High-Performance Data Exchange in Microservices

DISSERTATION

Submitted in partial fulfilment of the requirements of the
M. Tech. Software Engineering Degree programme

By

Shraddha Gulati
2020HS70018

Under the supervision of
Dharmesh Rana, Architect at SAP Labs India

Dissertation work carried out at
SAP Labs, Bangalore

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE
Pilani (Rajasthan) INDIA
(April 2024)

SE SAP ZG629T DISSERTATION

High-Performance Data Exchange in Microservices

Submitted in partial fulfilment of the requirements of the
M. Tech. Software Engineering Degree programme

By

Shraddha Gulati
2020HS70018

Under the supervision of
Dharmesh Rana, Architect at SAP Labs India

Dissertation work carried out at
SAP, Bangalore

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE
PILANI (RAJASTHAN)

(April 2024)

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI CERTIFICATE

This is to certify that the Project Work entitled **High-Performance Data Exchange in Microservices** and submitted by **Shraddha Gulati**, ID No. **2020HS70018** in partial fulfilment of the requirements of SESAP ZG629T Project Work, embodies the work done by her under my supervision.

Signature of the Supervisor

Name: **Dharmesh Rana**

Date: April 4th, 2024

Designation: **Architect**

ACKNOWLEDGEMENTS

It would be inadequate if I didn't extend my appreciation to everyone who contributed to the success of this endeavor. I am grateful to all those who supported me throughout this project and my educational journey.

Mr. Dharmesh's consistent guidance, supervision, and encouragement played a pivotal role in the completion of this project. I am thankful for the opportunity to be part of this program. I wish to convey my appreciation to Mrs. Sujith for her dedication and hard work on this project. Your valuable suggestions and insights for enhancing this project are sincerely valued.

I am grateful to Mr. Vinaya for his efforts in coordinating this project and providing timely feedback that enhanced its effectiveness. I am thankful for the knowledge and assistance provided by everyone involved during the implementation of the project.

Shraddha Gulati

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE,

PILANISESAP ZG629T PROJECT WORK

SECOND SEMESTER 2020-21

Project Work Title: **High-Performance Data Exchange in Microservices**

Name of Supervisor : Dharmesh Rana
Name of Student : Shraddha Gulati
ID No. of Student : 2020HS70018

Abstract

The objectives of my research are as follows:

o Minimize the latency in high-performance data interchange system in microservice architecture.

o Analyze how much is the latency when we call a service in microservice architecture due to data transmission technologies (JSON, XML, YAML).

o Exploration of new data transmission technologies to minimize the latency in the microservices execution (BSON, MessagePack, Protocol buffers, etc.)

o Exploration of HTTP/2 Protocol for microservice execution instead of HTTP/1.1 o Concluding the transmission technologies based on the scenarios analyzed.

Signature of Supervisor

Signature of Student

Table of Contents

Introduction 1
Problem statement 2
Objective of the project 3
Uniqueness of the project 4
Benefit to the Organization 5
Scope of Work 6
Solution Architecture 7
Resources needed for the project 11
Project plan and Deliverables 12
Work accomplished so far 14
Key Challenges Faced during the project 16
Potential Risks and Mitigation plan 17
Plan for remainder of the project 18
Bibliography/References 19
Appendices 20

LIST OF TABLES

Table 1*: Project plan and deliverables*..... 12

Table 2: *Work completed so far*..... 14

Table 3: *Plan for remaining work*..... 15

Chapter 1

Introduction

Initially, the standard architecture of a service was a HTTP/1.1 REST communication of a monolithic architecture, we understand that there is one service, and that service needs to communicate to a data source (OLAP, REDIS, Database). All the calls are made, they go to the monolithic service, and get executed. Later, looking at the latency, monolithic architecture was overpowered by the microservice architecture where we divided the service based on the modules, features and various other parameters.

In a true microservice architecture, we see that each microservice communicates to individual data sources. Later, we observed that there are redundancy issues of the code present at various microservices executing the similar logic. To avoid these redundancy issues, we introduced a common service where the common code/features can be bundled, and redundancy can be minimized among other microservices.

But, when the common service was introduced, we understand that the different microservices can have different way of communicating to various databases. For example, Service A connects with SQL database while Service B connects with OLAP system. Now, for a common service C, it is difficult to connect with the various database and maintain the database call protocols.
This architecture might be applicable where there are less microservices.

