Root Cause Analysis (RCA) Report

Name : Bhavana Shanivendram

Role : Data Analyst

Date : 06/16/2025

Summary

This report analyzes the underlying factors contributing to elevated system response times (**DELAY**) using the provided dataset.
A combination of **data exploration, correlation analysis**, and **Root Cause Analysis (RCA)** tools — including the **Fishbone (Ishikawa) Diagram** and **5 Whys** method — were utilized to uncover performance bottlenecks.

The analysis revealed that **application-level errors**, specifically **ERROR_1000**, exhibit the **strongest positive correlation** with DELAY and are therefore the primary contributors to latency issues.
The following sections present supporting charts, RCA diagrams, analytical insights, and recommendations for remediation and prevention.

---

1. Data Description

The dataset consists of **1,000 records** and **9 attributes**, structured as follows:
ID, CPU_LOAD, MEMORY_LEAK_LOAD, DELAY, ERROR_1000, ERROR_1001, ERROR_1002, ERROR_1003, and ROOT_CAUSE.

The numerical variables — particularly **DELAY**, **CPU_LOAD**, **MEMORY_LEAK_LOAD**, and **ERROR_1000–ERROR_1003** — were analyzed to determine which factors most significantly influence system delay.
A sample excerpt (first 10 rows) is available in the appendix for reference.

---

2. Exploratory Data Analysis (EDA) and Visual Insights

Exploratory analysis was conducted to evaluate relationships between DELAY and other performance indicators.
Pairwise correlations and error frequency counts were computed to identify patterns and potential causative factors.

Key Observations:

- **ERROR_1000** displayed the highest correlation with DELAY, suggesting a strong association with latency spikes.
- **ERROR_1001** and **ERROR_1002** occurred more frequently but showed weaker correlations.

- **CPU_LOAD** and **MEMORY_LEAK_LOAD** exhibited minimal correlation, implying a lesser but potential secondary impact on performance.

