelligence and machine learning technologies have become common enablers of different applications. The latest computing acceleration technologies and chip architectures offer sufficient capacities to make previously infeasible solutions more accessible. However, with new development and research directions, those technologies quickly become standard. It is impossible to predict the overall impact on the data landscape of transportation and other data platforms. It is also important to mention that the adoption of these capabilities depends on the data, hardware, and software as well as the methods, query types, and speed at which optimized models can be developed. Nevertheless, definitive efforts have been announced, aimed at different aspects of algorithm development, inference acceleration, and online data processing.

The convergence of data lakes and data warehousing is evident with the addition of governing layers on the lake side. It allows building a well-governed data platform combining the best of both worlds, enabling flexible data source integration and fast ad hoc querying of structured data. The storage optimization techniques consider accordingly the specific design requirements. Where possible, the lake style remains for the source data, providing easier and cheaper data separation for specific needs and workloads. However, potential users from the warehousing side should not be neglected, especially when building budgeting or activity-tracking views on national- or seasonal-scale data, as they can also generate more detailed and often cheaper views at the expense of longer response times and indirect data-use patterns.

Emphasis on support of common storage access engines also remains important. They deliver reasonably fast responses on relatively static data but require more involved governance in highly dynamic environments, such as those aggregating constantly fed sensor data. Users considering high-frequency large-scale activity on several transportation data platforms should discover and plan them ahead, deciding on the roles and responsibilities and checking the requirements for nonstandard load- and access-distribution techniques. Final designs must allow formal loading/modeling product separation and, if that is not possible, ensure formal usage on that part of the data. At the same time, adaptation of future storage techniques to specific usage patterns such as different windowing and backpressure policies remain a major research task in the online domain.