If we have a higher number of microservices, it becomes difficult to maintain different data source adapters. To reduce the above complexity, we can do microservice to microservice call via REST, so that the original logic will lie in its microservice and reuse the same logic by doing a REST call.

But, in high performance system; the microservices do not scale properly and this call can take a lot of time. Hence, we need to revisit the current architecture and look at the parameters that can be improved in reducing the latency of the data transmission among microservices.

Chapter 2

Problem Statement

Develop a Proof of Concept (PoC) to compare the performance of different data exchange formats, namely JSON, XML, and YAML, in a Node.js environment. The PoC aims to measure and analyze the latency of serialization and deserialization operations for each data format, providing insights into their efficiency and suitability for various use cases.

Key Objectives:

Implement serialization and deserialization functions for JSON, XML, and YAML data formats.
Measure the latency of serialization and deserialization operations for each data format.
Compare the performance of JSON, XML, and YAML based on latency measurements.
Present the findings in a clear and understandable format for analysis.

Expected Deliverables:

Functional Node.js application capable of measuring latency for serialization and deserialization.
Accurate latency measurements for JSON, XML, and YAML data formats.
Comparative analysis highlighting the performance differences between data exchange formats.

Constraints:

The PoC should handle large sample data efficiently to provide realistic latency measurements.
Latency measurements should be consistent and reliable across different data sizes.

Success Criteria:

The PoC accurately measures and compares the latency of serialization and deserialization operations for JSON, XML, and YAML.
Findings provide insights into the performance characteristics of each data exchange format.
Comparative analysis helps in selecting the appropriate data format based on specific use case requirements.

Assumptions:

Node.js environment provides consistent performance across different platforms and configurations.
Sample data used for testing is representative of real-world scenarios and sizes.

Chapter 3

Objective of the Project

Minimize the latency in high-performance data interchange system in microservice architecture.

o Analyze how much is the latency when we call a service in microservice architecture due to data transmission technologies (JSON, XML, YAML).

o Exploration of new data transmission technologies to minimize the latency in the microservices execution (BSON, MessagePack, Protocol buffers, etc.)

o Exploration of HTTP/2 Protocol for microservice execution instead of HTTP/1.1 o Concluding the transmission technologies based on the scenarios analyzed.

Chapter 4

Uniqueness of the Project

As per current progress done, the project stands out for its meticulous comparison of JSON, XML, and YAML data exchange formats tailored specifically for Node.js. By employing real-world scenarios and large data sizes, it ensures practical relevance. Additionally, it delves into Node.js intricacies, offering insights into platform-specific performance. Through objective evaluation and clear data visualization, developers gain actionable insights for optimizing their data handling strategies.

Chapter 5

Benefit to the Organization

The organization stands to benefit significantly from this project in several ways:

1. Optimized Performance: By understanding the performance characteristics of different data exchange formats in a Node.js environment, the organization can make informed decisions about which format to use in its applications. This can lead to optimized performance and reduced latency in data processing tasks.

2. Cost Savings: Improved performance can result in cost savings by reducing the computational resources required to handle data exchange operations. This is particularly important for organizations operating at scale, where even minor improvements in efficiency can translate into significant cost savings over time.

3. Enhanced Scalability: With insights into the performance of various data exchange formats, the organization can better plan for scalability. By choosing the most efficient format for its needs, the organization can ensure that its applications can scale effectively to handle increasing data volumes and user traffic.

4. Better Decision Making: The project provides valuable data and insights that enable better decision-making within the organization. Armed with knowledge about the pros and cons of different data exchange formats, stakeholders can make more informed choices about architecture, technology stack, and resource allocation.

5. Competitive Advantage: By leveraging optimized data exchange processes, the organization can gain a competitive advantage in the market. Faster, more efficient data handling can lead to improved user experiences, increased customer satisfaction, and a stronger position relative to competitors.

Overall, the project's benefits extend beyond technical improvements to encompass cost savings, scalability enhancements, better decision-making, and competitive advantages, all of which contribute to the organization's success and growth.

