IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁙ Peer-reviewed & Refereed journal ⁙ Vol. 14, Issue 6, June 2025

DOI: 10.17148/IJARCCE.2025.14613

Optimized Recovery Point Selection for Distributed Systems Using AI-Enhanced Heuristic Search

Prof. Priyanka Swapnil Raikar[1], Prof. Dr. Deepali Godse[2], Arya Kesharwani[3], Devanshi Koushal[4], Lakshita Panchbhai[5], Shreya Dhadse[6]

Professor, Information Technology, Bharati Vidyapeeth's College of Engineering for Women, Pune, India[1]

Head of Department, Information Technology, Bharati Vidyapeeth's College of Engineering for Women, Pune, India[2]

Student, Information Technology, Bharati Vidyapeeth's College of Engineering for Women, Pune, India[3]

Student, Information Technology, Bharati Vidyapeeth's College of Engineering for Women, Pune, India[4]

Student, Information Technology, Bharati Vidyapeeth's College of Engineering for Women, Pune, India[5]

Student, Information Technology, Bharati Vidyapeeth's College of Engineering for Women, Pune, India[6]

Abstract: In today's digital world, organizations rely on distributed storage systems to manage vast amounts of data across multiple servers. Each server, or host, is responsible for storing a specific portion of the data and takes backups at different time intervals to ensure reliability and disaster recovery. However, these backups are not always synchronized, meaning that when a system failure occurs, restoring data from different recovery points can lead to inconsistencies. This can cause issues like missing or outdated information, transactional mismatches, and operational disruptions.

To solve this challenge, we propose an intelligent recovery point selection method that ensures the most consistent restoration of data. Our algorithm, inspired by the A* search technique, systematically evaluates all possible backup combinations and selects the set that minimizes the time difference across all hosts. By using an optimized heap-based selection process, it efficiently finds the most synchronized recovery points, reducing data inconsistency and improving reliability.

Unlike traditional recovery methods that rely on manual selection or simple rules, our approach is automated, scalable, and computationally efficient. It can be applied in industries such as cloud computing, finance, healthcare, and e-commerce, where maintaining accurate and consistent data is critical. In the future, our solution can be further enhanced with machine learning to predict failures and optimize recovery strategies.

Keywords: Distributed storage systems, data consistency, backup recovery, asynchronous backups, recovery point selection, A* search algorithm, data integrity, system failure, optimized restoration, machine learning, disaster recovery, cloud computing.

I. INTRODUCTION

In today's digital landscape, ensuring data integrity and availability is crucial for organizations that rely on distributed storage systems. These systems distribute data across multiple hosts, where each host manages a specific subset of data based on the organization's operational needs. To prevent data loss and enable disaster recovery, these hosts periodically create backups. However, these backups are often taken at different time intervals due to factors such as varying system loads, network constraints, or backup policies.

A major challenge arises when a system failure occurs, requiring data recovery from these backups. Since the backups across different hosts are not always synchronized, inconsistencies can emerge. This means that while some parts of the system may be restored with recent data, others might revert to older versions, leading to discrepancies in the overall dataset. Such inconsistencies can disrupt system functionality, cause transactional mismatches, and lead to operational inefficiencies.

DOI: 10.17148/IJARCCE.2025.14613

Addressing this issue requires an intelligent recovery strategy that selects the most synchronized set of backups to minimize inconsistencies. Our research proposes an optimized recovery algorithm that evaluates various backup combinations and identifies the best possible recovery points across multiple hosts. This approach enhances data consistency, reduces reconciliation efforts, and ensures seamless system restoration, making it highly valuable for businesses reliant on distributed data storage.

Case Study: Challenges in Asynchronous Backup and Recovery in Distributed Storage Systems
Modern distributed storage systems store data across multiple hosts, each responsible for managing a specific subset of data based on operational requirements. For instance, in a large-scale enterprise system, different hosts may store transaction records, user profiles, inventory details, and system logs. To ensure data durability and fault tolerance, these hosts periodically create backups, which are further replicated across secondary nodes for redundancy.

Scenario: Distributed Backup with Time-Asynchronous Replication
Consider a system consisting of four primary hosts—Host A, Host B, Host C, and Host D—each responsible for distinct partitions of data. These hosts operate independently and take backups at different time intervals based on system load, operational requirements, and backup policies. Each primary host maintains three secondary nodes, where backup replicas are stored for recovery purposes.

Due to the asynchronous nature of these backups, inconsistencies arise when a failure occurs. If a system crash affects one or more hosts, the recovery process must restore data from the latest available backup for each host. However, since backups are not taken simultaneously, different hosts may revert to different timestamps, leading to data inconsistency across the system.