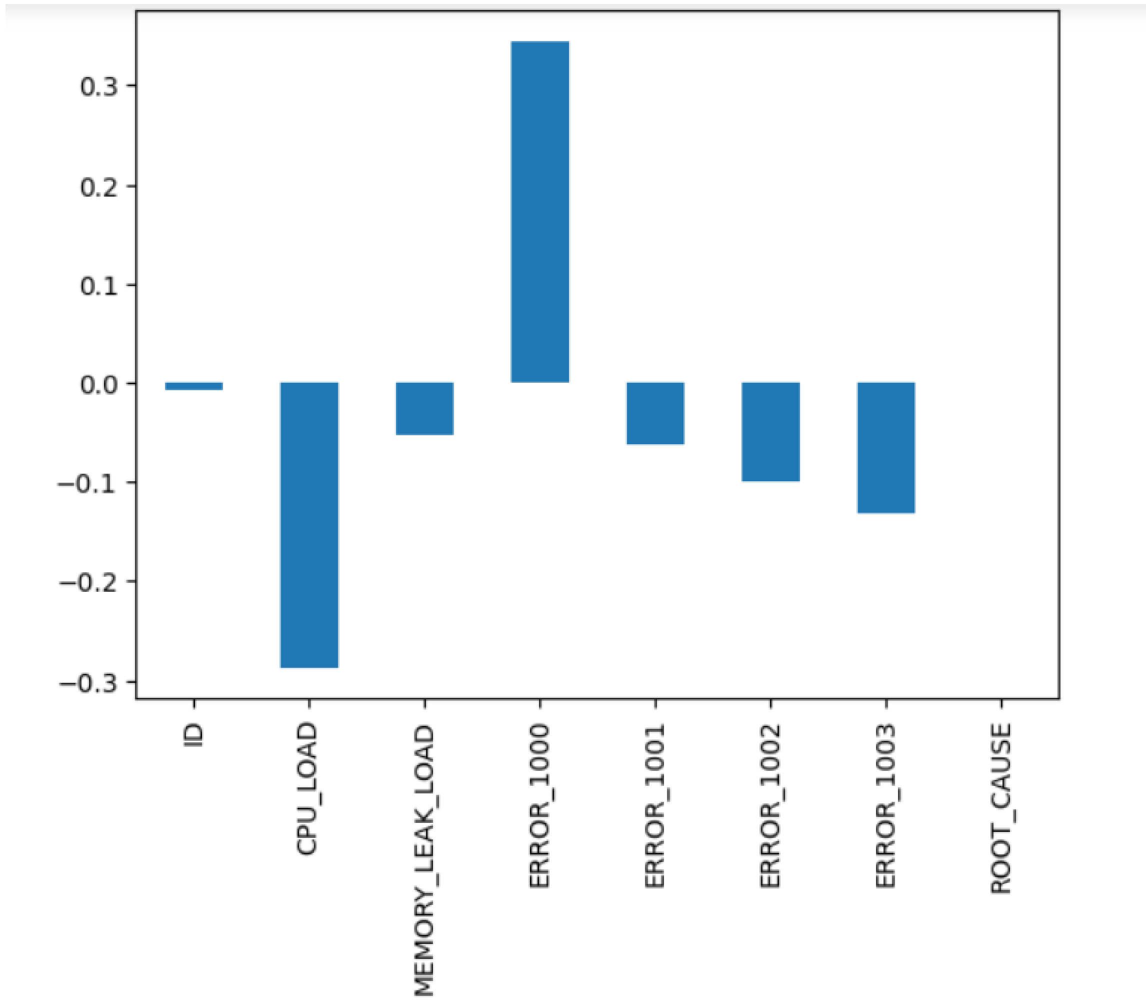

Figure 1: Correlation of Features with DELAY

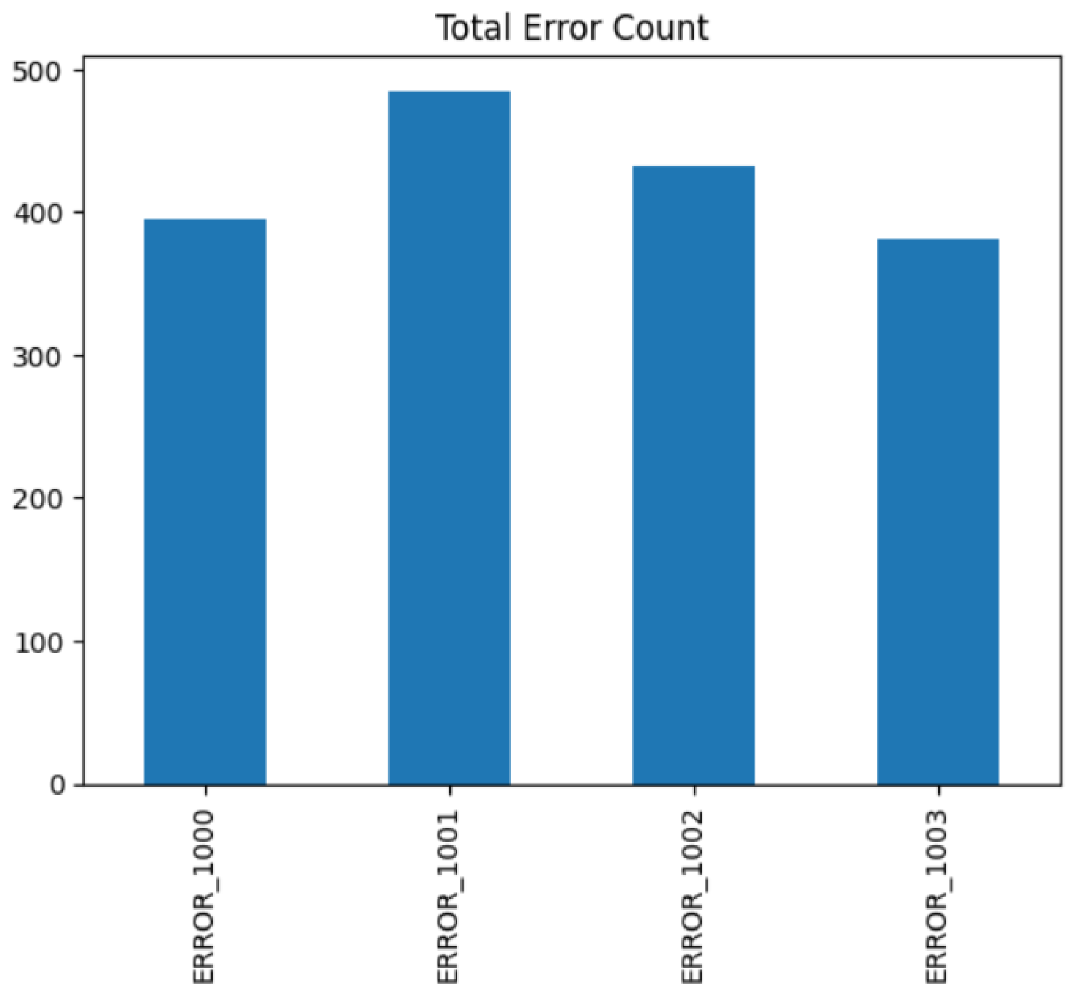

Figure 2: Total Occurrences by Error Type

---

3. Root Cause Analysis Techniques

To systematically identify the drivers of system delay, two RCA techniques were applied:

1. **Fishbone (Ishikawa) Diagram:**
   Categorizes possible causes under six dimensions — *Machine*, *Method*, *Material (Software)*, *Man (People)*, *Measurement*, and *Environment.*

2. **5 Whys Method:**
   A sequential questioning framework used to trace the issue from symptoms to root cause.

