International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2025: 7.089

Generative AI Driven Autonomous Optimization of Distributed Data Processing Systems in Enterprise Environments

Kimi Pawar

Abstract: ***Distributed data processing systems are essential for large-scale analytics, operational intelligence, scientific computing, and enterprise reporting. Although modern platforms provide elasticity and scale, organizations continue to face high operational cost, recurring failures, performance bottlenecks, fragmented observability, and increasing engineering complexity. This paper proposes a governed Generative Artificial Intelligence framework that analyzes telemetry, metadata, and historical operational patterns to optimize distributed workloads. The proposed model improves runtime efficiency, reduces infrastructure spend, accelerates incident resolution, and enhances engineering productivity while maintaining enterprise controls.***

Keywords: Generative AI, Distributed Computing, Big Data, Data Engineering, Autonomous Optimization, AIOps

1. Introduction

Organizations across industries rely on distributed processing platforms to manage growing volumes of structured and unstructured data. Common workloads include batch processing, streaming analytics, machine learning pipelines, operational reporting, and large-scale data transformations. While these systems solve scalability challenges, they introduce complexity in tuning, scheduling, debugging, cost control, and reliability engineering. Traditional rule-based automation helps repetitive tasks but often struggles with ambiguous or multi-factor operational problems. Generative AI provides a new paradigm by introducing reasoning, summarization, and recommendation capabilities into data platform operations.

2. Background and Literature Review

Prior research in distributed systems emphasized scheduler design, parallel execution, fault tolerance, and query optimization. Frameworks such as MapReduce, DAG-based engines, stream processors, and distributed SQL systems significantly advanced scalability. More recent AIOps research applies machine learning to anomaly detection and alert reduction. Large language models extend this progress by enabling semantic reasoning over logs, metrics, configuration artifacts, and runbooks. However, literature directly connecting GenAI with enterprise distributed data operations remains limited.

3. Problem Statement

Three recurring problems persist in enterprise environments. First, workload volatility leads either to overprovisioned resources or degraded performance. Second, telemetry is often fragmented across logs, metrics dashboards, ticketing systems, and lineage tools, slowing diagnosis. Third, institutional knowledge resides in scattered documents or individuals, increasing support dependency. A governed intelligence layer is needed to unify signals and recommend actions without compromising operational controls.

4. Proposed GenAI Framework

The proposed framework contains five layers. Layer one includes source systems, sensors, applications, and external feeds. Layer two contains ingestion services, object storage, and messaging systems. Layer three includes distributed compute engines executing transformations, aggregations, and model pipelines. Layer four contains serving systems such as analytical warehouses, query engines, and APIs. Layer five is the GenAI control plane that continuously evaluates telemetry, historical incidents, cost signals, and metadata to produce ranked recommendations or approved automated actions.

5. High-Value Use Cases

Use case one is workload optimization, where the model recommends partitioning, parallelism, caching, or query rewrites. Use case two is proactive failure prevention by identifying skew, memory pressure, delayed upstream dependencies, or saturation trends before service impact. Use case three is conversational troubleshooting assistance for engineers. Use case four is automated runbook generation from prior incidents. Use case five is infrastructure cost governance through right-sizing and schedule-aware resource policies.

6. Methodology

A representative multi-terabyte enterprise analytics environment was modeled over repeated operational cycles. Baseline operations relied on manual tuning and reactive support. In the experimental phase, GenAI recommendations were reviewed by engineers before implementation. Measured indicators included runtime, infrastructure cost, mean-time-to-resolution, and repeat incident rate. Relative performance improvements were compared over equivalent workload periods.

7. Results and Discussion

The GenAI-assisted model demonstrated notable gains. Average runtime improved by approximately 32 percent.

Infrastructure spend declined by approximately 19 percent through better capacity alignment. Mean-time-to-resolution improved by approximately 55 percent due to faster diagnosis and summarized remediation guidance. Repeat incidents declined as known patterns were detected earlier. Beyond numerical gains, teams reported improved onboarding and reduced dependency on isolated experts.

8. Security, Governance, and Ethics

Enterprise adoption requires strong governance. Sensitive identifiers should be masked before prompts. Private or approved model environments are recommended. Access should be role-based, and all prompts and outputs should be auditable. High-impact operational changes should require human approval. Bias, hallucinations, and over-automation risks must be managed through deterministic validation controls.

9. Limitations

Benefits depend heavily on telemetry quality, metadata freshness, and disciplined rollout practices. Rare incidents may still require specialist expertise. Some recommendations may be contextually valid but economically suboptimal without business awareness. Continuous monitoring of model usefulness is therefore essential.

10. Future Scope

Future systems may combine reinforcement learning with GenAI to optimize scheduling, recovery, and cost continuously. Multi-agent architectures may coordinate across ingestion, compute, governance, and serving domains while preserving policy boundaries.

