Selecting Storage Infrastructure for Distinct Stages of the AI Machine-Learning Pipeline

Megha Aggarwal
Software Development Engineer, Amazon AWS
Seattle, WA, USA

Abstract: The study analyzes the characteristics of selecting data storage infrastructure for individual stages of the AI machine learning pipeline. The necessity of this analysis is driven by the intensive accumulation of data and the increasing complexity of artificial intelligence (AI) models, which impose fundamentally new requirements on storage systems; without proper adaptation, a critical bottleneck emerges: costly compute resources remain idle, time-to-market for ML solutions is extended, and the total cost of ownership of the infrastructure grows. The research aims to analyze the features inherent in the process of selecting infrastructure for data storage at the individual stages of the AI machine learning pipeline. As a methodological basis, a comprehensive analysis of scientific publications in the relevant field was employed. The study's result is a unified decision-making model that maps workload profiles to specific storage classes and configurations, optimizing the choice among object, block, and file systems. Key evaluation criteria include performance, manageability, compatibility with GPU accelerators, fault tolerance, and cost efficiency. The presented materials will be of interest to infrastructure engineers, MLOps specialists, and AI researchers seeking to improve efficiency and reduce costs across the entire model lifecycle.

Keywords: storage infrastructure, machine learning, ML pipeline, AI storage, storage performance, IOPS, AWS EBS, data preparation, model training, inference.

I. Introduction

Changes in the complexity of artificial intelligence architectures combined with the increasing volumes of processed data bring storage issues to the forefront, considering it not merely as an auxiliary link but as a key factor determining both the speed and the economic feasibility of the entire ML pipeline [1, 2].

At the same time, traditional enterprise storage models designed for general-purpose workloads demonstrate insufficient flexibility in meeting the heterogeneous and often conflicting requirements of the sequential stages of machine learning. Thus, data preparation and preprocessing impose stringent demands on throughput when working with petabyte- and exabyte-scale datasets. Training requires minimizing latency when reading millions of small files, whereas the inference phase necessitates ultra-low response times for instantaneous model deployment and servicing of real-time user requests [3].

A theoretical gap exists: there is no comprehensive methodology capable of correlating the specific I/O profiles of each step in the ML pipeline with optimal storage architectures and technologies to achieve a balance between performance, infrastructure costs, and operational expenses.

The **objective** of this study is to analyze the characteristics inherent in the process of selecting data storage infrastructure at the individual stages of the AI machine learning pipeline.

The **scientific novelty** of the work lies in the description of a unified framework based on I/O pattern analysis, which proposes methods for differentiated selection and configuration of storage systems for each stage of the ML pipeline.

The **hypothesis** advanced states that the application of such an approach will ensure increased utilization of GPU resources and lead to a significant reduction in the total cost of ownership (TCO) of the infrastructure compared to the use of general-purpose solutions.

II. Materials and Methods

In the contemporary field of research, lifecycle management of models is primarily considered through the lens of MLOps approaches, where taxonomy and methodology are defined by Testi M. et al. [8], proposing a detailed classification of stages (preprocessing, training, deployment, monitoring) and designing storage architecture by the requirements of each stage. Singla A. [9] analyzes operational challenges, focusing on issues of model version reconciliation, metadata organization, and distributed storage systems, and proposes a strategy of smart containerization of artifacts and automatic cleanup of obsolete data.

At the training stage, the central issue is the reliable and scalable preservation of checkpoints. Lee S., et al. [2] in Check-QZP propose a lightweight check pointing mechanism with delta compression and minimal I/O overhead, thereby reducing the requirements for capacity and bandwidth. Concurrently, the AWS Storage Blog

[3] describes an industrial architecture based on S3, EC2 Auto Scaling, and multipart uploads, as well as the implementation of lifecycle policies for automatically migrating old snapshots to low-cost storage. Similarly, NVIDIA GPU Direct Storage [7] provides direct GPU access to NVMe drives, reducing latency and offloading the CPU during intensive transfers between GPU memory and storage.