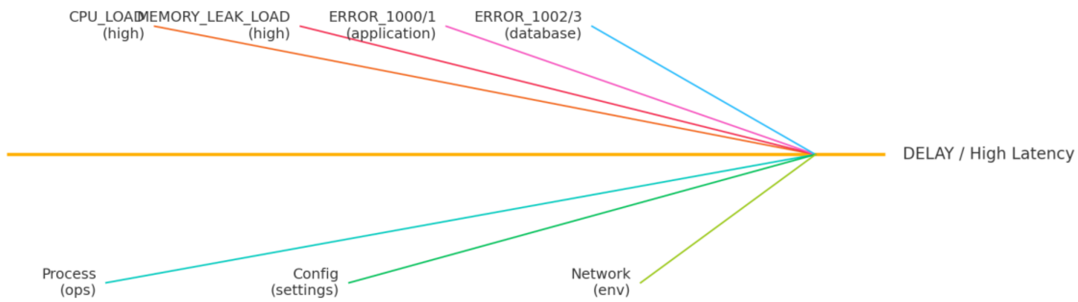

Figure 3: Fishbone Diagram (schematic)

---

4. 5 Whys Analysis

Problem:
Significant increase in system response time (DELAY).

Step	Question	Root Observation
1	Why did system delay increase?	Because requests failed due to ERROR_1000, leading to multiple retries and blocking.
2	Why did ERROR_1000 occur?	Component A made synchronous calls to Service B, which timed out.
3	Why did Service B time out?	Slow and unoptimized database queries delayed responses.
4	Why were queries inefficient?	Missing indexes and poor query optimization.
5	Why was this not detected earlier?	CI/CD pipeline lacked performance and integration testing for this scenario.

Root Cause Identified:

System delays were primarily caused by **application-level errors (ERROR_1000)** resulting from **unoptimized database operations** and **inadequate test coverage** during deployment.

5. Key Findings and Interpretation

- **ERROR_1000** demonstrated the **strongest positive correlation (r = 0.345)** with DELAY, confirming its direct influence on system latency.
- **ERROR_1001** and **ERROR_1002** occurred more often but contributed less to delay magnitude.
- **CPU_LOAD** and **MEMORY_LEAK_LOAD** did not show strong correlations but could still intensify issues under peak loads.
- Reducing ERROR_1000-related failures is expected to deliver the most immediate and measurable improvement in performance metrics.

---

6. Recommendations

Short-Term (Immediate Actions)

- Enable detailed logging for ERROR_1000, capturing timestamps, request IDs, and stack traces.
- Configure proactive alerting for ERROR_1000 spikes and high DELAY (P95 threshold).
- Roll back or patch the faulty deployment contributing to error generation.

Medium-Term (Stabilization)

- Optimize Service B's database queries; introduce missing indexes to reduce latency.
- Expand **CI/CD testing** to include integration and performance test suites.
- Conduct **load testing** in a controlled environment prior to each release.

Long-Term (Preventive Measures)

- Deploy **distributed tracing** and centralized monitoring dashboards.
- Develop **runbooks** to standardize incident response and resolution.
- Implement continuous performance regression testing within the CI/CD pipeline.

---

7. Monitoring and Validation Strategy

Post-remediation validation is essential to confirm issue resolution and prevent recurrence.

- Continuously track **DELAY percentiles (P50, P95, P99)** and **ERROR_1000 frequency**.
- Measure performance improvement for at least **7 days post-deployment** and compare with baseline metrics.
- Set alert thresholds for both **error occurrences** and **response time deviations**.
- Conduct **weekly reviews** for one month to ensure system stability and sustained performance.

Appendix

Dataset Sample:
The first 10 records of the dataset are saved separately as rca_sample.csv.

Python Analysis Snippet:

```
import pandas as pd

df = pd.read_csv('root_cause_analysis.csv')

corr = df.corr()

error_counts = df[[c for c in df.columns if 'ERROR_' in c]].sum()

corr['DELAY'].plot(kind='bar')
```