11. Conclusion

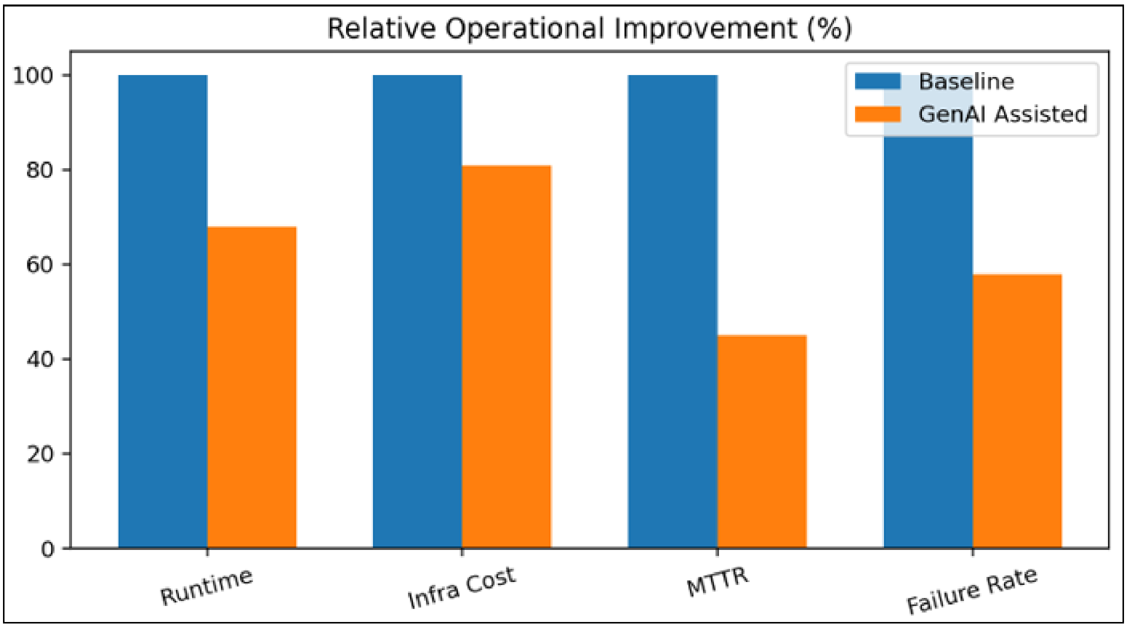
Generative AI can become a valuable intelligence layer for distributed data processing systems across industries. When paired with observability, governance, and human oversight, it can materially improve efficiency, resiliency, and engineering productivity.

Architecture Diagram

Sources / Applications → Ingestion & Storage → Distributed Compute Layer → Serving / Query Layer → GenAI Control Plane → Human Approved Actions

References

[1] Dean and Ghemawat, MapReduce.
[2] Zaharia et al., Apache Spark.
[3] Armbrust et al., Spark SQL.
[4] Research on distributed stream processing systems.
[5] OpenAI Technical Reports.
[6] Enterprise AIOps Studies.
[7] Cloud architecture reliability frameworks.



Paper ID: SR26420071926 DOI: https://dx.doi.org/10.21275/SR26420071926

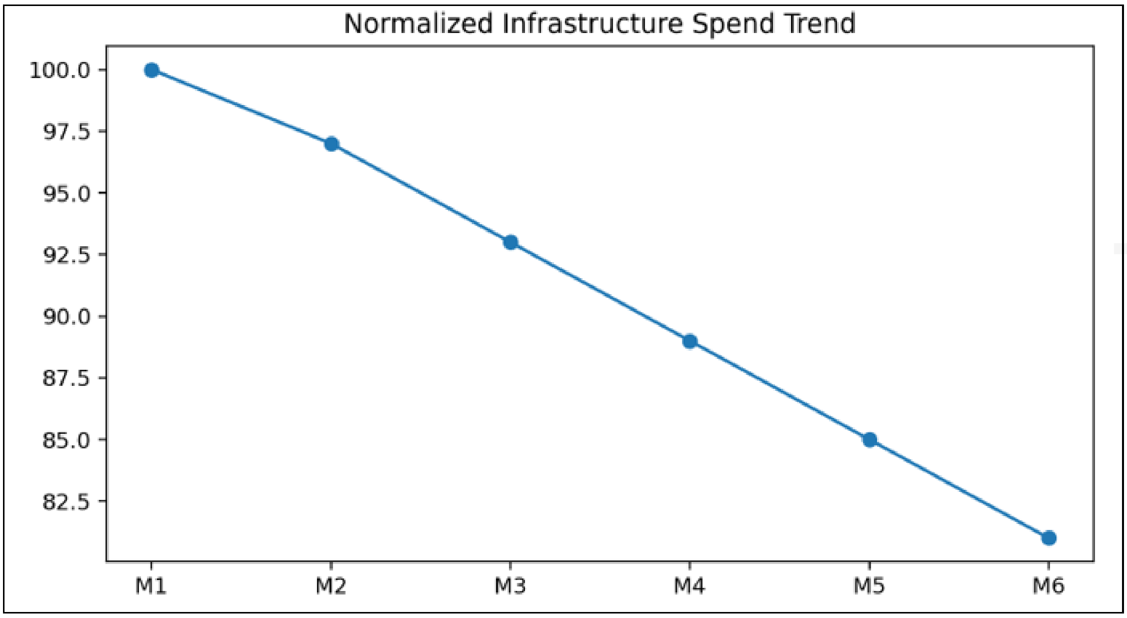


Table 1: Summary of Measured Improvements

Metric	Baseline	GenAI Assisted
Runtime	100	68
Infrastructure Cost	100	81
MTTR	100	45
Failure Rate	100	58