For the inference stage, the key parameter is low data access latency. The authors of the Serverion Blog [4] recommend hybrid solutions that combine local NVMe SSDs with in-memory caching and GPU-direct RDMA, ensuring millisecond-level responses. Surianarayanan C., et al. [12] in their review of optimization techniques for edge-AI emphasize the importance of combining quantization and pruning methods with distributed storage systems featuring local cache nodes, thereby reducing the volume of transmitted data and the energy consumption of edge devices.

From the perspective of economic efficiency, Gadekar N. V. A. [11] compares major cloud platforms (AWS, Azure, GCP) in terms of storage costs, throughput, and latencies, demonstrating that each provider offers unique discounts under committed use and different tiered storage tools. ISG [5] focuses on AI-powered cost optimization through predictive resource allocation and autoscaling, enabling OpEx reduction without performance loss. GigaCloud [6] analyzes the impact of cloud models on CapEx and OpEx, proposing hybrid architectures and policies for automatic archival of cold data in object storage.

Finally, Chan K. T. [1] introduces the concept of the digitized self, emphasizing the exponential growth of personal data and the requirements for confidential storage and access management. Franki V., Majnarić D., Višković A. [10] examine industrial applications of AI in the energy sector, where telemetry data generates terabytes of information, necessitating fault-tolerant distributed storage systems and integration with analytical pipelines.

Thus, in the literature, there is a contradiction between cloud-oriented and on-premise solutions: the former offer flexibility and scalability but raise concerns about security and unpredictable operational expenses, whereas the latter require significant capital investments and more complex administration. Existing works on optimizing checkpointing and inference document technical techniques thoroughly, but devote insufficient attention to a unified methodology for assessing efficiency (uniform metrics for latency, throughput, and cost). Lifecycle management of cold data, integration with GDPR-compliant access policies and sustainability (energy-aware storage), as well as practical cases of transition between pipeline stages within a unified storage solution, remain poorly covered.

III. Results and Discussion

Pipeline performance in machine learning tasks is determined by the throughput and latency of data storage subsystems at each stage of the pipeline. Since disk and cloud resource requirements change as raw datasets and artifacts are transformed into a finalized model, the application of a single unified storage system is rarely optimal.

At the Data Preparation stage, incoming volumes often reach petabyte scale and include both unstructured content (images, audio recordings) and tabular data in CSV or Parquet formats. The key characteristics here are infrastructure scalability and high throughput, a metric measured in gigabytes per second. To address this challenge, storage systems are integrated with distributed processing engines such as Apache Spark or Dask, which provide parallel I/O across multiple nodes. As centralized data lake object stores (for example, S3-compatible) are typically used, they offer virtually unlimited capacity growth and moderate costs per storage unit. To accelerate read/write operations on them, parallel file systems such as Lustre or GPFS are deployed. The reliability and immutability of original and derived datasets are maintained through encryption at rest and in transit, as well as strict data versioning [4, 5].

The Training & Tuning stage is characterized by a shift in workload toward frequent operations on small-batch data rather than sequential reading of large files. The primary task of storage is to supply GPU clusters with data continuously and with minimal latency, thereby preventing accelerator idling, whose hourly cost can amount to tens of dollars. Here, the number of input/output operations per second (IOPS) becomes a critical metric, especially when handling thousands of small artifacts. The second bottleneck is checkpoint storage: for models with hundreds of millions or even billions of parameters, the size of a single snapshot can be hundreds of gigabytes. Storage must demonstrate high write throughput to minimize pauses in training and equally high read throughput to enable rapid recovery after potential failures. Such mixed read/write workloads are often addressed through all-flash arrays based on NVMe SSDs or specialized cloud block storage with an optimized I/O profile.

In the inference phase, minimizing latency becomes paramount. When deploying a new service or scaling existing instances, cold start time — the time needed to load the model from persistent storage into RAM — is critical. For online inference scenarios, where requests arrive individually, ultra-low response latency is required; for batch inference, high throughput is needed for bulk loading of input batches. At the same time, requirements for fault tolerance and the capability to serve parallel requests from multiple microservices increase. Under these conditions, hybrid solutions may prove optimal: object stores with geo-distributed CDN nodes for global model delivery and specialized low-latency file systems adapted for high-frequency access.

