

Project Title: Haystack - Dynamic Replication

TEAM_ID_45

Vilal Ali_2024701027 and Shriom Tyagi_2023900034

1. Statement of the Problem

In distributed storage systems, particularly those designed to manage large datasets and high-throughput demands, traditional static replication strategies often fall short. These static methods lack adaptability to changing network conditions and varying user demands, leading to resource inefficiencies, latency spikes, and inconsistent data access times. As data access patterns evolve, so too should the replication strategy, allowing frequently accessed data to be available close to the end-user or critical application, while minimizing unnecessary storage and processing costs.

"Haystack - Dynamic Replication" addresses this issue by proposing a flexible, real-time approach to replication in distributed storage. Through dynamic replication, the system can adjust the number and location of data replicas based on real-time conditions such as server load, access frequency, and network latency. This adaptability is aimed at achieving an optimized balance between performance and resource usage, leading to a system capable of better handling load distribution and reducing latency in data access.

2. Important Papers and Materials

To inform the project design and implementation, we will review several key publications and resources that outline the challenges and advancements in distributed storage, dynamic replication, and data availability. These materials include:

- **"Haystack: Efficient Object Storage for Facebook's Photos"** — This paper provides a foundational understanding of the Haystack architecture, detailing its use case, storage efficiency, and performance considerations. It serves as the baseline on which our project's dynamic replication model will build.
- **"Dynamic Replication in Distributed Systems: A Survey"** — This paper surveys various dynamic replication strategies and approaches, allowing us to evaluate different models and their applicability to the Haystack system.
- Technical documentation and resources on gRPC, distributed systems load balancing, and network monitoring for real-time performance adjustments.

This literature will guide our design decisions and help us understand existing limitations while inspiring innovation in dynamic replication for distributed storage.

3. Scope of the Project

Our project will focus on adding a dynamic replication feature to the Haystack storage system, enabling the system to make adaptive decisions on data replication. This feature will evaluate data access patterns and respond to load fluctuations by redistributing or duplicating data in real time.

Feature Overview: The dynamic replication feature will introduce a replication policy engine, which will monitor storage nodes and adjust the replication policy based on specific factors:

- **Access Patterns:** Data accessed frequently will have additional replicas on high-performance nodes, while infrequently accessed data will maintain fewer replicas to conserve resources.
- **System Load:** During high-traffic periods, the replication engine will duplicate popular data across multiple nodes, ensuring access reliability.

- **Latency Requirements:** Data will be moved closer to users or services that demand low-latency access, thus optimizing retrieval times and user experience.

Use Cases: This feature targets high-volume data environments where storage demands and access frequency are volatile, such as media hosting services, content delivery networks, and cloud storage for social media applications.

4. Solution Approach

To implement this feature, we plan the following approach:

a. Main Components:

- **Replication Policy Engine:** This component will use real-time analytics to decide when and where replicas should be created, adjusted, or removed. The policy engine will factor in metrics like data access frequency, latency, and server load.
- **Data Placement Service:** A service to handle the physical replication and placement of data across storage nodes, orchestrating data transfer while minimizing disruption.
- **Monitoring and Metrics System:** A real-time monitoring system that captures performance metrics and provides the policy engine with the necessary data for decision-making.

Technologies and Tools: Our implementation will utilize gRPC for inter-service communication, as it allows for efficient and scalable interaction between storage nodes and management components. For monitoring, we will integrate an analytics and metrics tool, likely Prometheus, to capture load metrics and server response times. Testing will involve simulating varying access loads and network conditions to evaluate the adaptability of the dynamic replication feature.

Dataset: We will either simulate a dataset of large objects or utilize an open-source media dataset to test the effectiveness of dynamic replication on realistic, large-scale data. This setup allows us to measure the performance improvements and resource efficiency in real-world-like conditions.

5. Project Timeline and Milestones

The project timeline is structured around key deliverables and testing phases:

- **Weeks 1:** Complete literature review and finalize system design and architecture. This will involve defining the metrics for replication triggers and selecting the initial replication policy.
- **Weeks 2-3:** Develop and implement the Replication Policy Engine and Data Placement Service, focusing on integrating these components with Haystack's existing structure.
- **Weeks 4:** Testing and iteration. We will conduct tests under various simulated load conditions, analyze system behavior, and make adjustments to the policy engine as needed.
- **Weeks 5:** Final testing, report compilation, and presentation. Final adjustments will be made to the system, and we will document our findings and outcomes in preparation for the final submission.