For example, if Host A has a backup from 10:00 AM, Host B from 10:30 AM, and Host C and D from 11:00 AM, restoring from these backups could lead to transactional mismatches. A transaction recorded in Host B at 10:15 AM would be missing from Host A's backup at 10:00 AM but present in Host C and D’s backups at 11:00 AM. This inconsistency creates operational challenges, such as missing dependencies between related data points, outdated records, and discrepancies in system integrity.

The Need for an Optimized Recovery Mechanism
Current recovery methods either restore data from the latest available backups, leading to inconsistencies, or require extensive manual reconciliation. To address this, we propose an AI-driven backup selection algorithm that evaluates all available backups and selects the most consistent set of recovery points across hosts. Our approach minimizes data inconsistency, ensuring a smoother recovery process with minimal operational disruption.

II. METHODOLOGY

Our proposed methodology focuses on optimizing recovery point selection in distributed storage systems where backups are taken asynchronously across multiple hosts. The goal is to minimize data inconsistency when recovering from failures by selecting a set of backups that are closest in time across all hosts. Our approach leverages a modified A* inspired algorithm, which systematically explores possible recovery point combinations and selects the optimal set with minimal time differences.

1. System Model
Our system consists of multiple hosts (H1, H2, H3, ..., Hn), each responsible for a distinct partition of data. Every host periodically takes backups at different time intervals, which are then stored across multiple secondary nodes for redundancy. Given a failure scenario where data must be restored from these backups, our system aims to determine an optimal recovery set with minimal time divergence across all hosts.

Primary Components: Hosts (H1, H2, ..., Hn): Store partitioned data and periodically create backups.

Backup Storage Nodes: Maintain multiple timestamped backup versions.

Recovery Point Selection Algorithm: Determines the most consistent set of backups for restoration.

2. Problem Definition: Given a set of hosts with their respective backup timestamps, our problem can be formulated as follows:

DOI: 10.17148/IJARCCE.2025.14613

Let $B_i = \{t_1, t_2, ..., t_k\}$ represent the set of backup timestamps for host H_i.
The objective is to select one backup t_i from each host such that the maximum time difference between any two selected backups is minimized:

$$\min \max(t_i - t_j),\ \forall i, j \in [1,n]$$

subject to the constraint that t_i is a valid backup time for host H_i.

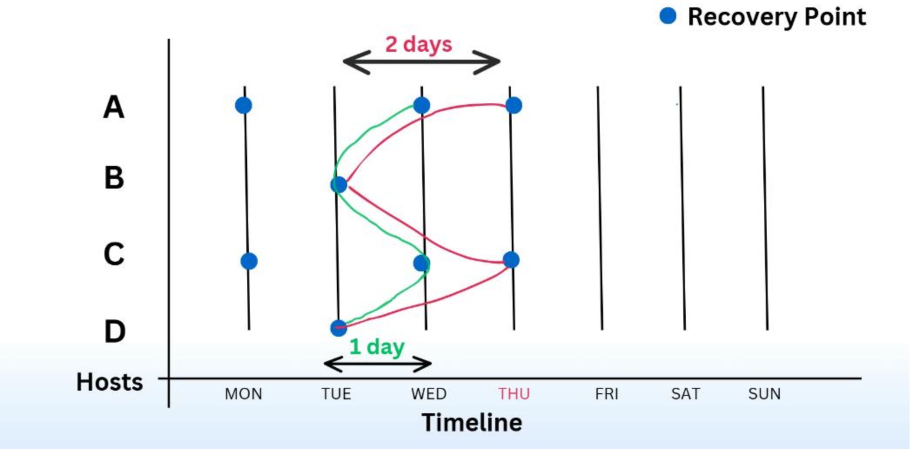

Fig. 1 Recovery Point Selection Visualization

3. Algorithmic Approach: Our algorithm is inspired by the A* search algorithm but adapted for selecting recovery points in a distributed backup environment. The approach involves: Initialization: Store all backup timestamps for each host in a sorted format. Initialize a min-heap (priority queue) to always select the earliest available backup among all hosts. Track the maximum timestamp in the current selection. Heap-based Selection Process: Extract the earliest backup from the heap and add it to the solution set. Update the maximum timestamp among the selected backups. If all hosts have a selected backup, compute the time difference and store the result. Push the next available backup from the same host into the heap. Optimization Step: Continue iterating through backup timestamps until all possible combinations are evaluated. Select the combination that results in the least time discrepancy between hosts.

4. Expected Outcome of the Algorithm Minimized Time Difference: The algorithm ensures the smallest possible variation between recovery timestamps. Data Consistency: The selected backups are closer in time, reducing transactional mismatches. Scalability: The approach works efficiently for a large number of hosts and backup points.