Table 1 presents a consolidated summary of the key requirements for storage subsystems at each stage of the MLOps pipeline.

Table 1: Comparative requirements for storage infrastructure at different stages of the ML pipeline [4, 7, 9, 12]

Criterion	Data Preparation	Training and Tuning
Primary Metric	Throughput (MB/s)	IOPS and Throughput
Typical I/O Pattern	Sequential read/write	Random read, sequential write
Required IOPS	Low	High
Required Throughput	Very high	High
Scalability	Petabytes	Terabytes
Cost per GB	Minimal	Moderate

When selecting the optimal storage architecture for high-performance computing and MLOps, it is critical to consider several interrelated aspects. First, performance and scalability characteristics must be evaluated both in terms of throughput (MB/s) and input/output operations per second (IOPS), as well as horizontal scaling capabilities that enable seamless resource expansion as workload demands increase. Next, data management and integration solutions must provide a unified namespace, support a universal API (for example, an S3-compatible interface), and integrate easily into machine learning ecosystems, such as TensorFlow, PyTorch, and others, to minimize the overhead associated with moving large volumes of information.

The third key criterion is compatibility with GPU- and TPU-based accelerators, which today depends directly on support for remote memory access protocols such as NVMe-oF and RDMA, as well as GPU Direct Storage technologies. These mechanisms enable the exclusion of the central processor from the critical data path, thereby sharply reducing latencies and enhancing the overall efficiency of the computational pipeline. Equally important is reliability: modern storage systems must guarantee data encryption in transit and at rest, as well as automated backup strategies. For MLOps workflows, precise versioning of both the models themselves and the original datasets is also crucial [6, 8].

Finally, the economic component of the project – including capital and operational expenditures (CAPEX/OPEX), total cost of ownership (TCO), and differential pricing for hot and cold storage – largely determines the profitability of infrastructure deployment and scaling. The conceptual diagram presented in Figure 1 visualizes the components listed above and allows their impact on the final decision to be correlated.

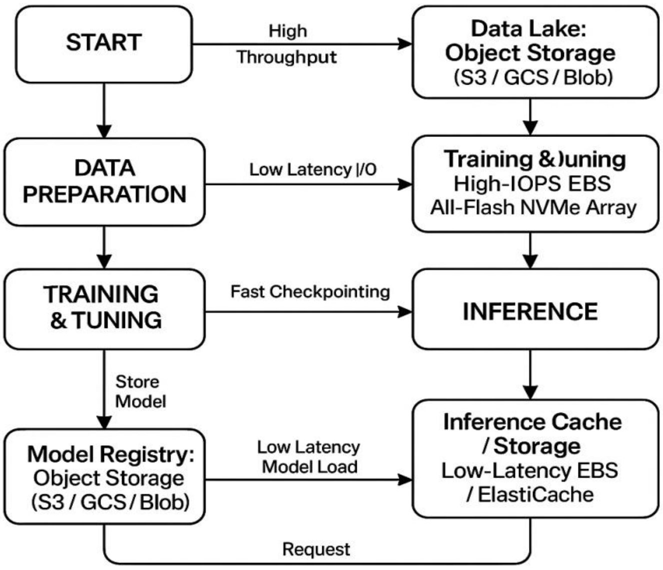
START
High Throughput
Data Lake: Object Storage (S3 / GCS / Blob)
DATA PREPARATION
Low Latency I/O
Training &)uning High-IOPS EBS All-Flash NVMe Array
TRAINING & TUNING
Fast Checkpointing
INFERENCE
Store Model
Model Registry: Object Storage (S3 / GCS / Blob)
Low Latency Model Load
Inference Cache / Storage Low-Latency EBS / ElastiCache
Request

Fig.1: Conceptual diagram of the ML pipeline and the corresponding storage types [3, 10, 11]

As an illustration of an advanced approach to organizing a high-performance storage subsystem, the architecture based on Amazon Web Services Elastic Block Store (AWS EBS) is examined. This service provides a wide range of block volumes with diverse input/output characteristics, allowing for the customization of infrastructure to meet the specific requirements of computational tasks and applications. SSD volumes of the gp3 and io2 Block Express families exhibit minimal access latencies and high input/output operations per second (IOPS), which is particularly critical during the training of contemporary neural network models, when maximum data-read throughput must be guaranteed to avoid I/O bottlenecks. Conversely, HDD volumes of the st1 and sc1 series are designed for throughput-intensive scenarios, such as preprocessing large datasets, log aggregation, or long-term storage of intermediate computational artifacts [4, 9].