Chapter 6

Scope of Work

The scope of this project is currently identified as:

1. Researching and analyzing performance characteristics of JSON, XML, and YAML data exchange formats.
2. Implementing serialization and deserialization functions for each format in a Node.js environment.
3. Testing performance by measuring latency during data exchange operations.
4. Comparing performance metrics to identify the most efficient data format.
5. Documenting implementation details, findings, and recommendations.
6. Presenting results to stakeholders through a concise presentation.
7. Compiling a detailed report summarizing the project methodology and outcomes.

Chapter 7

Solution Architecture

The architecture of the Proof of Concept (PoC) for comparing data exchange formats in a Node.js environment can be depicted as follows:

1. Node.js Application:
   - At the core of the architecture is a Node.js application responsible for orchestrating the PoC activities, including data serialization, deserialization, performance measurement, and analysis.

2. Serialization and Deserialization Modules:
   - These modules handle the conversion of data between JSON, XML, and YAML formats. Each module provides functions for both serialization (converting JavaScript objects to strings) and deserialization (parsing strings back into JavaScript objects).

3. Performance Measurement Module:
   - This module captures timestamps before and after serialization/deserialization operations to calculate the latency. It measures the time taken by each operation and records the results for analysis.

4. Testing Framework Integration:
   - A testing framework, such as Mocha or Jest, can be integrated into the architecture to automate the execution of performance tests. Test cases are written to simulate various scenarios, such as different data sizes and complexities.

5. Comparison and Analysis Component:
   - This component analyzes the latency measurements obtained from different data exchange operations. It identifies performance trends and patterns, compares the performance metrics for each format, and draws conclusions based on the analysis.

6. Data Visualization Component:
   - To present the performance comparison results visually, a data visualization component is included. This may involve generating charts, graphs, or tables to illustrate the latency measurements and facilitate easier interpretation of the findings.

7. Documentation and Reporting Module:
   - A module dedicated to documenting the PoC's architecture, implementation details, test results, performance metrics, and analysis findings. This documentation serves as a comprehensive record of the PoC and aids in sharing insights with stakeholders.

8. External Dependencies:
   - The architecture may rely on external dependencies such as libraries for JSON, XML, and YAML handling, as well as Node.js packages for HTTP requests, file operations, and performance measurement.

Chapter 8

Resources Needed for the Project

1. The technology stack Hardware: Computers with ample processing power and storage.
2. Software: Node.js runtime, text editor/IDE, required packages.
3. Data Samples: JSON, XML, and YAML data for testing.
4. Documentation: Node.js and package documentation.
5. Version Control: Git or similar for code management.
6. Testing Environment: Environment for performance testing.
7. Monitoring Tools: Tools for system performance monitoring.
8. Team Collaboration Tools (Optional): Communication tools for team coordination.

Guidance from Supervisor and Examiner

Chapter 9

Project Plan & Deliverables

The details of the deliverables over the course of this project for 16 weeks have been defined as follows:
Table 1: Project Deliverables

S. No.	Tasks or Subtask to be done	Start Date - End Date	Planned Duration in Weeks	Specific Deliverable in terms of project
1.	Outline of the project and review from the Supervisors and Internal Examiners	Feb 5th – Feb 8th	0.7	Finalized Outline of the project
2.	Exploration on data exchange formats	Feb 8th – Feb 10th	0.3	Understanding about the transmission formats
3.	Explore and understand the latencies that data exchange format JSON brings in.	Feb 12th – Feb 17th	1	Understanding about the latencies JSON holds currently
4.	Explore and understand the latencies that that data exchange formats XML and YAML brings in.	Feb 19th – Feb 24th	1	Understanding about the latencies XML, YAML holds currently
5.	Explore and analyze how the latencies can affect the current architecture.	Feb 26th – Mar 2nd	1	Understanding how these latencies are becoming a bottleneck.
6.	Work on exploring techniques to minimize the latency.	Mar 4th- Mar 9th	1	Understanding on reducing the latencies
7.	Work on comparing the techniques with the transmission formats	Mar 11th – Mar 16th	1	Comparison of all the findings.
8.	Examine and evaluate the techniques with the help of a small PoC	Mar 18th – Mar 23rd	2	PoC on the comparison where all the latencies are well reflected and how to minimize those.
9.	Document and Submit the PoW of the study done.	Apr 1st – Apr 6th	1	PoW and Documentation of the study done.

Chapter 10

Work Accomplished So Far

During the first 8 weeks of the project, the following work has been accomplished.