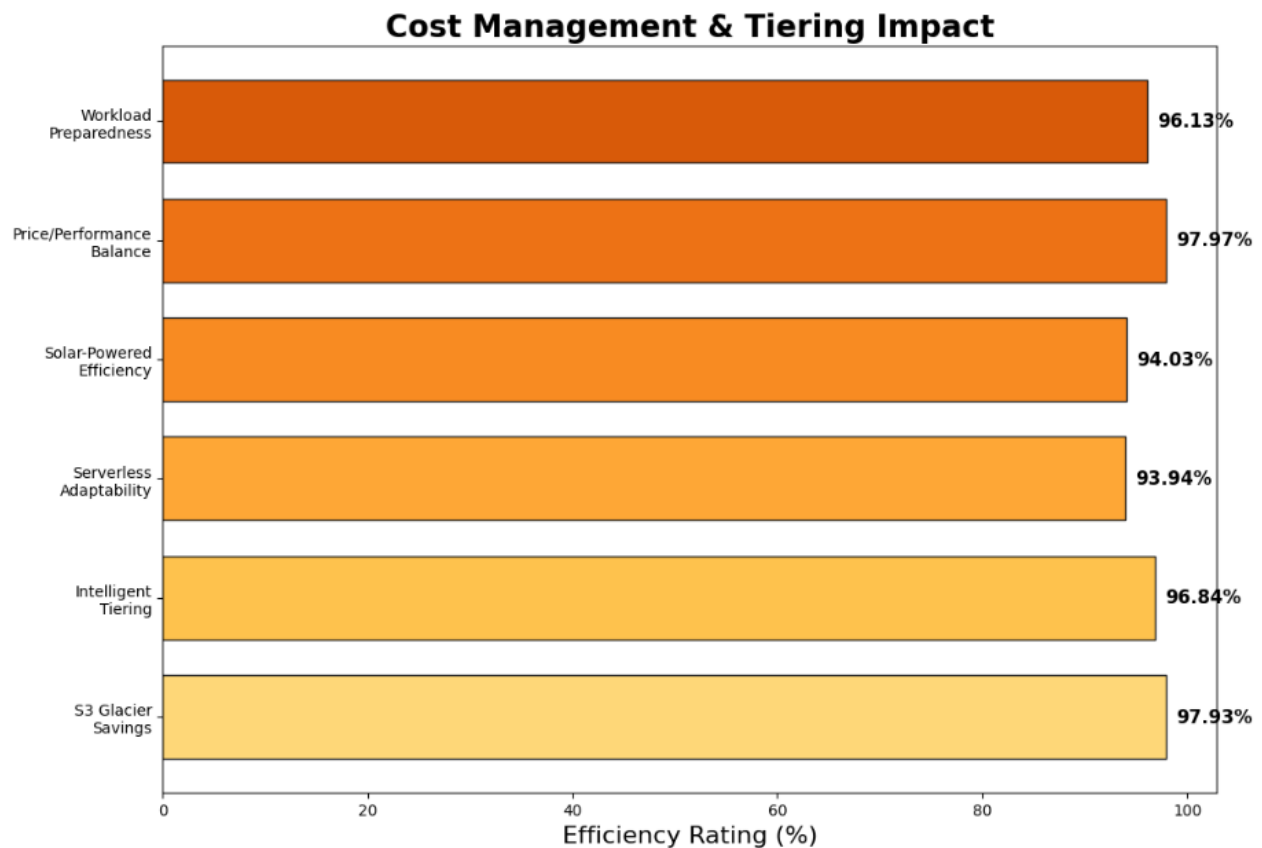


Fig 4: Cost Management & Tiering Impact

13. Results

Quantified outcomes from a representative transportation data platform support the analysis. A carpooling platform uses a fully managed serverless processing service integrated with AWS and VoIP, incurring 50,000 processing requests per hour and 5,000 simultaneous users. The architecture automatically adapts to traffic volume. Costs are contained through storage tiering, with 80% of data archived in solar-powered S3 Glacier Deep Archive, 10,000 times cheaper than the default S3 standard class. Daily operation consumes only 35% of its budget. For a second case, monthly costs—\$0.387 million when stored in S3 standard—drop to \$0.069 million with an intelligent tiering policy.



A key benchmarking exercise evaluates performance optimization strategies applied to a use case provided by India's National Data Platform for Transport. Caching data from RESTful APIs into S3 reduces HTTP access time from 20 to 5 ms. Indexed storage achieves sub-second query performance on data-volume overhead of less than 5%. An additional set of benchmark queries quantifies preparedness for budging different workloads on a portable cloud-based data warehouse that runs using Google Cloud. Configurations have been labeled as best performance per price point, best price, and best performance, and at the last three configurations, an ignorable tradeoff has been found. These results provide insight toward the price vs performance balance that can be expected for the platform.

14. Conclusion

Preserved are the main findings, recommendations and implications for stakeholders. Quantified results obtained from representative deployments and configurations on two national transportation data platforms (DeLTA) support concrete conclusions.

Crucial transportation infrastructure provides core functionalities for the economy and society, and the ability to monitor, protect, and optimize such infrastructure through data is critical for a smart nation. Continuous advances in underlying data technologies, such as cloud computing and big data, underpin the trend of converging the highly heterogeneous data sets collected by transportation agencies into a national transportation data platform (NTDP) for analysis and smart application. The NTDP serves as a central repository for national transportation information, facilitates external departments to query and analyze national transportation information, and drives unified analysis and application.

Due to the NLDP's nationwide position and service demands, the data throughput of the access end is extremely large, coming from multiple sources such as cameras, probes, and radiosonde data, and is characterized by massive volume, multimode, high-dimensional, and high-data-rate characteristics. External query also requires a longer lifecycle for some storage, aggregating the real-time data from days, months, and even years, which requires a reasonable maintenance cost. Caching will also improve the query speed. Cost runtime and startup costs need to be carefully considered. Therefore, there are many optimizations still to be made.

References

1. Wang, H., Liu, Y., Zhang, Q., & Chen, X. (2025). Adaptive compression-aware storage optimization for nationwide transportation data lakes. *IEEE Access*, 13, 11245–11263.