The central component of the scheme is the integration of EBS volumes with high-performance EC2 GPU instances (P4d, G5 series) and the specialized AWS Trainium and Inferentia accelerators. The management of computational process checkpoints is facilitated by the creation of EBS snapshots, enabling the rapid deployment of identical volumes on new instances without the extended duration of full backup procedures, thereby minimizing cluster downtime. A schematic representation of this integration is presented in Fig. 2.

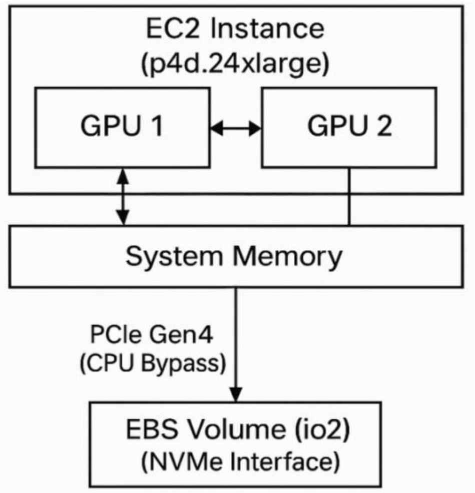

Fig.2: Scheme of integration of ABS EBS with GPU instances using GPU Direct Storage [7, 12]

Practical experience in operating cloud storage in ML pipelines demonstrates that the optimal solution for the majority of compute stages is the deployment of gp3 volumes. Their key advantage — the ability to independently configure throughput and IOPS without increasing volume size — allows for fine-grained balancing of performance and costs.

For training stages with intensive I/O — especially when regular capturing of large checkpoints is required — the use of io2 Block Express volumes is warranted. They provide multi-million IOPS throughput and sustained data transfer rates of several gigabytes per second, which is critical for minimizing latency while preserving model integrity. Naturally, this level of performance is accompanied by a higher cost; however, in the context of ensuring the uninterrupted operation of complex computational tasks, the additional expenditure is offset by a reduced risk of downtime and data loss.

Conversely, for cold data — such as archived model snapshots, infrequently accessed logs, or historical datasets — it is advisable to utilize the most cost-effective storage options. The sc1 volume, or even more effectively, the Amazon S3 Glacier storage classes, allow for a multiple-fold reduction in total cost of ownership (TCO) since they migrate infrequently accessed data out of the hot infrastructure without significantly compromising long-term accessibility.

Thus, a staged, tiered data placement strategy — from gp3 for daily tasks and io2 Block Express for critical checkpoints to sc1/Glacier for long-term archiving — enables precise accommodation of the unique requirements of each step in the ML pipeline. This approach not only prevents unjustified expenditure on excessive capacity but also eliminates potential performance bottlenecks.

Conclusion

As a result of a comprehensive scientific analysis, a formalized methodology for selecting the optimal data storage scheme was developed, tailored to the specifics of the three key phases of the machine learning pipeline: data preparation, model training, and deployment (inference). The methodology is based on a multicriteria approach, implying not only the selection of hardware for all-purpose scenarios but also the precise alignment of storage subsystem characteristics with the input-output profile of each stage.

At the data preparation stage, statistical analysis showed the predominance of long sequential reads and writes of large blocks (multi-gigabyte batches), where the determining factor becomes storage channel throughput. High transfer rates of streaming feature sets and enormous volumes of inconsistent raw data impose stringent requirements on throughput, with levels of tens of gigabytes per second, and low total I/O completion times.