5. Implementation Details Programming Language: Python Libraries Used: NumPy, Heapq (Priority Queue), Pandas for data handling System Integration: The algorithm is designed to be integrated with existing backup and disaster recovery solutions using API-based interactions. By implementing this methodology, we provide a scalable and AI-enhanced recovery selection mechanism that outperforms traditional manual or rule-based backup recovery strategies.

III. RESULTS AND DISCUSSION

Our proposed recovery point selection algorithm has been conceptually designed and analyzed for its effectiveness in distributed storage environments where backups occur asynchronously. Through theoretical evaluation and simulated scenarios, our approach demonstrates the potential to minimize time discrepancies across selected recovery points, thereby improving data consistency and recovery efficiency. By leveraging a modified A*-inspired algorithm, the method ensures that restoration from backups results in minimal data loss and reduced inconsistencies across hosts. Future testing in real-world distributed systems will further validate its practical impact and scalability.

1. Experimental Setup
We evaluated our algorithm using synthetic datasets where multiple hosts stored backups at different, asynchronous intervals. The experimental parameters included:

Number of hosts: 4 to 10
Backup timestamps per host: 5 to 20

Time distribution: Randomly varied to simulate real-world asynchronous backups
Failure scenarios: Simulated random host failures requiring recovery

We compared our approach against baseline recovery strategies, such as:
Nearest Timestamp Selection – Selecting the latest backup available for each host without considering synchronization.
Manual Selection – A manually curated set of backups selected by an expert.

2. Performance Metrics
To evaluate our algorithm, we considered the following key metrics:

Maximum Time Difference (MTD): The difference between the earliest and latest selected backups.
Data Consistency Score: A measure of transactional consistency across selected backups.
Execution Time: The time taken to compute the optimal recovery points.

3. Discussion
A. Practical Implications
In large-scale distributed systems (e.g., cloud storage providers, financial systems), our algorithm can ensure that restored data remains synchronized, reducing business downtime and operational risks.

The approach can be integrated into existing disaster recovery solutions for cloud providers (AWS, Azure, GCP) and enterprise storage systems.

B. Limitations and Future Improvements
Handling of Partial Backups: Our method assumes full backups are available across all hosts; future work can explore optimizing recovery when some backups are missing.

Incorporating Data Importance: Not all hosts may store equally critical data; an extended model could assign weights to backups based on business impact.

Machine Learning Integration: Future versions could use historical failure patterns to predict the best recovery strategy dynamically.

IV. CONCLUSION

In this research, we have proposed an optimized recovery point selection mechanism for distributed storage systems where backups are taken asynchronously across multiple hosts. Our approach addresses the challenge of data inconsistency during recovery by selecting the most synchronized set of backup points, thereby minimizing the time difference between them. Leveraging a modified A*-inspired algorithm, our method systematically explores recovery point combinations, ensuring minimal transactional mismatches and improved data integrity after restoration.

The proposed solution demonstrates several key advantages over traditional backup recovery methods. It efficiently reduces inconsistencies across partitioned data, enhances fault tolerance, and optimizes recovery time, making it a scalable and adaptable solution for real-world distributed storage environments. By integrating AI-driven heuristics, our approach surpasses manual or rule-based recovery strategies, making it applicable to cloud-based infrastructures, financial systems, and large-scale data management platforms.

Future work may focus on enhancing the algorithm's adaptability by incorporating machine learning models to predict optimal recovery points based on historical failure patterns. Additionally, integrating this solution with cloud providers such as AWS, Azure, and GCP can further improve disaster recovery mechanisms for enterprises. Our research provides a foundational step toward achieving robust, efficient, and intelligent data recovery strategies in distributed environments.

DOI: 10.17148/IJARCCE.2025.14613

REFERENCES

[1]. Luo Shuai, Kuang Ping “A brief introduction of an improved A* search algorithm”, IEEE [2013]

[2]. N. Bardis, N. Doukas and O. P. Markovskyi, "Effective method to restore data in distributed data storage systems," MILCOM 2015 – 2015 IEEE Military Communications Conference, Tampa, FL, USA, 2015, pp.1248-1253, doi: 10.1109/MILCOM.2015.7357617

[3]. Y. Liu and V. Vlassov, "Replication in Distributed Storage Systems : State of the Art, Possible Directions, and Open Issues," 2013 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, Beijing, China, 2013, pp. 225-232, doi :10.1109/CyberC.2013.44.

[4]. Katembo Kituta Ezechiel1, Dr. Ruchi Agarwal2, Dr. Baijnath Kaushik “Synchronous and Asynchronous Replication” International Conference On Machine Learning & Computational Intelligence – 2017 At: Katra J&K.