Table 2: Work done so far

S. No.	Tasks or Subtask to be done	Start Date - End Date	Planned Duration in Weeks	Specific Deliverable in terms of project
1.	Outline of the project and review from the Supervisors and Internal Examiners	Feb 5th – Feb 8th	0.7	Finalized Outline of the project
2.	Exploration on data exchange formats	Feb 8th – Feb 10th	0.3	Understanding about the transmission formats
3.	Explore and understand the latencies that data exchange format JSON brings in.	Feb 12th – Feb 17th	1	Understanding about the latencies JSON holds currently
4.	Explore and understand the latencies that that data exchange formats XML and YAML brings in.	Feb 19th – Feb 24th	1	Understanding about the latencies XML, YAML holds currently
5.	Explore and analyze how the latencies can affect the current architecture.	Feb 26th – Mar 2nd	1	Understanding how these latencies are becoming a bottleneck.
6.	Work on exploring techniques to minimize the latency.	Mar 4th- Mar 9th	1	Understanding on reducing the latencies
7.	Work on comparing the techniques with the transmission formats	Mar 11th – Mar 16th	1	Comparison of all the findings.
8.	Examine and evaluate the techniques with the help of a small PoC	Mar 18th – Mar 23rd	2	PoC on the comparison where all the latencies are well reflected and how to minimize those.
9.	Document and Submit the PoW of the study done.	Apr 1st – Apr 6th	1	PoW and Documentation of the study done.

Chapter 11

Key Challenges Faced During the Project

1. **Performance Variability:** Different data sizes, structures, and complexities may lead to varying performance results across different data exchange formats. Managing this variability and ensuring consistent and reliable performance measurements can be challenging.
2. **Resource Limitations:** Limited computational resources, such as CPU, memory, and disk space, may impact the scalability and efficiency of the performance testing process. Optimizing resource utilization while conducting tests on large data sets is crucial but challenging.
3. **Dependency Management:** Managing dependencies, including libraries, frameworks, and external services, can be challenging, particularly when updates or changes are required during the project. Ensuring compatibility and stability across dependencies is essential for the success of the project.
4. **Cross-Platform Compatibility:** Ensuring compatibility and consistency of performance measurements across different operating systems and environments (e.g., Windows, Linux, macOS) can pose challenges. Differences in system configurations and behaviors may influence performance results.
5. **Data Integrity and Security:** Ensuring the integrity and security of the data used for testing, especially when dealing with sensitive or confidential information, is critical. Implementing measures to protect data privacy and prevent unauthorized access or leakage can be challenging.
6. **Interpretation of Results:** Analyzing and interpreting performance results accurately to draw meaningful conclusions and make informed decisions can be challenging, especially when dealing with large volumes of data and complex performance metrics.
7. **Documentation and Reporting:** Creating comprehensive documentation and reports that effectively communicate the project methodology, findings, and recommendations to stakeholders can be challenging. Ensuring clarity, accuracy, and relevance in documentation requires careful attention to detail.
8. **Time and Resource Constraints:** Managing project timelines and resource constraints, including budget and personnel, can be challenging. Balancing competing priorities and ensuring timely delivery of project milestones and deliverables is essential for project success.

Chapter 12

Potential Risk and Mitigation

Major potential risks in the project include:

1. Technical Complexity: Implementing serialization and deserialization functionalities for multiple data formats may pose technical challenges, leading to delays or errors in the project.

Mitigation: Conduct thorough research and planning to understand the technical requirements of each data format. Break down the implementation process into manageable tasks and allocate sufficient time for testing and troubleshooting.

2. Performance Variability: Variability in performance results across different data sizes and formats may affect the reliability of comparisons and undermine the validity of the findings.

Mitigation: Standardize testing methodologies and environments to minimize performance variability. Conduct extensive testing with a diverse range of data sets to identify patterns and trends, enabling more accurate performance assessments.

3. Dependency Management: Dependencies on external libraries or services may introduce compatibility issues, version conflicts, or security vulnerabilities, potentially disrupting project progress.

Mitigation: Maintain strict version control and regularly update dependencies to mitigate security risks and ensure compatibility. Develop contingency plans to address dependency-related issues promptly, such as alternative libraries or custom implementations.