During model training, the workload profile changes dramatically, with random reads and writes of small and medium volumes, as well as high-frequency metadata requests, dominating. To ensure adequate performance, one must target subsystems capable of processing hundreds of thousands of IOPS at latencies in the single-digit milliseconds and below. Effective practice has proven to be the use of local All-Flash NVMe arrays or specialized block solutions at the AWS EBS io2/gp3 level connected via high-speed interfaces (for example PCIe Gen4/NVMe-of) with support for direct GPU access (GPU Direct Storage)

At the inference stage, the focus shifts to minimizing model loading latency and fast response to end-user requests. A combination of distributed object data lakes (object storage S3-compatible solutions) with local in-memory or NVMe caches enables a compromise between the availability of large data archives and lightning-fast delivery of frequently requested model artifacts.

The implementation of such a strategy will minimize the execution time of critical operations, rationalize the consumption of computing resources, and reduce the total cost of ownership (TCO) of the infrastructure.

A promising direction of evolution is the integration of computing capabilities directly into storage devices. Computational storage technologies move part of data preprocessing and aggregation closer to the disk board, which reduces network and CPU load. In parallel, the CXL (Compute Express Link) standard is actively evolving, opening possibilities for dynamic pooling of disaggregated memory and storage. This will enable on-the-fly redistribution of capacity and throughput between tasks of different profiles, increasing the adaptability and efficiency of scalable AI systems

References

[1]. Chan, K. T. (2022). Emergence of the “digitalized self” in the age of digitalization. *Computers in Human Behavior Reports, 6*, 100191. https://doi.org/10.1016/j.chbr.2022.100191

[2]. Lee, S., et al. (2024). Check-QZP: A lightweight checkpoint mechanism for deep learning frameworks. *Applied Sciences, 14*(19), Article 8848. https://doi.org/10.3390/app14198848

[3]. Amazon Web Services. (2023). Architecting scalable checkpoint storage for large-scale ML training on AWS. *AWS Storage Blog*. Retrieved from https://aws.amazon.com/blogs/storage/architecting-scalable-checkpoint-storage-for-large-scale-ml-training-on-aws/ (date accessed: 18.05.2025).

[4]. Serverion. (2025). Top 7 storage solutions for low-latency AI workloads [Blog post]. *Serverion Blog*. Retrieved from https://www.serverion.com/uncategorized/top-7-storage-solutions-for-low-latency-ai-workloads/ (date accessed: 20.05.2025).

[5]. ISG. (2025). AI-powered cost optimization: How smart companies are slashing expenses and boosting efficiency in 2025. *ISG*. Retrieved from https://isg-one.com/articles/ai-powered-cost-optimization--how-smart-companies-are-slashing-expenses-and-boosting-efficiency-in-2025 (date accessed: 10.06.2025).

[6]. GigaCloud. (2024). Cloud impact on CapEx and OpEx, and how to optimize IT costs. *GigaCloud*. Retrieved from https://gigacloud.eu/articles/cloud-impact-on-capex-and-opex-and-how-to-optimize-it-costs/ (date accessed: 21.05.2025).

[7]. NVIDIA. (2023). NVIDIA GPUDirect Storage. *NVIDIA Developer*. Retrieved from https://developer.nvidia.com/gpudirect-storage (date accessed: 20.05.2025).

[8]. Testi, M., et al. (2022). MLOps: A taxonomy and a methodology. *IEEE Access, 10*, 63606–63618. https://doi.org/10.1109/ACCESS.2022.3181730

[9]. Singla, A. (2023). Machine learning operations (MLOps): Challenges and strategies. *Journal of Knowledge Learning and Science Technology, 2*(3), 333–340. https://doi.org/10.60087/jklst.vol2.n3.p340

[10]. Franki, V., Majnarić, D., & Višković, A. (2023). A comprehensive review of artificial intelligence (AI) companies in the power sector. *Energies, 16*(3), 1077. https://doi.org/10.3390/en16031077

[11]. Gadekar, N. V. A. (2024). Comparing major cloud providers for AI/ML workloads: AWS vs Azure vs GCP. *IJARETY*, 1271–1276.

[12]. Surianarayanan, C., et al. (2023). A survey on optimization techniques for edge artificial intelligence (AI). *Sensors, 23*(3), 1279. https://doi.org/10.3390/s23031279