2. Kumar, S., Reddy, P., & Sharma, V. (2025). Intelligent tiered storage frameworks for mobility big data infrastructures. *Journal of Big Data*, 12(1), 44–61.
3. Li, F., Zhao, Y., & Sun, J. (2025). Real-time compression pipelines for high-velocity transportation sensor streams. *Future Generation Computer Systems*, 158, 233–248.
4. Mangalampalli, B. M., Kolla, S. K., Bandi, V. D. V. K., Yandamuri, U. S., & Rani, P. S. (2025). Designing Intelligent Healthcare Ecosystems through Adaptive Data Integration and Autonomous Learning Systems. *Vascular and Endovascular Review*, 8(20s), 330–347.
5. Singh, A., Patel, K., & Rao, M. (2025). Scalable mobility data vault architectures using cloud-native storage services. *Information Systems Frontiers*, 27(2), 381–399.
6. Chen, Y., Wu, H., & Lin, Z. (2025). Data deduplication strategies for intelligent transportation analytics platforms. *ACM Transactions on Internet Technology*, 25(1), 1–22.
7. Gupta, N., Verma, S., & Agarwal, R. (2025). AI-assisted lifecycle management in transportation data lakes. *Journal of Cloud Computing*, 14(1), 77–95.
8. Kolla, S. H., & Mattaparthi, R. (2025). Hybrid Gen AI Systems: Integrating Small LMs with Large Language Models for Cost-Efficient Enterprise Automation and Decision Intelligence. *International Journal of Research Publications in Engineering, Technology and Management (IJRPETM)*, 8(6), 13345–13357.
9. Ibrahim, M., Noor, A., & Rahman, F. (2025). Hybrid storage orchestration for large-scale urban mobility platforms. *Sustainable Computing: Informatics and Systems*, 45, 101003.
10. Das, P., Roy, S., & Mukherjee, T. (2025). Performance optimization of smart mobility data warehouses through adaptive indexing. *Expert Systems with Applications*, 259, 124815.
11. Lee, J., Park, S., & Kim, H. (2025). Efficient compression algorithms for nationwide GPS telemetry archives. *IEEE Access*, 13, 22190–22208.
12. FinOps Strategies for AI-Enabled Real-Time Compliance Platforms in Cloud Native Environments. (2025). *MSW Management Journal*, 35(2), 2080–2088.



13. Alotaibi, Y., Alharbi, F., & Khan, M. (2025). Secure and compressed storage architectures for smart city mobility datasets. *Sensors*, 25(4), 1654.
14. Mehta, R., Joshi, D., & Kulkarni, P. (2025). Automated retention-aware storage tiering for transportation analytics clouds. *Journal of Network and Computer Applications*, 238, 104102.
15. Zhou, X., Li, W., & Chen, M. (2025). Streaming ingestion optimization in multimodal transportation data platforms. *IEEE Transactions on Intelligent Transportation Systems*, 26(5), 4892–4907.
16. Kolla, S. K., & Bandi, V. D. V. K. (2025). Autonomous Clinical Monitoring Platforms Using Reinforcement Learning and Deep Neural Networks. *International Journal of Advanced Research in Computer Science & Technology (IJARCST)*, 8(6), 13300-13313.
17. Park, D., Choi, Y., & Lee, H. (2025). Compression-enhanced caching mechanisms for mobility data lakes. *IEEE Access*, 13, 43102–43119.
18. Rao, K., Bhat, A., & Iyer, S. (2025). Data lineage and governance optimization for transportation cloud ecosystems. *Journal of Information Systems Engineering and Management*, 10(1), 55–73.
19. Martins, E., Silva, F., & Pereira, J. (2025). Intelligent storage orchestration for autonomous transportation platforms. *Future Generation Computer Systems*, 160, 319–337.
20. Mangalampalli, B. M., Bandi, V. D. V. K., Kolla, S. K., & Kumar, M. V. K. (2025). Towards Self-Evolving Healthcare Intelligence: Integrating Advanced Learning Systems with Real-Time Clinical Data Pipelines. *Cultura: International Journal of Philosophy of Culture and Axiology*, 22(12s), 464-486.
21. Sharma, R., Gupta, P., & Tiwari, A. (2025). High-throughput storage optimization for transportation IoT workloads. *IEEE Internet of Things Journal*, 12(7), 8411–8428.
22. Lin, H., Gao, X., & Zhou, Y. (2025). Scalable data lifecycle automation in smart transportation infrastructures. *Journal of Systems Architecture*, 152, 103015.