Chapter 13

Plan For Remainder of The Project

Table 3: Plan for remaining work

S. No.	Tasks or Subtask to be done	Start Date - End Date	Planned Duration in Weeks	Specific Deliverable in terms of project
10.	Understand how HTTP/2 can boost the performance than HTTP/1.1	Apr 8th – Apr 13th	1	Understanding about HTTP/2 and HTTP/1.1
11.	Demonstrate and analyze the HTTP/2 with a small comparison PoC.	Apr 15th – Apr 20th	1	PoC on analyzing the speed of HTTP/2 over HTTP/1.1
12.	Bring all the PoC together to compare the overall performance of the data exchange happen with the newer methods.	Apr 22nd – Apr 27th	1	Merge of both the PoCs and comparison on better methods.
13.	Concluding the PoC done and, the scenarios analyzed.	Apr 29th – May 4th	1	Document the PoCs and conclusion of the better performance methods
16.	Start of the preparation of the product documentation and research paper	May 6th – May 11th	1	Product Documentation & Research Paper
17.	Working on the research paper	May 13th – May 18th	1	Product Documentation & Research Paper
18.	Working on the feedback from supervisor and examiner	May 20th – May 27th	1	Product Documentation & Research Paper
19.	Project Report Submission	May 27th		Product Documentation & Research Paper

APPENDIX

A.1: Data Size

1. Large Data Samples:
    - JSON Data Size: Approximately 263 MB
    - XML Data Size: Approximately 263 MB
    - YAML Data Size: Approximately 263 MB

A.2: Performance Metrics

1. JSON Serialization Time (ms): 267.65225
2. JSON Deserialization Time (ms): 406.195459
3. XML Serialization Time (ms): 8014.642833
4. XML Deserialization Time (ms): 7656.671084
5. YAML Serialization Time (ms): 18900.78
6. YAML Deserialization Time (ms): 1299.826083

REFERENCES

[1] **Official Documentation `fs, http, xml2js, and js-yaml`**

[2] **Research Papers and Articles**:
A Performance Comparison of JSON, XML, and YAML Data Formats in Node.js Applications."

FORM BITS – D - 1

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

< Semester 8 >

< SESAP ZG629T > DISSERTATION

MID SEMESTER EVALUATION FORM

Section I

(To be filled by the student and returned to the Supervisor)

ID No. 2020HS70018 Name of Student: Shraddha Gulati

Name of Supervisor: Dharmesh Rana

Name of the Examiner(s): Sujith Prathap

Dissertation Title: High-Performance Data Exchange in Microservices

Section II
(To be filled by the Supervisor in consultation with the examiner(s))

Comments on the dissertation from Examiner and Supervisor (Select Y or N)

1. **Quantum of work**
   a. **Justifiable as efforts for 8 weeks duration** **Y**
   b. **Work is in line with the commitments made in outline** **Y**

2. **Type of work**
   a. **Client assignment** **N**
   b. **Organization specific task** **Y**
   c. **General study project such as white paper** **Y**
   d. **Any other (kindly elaborate below in a line or two if Y)** **N**

3. **Nature of work**
   a. **Routine in nature** **N**
   b. **Involved creativity and rational thinking** **Y**
   Kindly elaborate below if answer for above is "Y"

4. **Evaluation methodology**
   a. **Evaluation done based on presentation to supervisor and examiner** **Y**
   b. **Evaluation done through Viva conducted by supervisor and examiner** **Y**
   c. **Student regularly interacted with supervisor and incorporated the suggestions made** **Y**

d. Brief description on the report submitted, quality of presentation and suggestions given for improvement

Impressive analysis of the Data Exchange formats. It was nice to see a working PoC that showed comparison of JSON, XML and YAML formats.

Suggestion: Since we are primarily focusing on communication over HTTP, let's include that in PoCs.

5. <u>Mid semester evaluation matrix</u>
Tick the appropriate box (1 is lowest and 5 is the highest)

Dimension Rank	1	2	3	4	5
Student abilities in general					
Understanding of the subject of dissertation				4	
Creative thinking: ability to come up with new ideas					**5**
Viva / Seminar presentation					
Communication ability					**5**
Organization of material					5
Response to review questions					**5**
Cohesive thinking ability					**5**
Report submitted					
Report structure and format					**5**
Technical content of the report					**5**
Explanation on the significance of the assignment					**5**
Analysis of alternative approaches				**4**	

Any other comments:

Date: April 4th 2024 Signature of examiner(s) Signature of Supervisor