A Comparative Study: Monolithic vs. Microservices Architectures

Aytan Gurbanova

July 2023

As mentioned in the previous report, the main purpose of the study is to conduct a performance comparison of two architectures by developing a simple e-commerce application. In this study, we have two types of data: test data, which consists of artificially created data for test scenarios, and real performance data, which comprises application metrics such as response time, number of threads, number of requests, and memory usage.

In this report, I will discuss the database tables that I have created for the test data, providing an overview of their structure and purpose. Since the actual performance data is not currently available, I will instead include similar tests conducted in related papers, which will provide me with an idea for my end-term paper.

Database Tables.

The application comprises five main tables. The structure of these tables and their relationships with one another can be seen in the figure below:

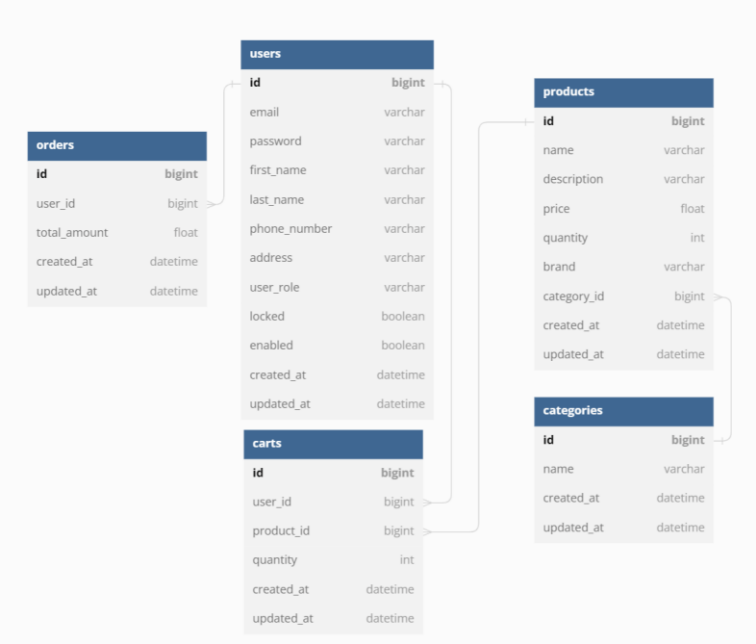

Figure 1. Database tables created for an e-commerce application. Source: Own Work

A high-level explanation of the tables above:

- **Users:** This table stores information about users who register and interact with the e-commerce application, including their email, password, name, contact details, role, and account status.
- **Categories:** This table represents the different categories or types of products available in the e-commerce application.
- **Products:** This table stores information about the products available in the e-commerce application, including their name, description, price, quantity, brand, associated category, and timestamps for creation and updates.
- **Carts:** This table represents the shopping carts of users, storing information about the products added to the cart, along with the quantity and timestamps. When a new product is added to the cart, it first checks for its existence. If the product already exists in the cart, only the quantity is increased. Otherwise, the product is added as a new record.

- **Orders:** This table is used to track customer orders, including the user who placed the order, the total order amount, and timestamps for creation and updates. For the sake of simplicity, the flow ends when an order is created; there is no payment integration, etc in the application.

Performance Comparison.

For this section, I have mainly analyzed two papers that have conducted a similar study. In [1], the authors analyzed the performance of two web applications, one based on monolith architecture and the other on microservices architecture. The authors defined two test scenarios to compare the performance of the applications based on metrics such as RAM memory availability, CPU usage, and response time. The results are presented in the form of tables. And [2] compares the performance and scalability of monolithic and microservice architectures in cloud deployment environments. The authors conducted three controlled experiments, varying the deployment environment and independent variables, and measured performance in terms of the number of requests processed per second. In both studies, JMeter tool was used for performance measurements.

In [1], they measured the performance on two test scenarios for both the monolith and microservices applications. The first scenario is relatively simple and consists of three steps. In the beginning, all items available in the shop are presented. After that, items having the letter "L" in the name are filtered out. The second scenario is more complex and consists of six steps. It includes adding items to the cart, removing items from the cart, and checking out.

For each scenario, the authors measured three distinct metrics: RAM memory availability on the server, the percentage of CPU usage, and response time from the end-user perspective. The results of these measurements were stored in six tables. In this paper, the data was stored only in tables and no statistical analysis were used. One example is as below:

Minute of a test	WebShop	WebShopMicroservice	ItemsService	ProducersService	StockService	ShopOrdersService
1	49.39	51.18	64.49	1.94	7.54	No requests
2	19.04	43.53	61.17	2.25	6.37	18.33
3	40.48	35.54	64.54	2.11	3.77	19.43

Figure 2. Server response time for second scenario. Source: [1]

In [2], the authors used throughput as the primary performance metric, which is calculated as the number of requests processed per second. The authors also measured infrastructure cost, which includes the cost of virtual machines, storage, and network traffic. Additionally, they collected data on response time, CPU usage, and memory usage, but did not use these metrics as primary performance measures. Unlike the first work, they used some scatter and bar plots in this study. They also used median statistics in plotting of throughput. The example is given below:

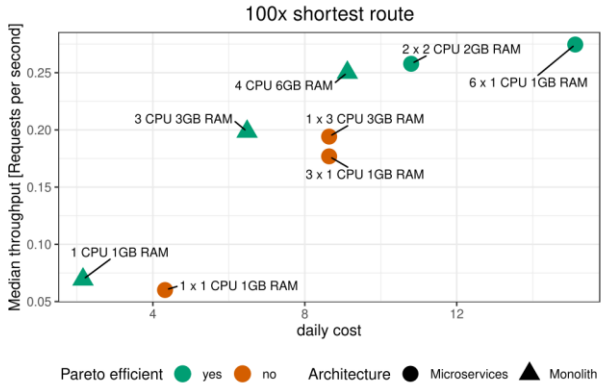

Figure 3. Throughput and cost in Azure spring cloud environment – route service. Source: [2]

For my end-term paper, I aim to measure similar metrics and use mostly tables, line and scatter plots for displaying my results.

References:

1. Barczak, A., & Barczak, M. Performance comparison of monolith and microservices based applications. In Proceedings of the 25th World Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI 2021).
2. G. Blinowski, A. Ojdowska, and A. Przybyłek, "Monolithic vs. Microservice Architecture: A Performance and Scalability Evaluation," Institute of Computer Science, Warsaw University of Technology, Warsaw, Poland, 2022.