23. Ali, S., Hussain, M., & Qureshi, T. (2025). Advanced deduplication frameworks for cloud-native mobility data repositories. *ACM Transactions on Storage*, 21(2), 1–26.
24. Kolla, T. (2025). Generative AI for Intelligent Medical Coding and Healthcare Analytics. *International Journal of Advanced Research in Computer Science & Technology (IJARCST)*, 8(6), 13285-13299.
25. Karthik, V., Narayanan, S., & Menon, A. (2025). Adaptive storage provisioning for transportation big data environments. *Journal of Big Data*, 12(1), 93–111.
26. Sun, Y., Zhang, H., & Li, X. (2025). Compression-driven analytics acceleration for cloud mobility platforms. *IEEE Transactions on Cloud Computing*, 13(2), 905–921.
27. Ibrahim, A., Latif, R., & Khan, S. (2025). Efficient distributed storage mechanisms for mobility digital twins. *Sensors*, 25(6), 2448.
28. Mangalampalli, B. M., & Kolla, S. K. (2025). Large Language Models for Automated Healthcare Data Dictionary Generation and Maintenance. *Vascular and Endovascular Review*, 8(20s), 363-375.
29. Morales, P., Garcia, J., & Vega, M. (2025). Cloud object storage resilience for intelligent transportation infrastructures. *IEEE Access*, 13, 60114–60131.
30. Chatterjee, S., Dutta, A., & Banerjee, P. (2025). Mobility data warehouse convergence using lakehouse architectures. *Future Internet*, 17(3), 141–159.
31. Kim, T., Lee, S., & Jung, H. (2025). Dynamic partitioning strategies for nationwide mobility datasets. *Data & Knowledge Engineering*, 157, 102377.
32. Mangala, N., & Kolla, S. H. (2025). Real-Time Feature Engineering for Streaming AI Workloads Using PySpark and Azure Event Hubs. *International Journal of Multidisciplinary Research in Science, Engineering, Technology & Management*, 1(6), 93-105..
33. Ahmed, N., Rahman, M., & Islam, S. (2025). Secure data retention and archival mechanisms for transportation cloud platforms. *IEEE Transactions on Services Computing*, 18(3), 1784–1799.



34. Li, J., Xu, Y., & Cheng, L. (2025). AI-powered workload balancing in transportation storage clusters. *Expert Systems with Applications*, 262, 125140.
35. Singh, P., Yadav, R., & Chauhan, M. (2025). Energy-efficient storage optimization for transportation big data analytics. *Sustainable Computing: Informatics and Systems*, 46, 101065.
36. AGENTIC AI FRAMEWORKS FOR AUTONOMOUS RISK DETECTION AND COMPLIANCE REMEDIATION IN ENTERPRISE DATA CENTER OPERATIONS. (2025). *Lex Localis - Journal of Local Self-Government*, 23(S6), 9672-9697.
37. Hassan, E., Mahmood, A., & Tariq, U. (2025). Compression and indexing co-design for transportation data repositories. *IEEE Access*, 13, 72810–72829.
38. Wu, J., Li, P., & Zhang, R. (2025). Federated storage governance in national transportation platforms. *ACM Computing Surveys*, 58(2), 1–34.
39. Nair, S., Krishnan, R., & Joseph, A. (2025). Smart autoscaling strategies for transportation data processing clusters. *Journal of Systems and Software*, 221, 112081.
40. Kolla, S. H., & Peddi, R. K. (2024). Designing Governance-Aligned GenAI Pipelines Using Small Language Models for Enterprise Workflow Intelligence. *International Journal of Science, Research and Technology*, 7(6), 13256-13268.
41. Verma, A., Sinha, D., & Khandelwal, P. (2025). Distributed mobility data lake optimization using hybrid cloud architectures. *IEEE Transactions on Network and Service Management*, 22(1), 456–472.
42. Chen, R., Wang, T., & Liu, S. (2025). Adaptive query acceleration for transportation lakehouse systems. *Future Generation Computer Systems*, 162, 421–438.
43. Khan, F., Malik, A., & Saeed, H. (2025). Compression-enabled scalable storage for smart mobility services. *IEEE Access*, 13, 81255–81273.
44. Reddy, V. A. R. (2025). *Journal of Rare Cardiovascular Diseases*. *Health*, 5(3), 402-422.



45. Prasad, M., Iqbal, S., & Jain, K. (2025). Secure storage architectures for national transportation intelligence networks. *Computers & Security*, 145, 103956.
46. Yang, H., Zhou, W., & Tang, X. (2025). Storage cost minimization for real-time transportation analytics. *Journal of Supercomputing*, 81(4), 5120–5141.
47. Kapoor, R., Bhattacharya, S., & Sen, A. (2025). Intelligent compression and caching frameworks for mobility cloud services. *IEEE Transactions on Cloud Computing*, 13(3), 1342–1358.
48. Kolla, S. H., & Mattaparthi, R. (2025). Hybrid Gen AI Systems: Integrating Small LMs with Large Language Models for Cost-Efficient Enterprise Automation and Decision Intelligence. *International Journal of Research Publications in Engineering, Technology and Management (IJPETM)*, 8(6), 13345-13357.
49. Choi, K., Lim, Y., & Han, J. (2025). Advanced cloud-native storage strategies for transportation digital infrastructure. *IEEE Access*, 13, 93120–93139.
50. Thomas, E., Wilson, J., & Clark, P. (2025). Benchmarking compression and deduplication techniques in mobility big data environments. *Journal of Big Data Analytics in Transportation*, 7(1), 15–34.