GW

Course: **Guided Research I**

Title: **Scalability experiment of microservice architecture on an online bookstore
application**

Weekly Progress Report 5

Student: **Tural Mehtiyev**

Instructors: Dr. Stephen Kaisler, Dr. Jamal Hasanov

Table of Contents

1. List of planned steps for the previous week

- ➤ Order Management Service Bug Fixing: **Done**
- ➤ Finalization of Performance Testing and Data Collection: **Done**
- ➤ Statistical Analysis and Visualization: **Done**
- ➤ Architecture Diagram Enhancement: **Done**
- ➤ Finalization of Final Presentation: **Done**

2. Progress Details

Revised Application Diagrams
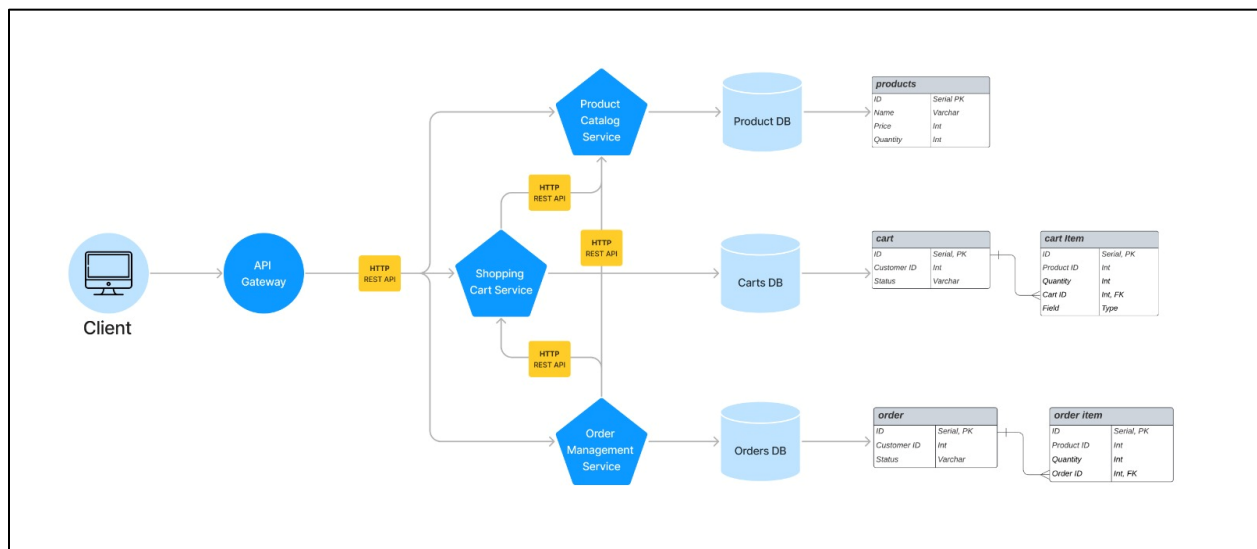
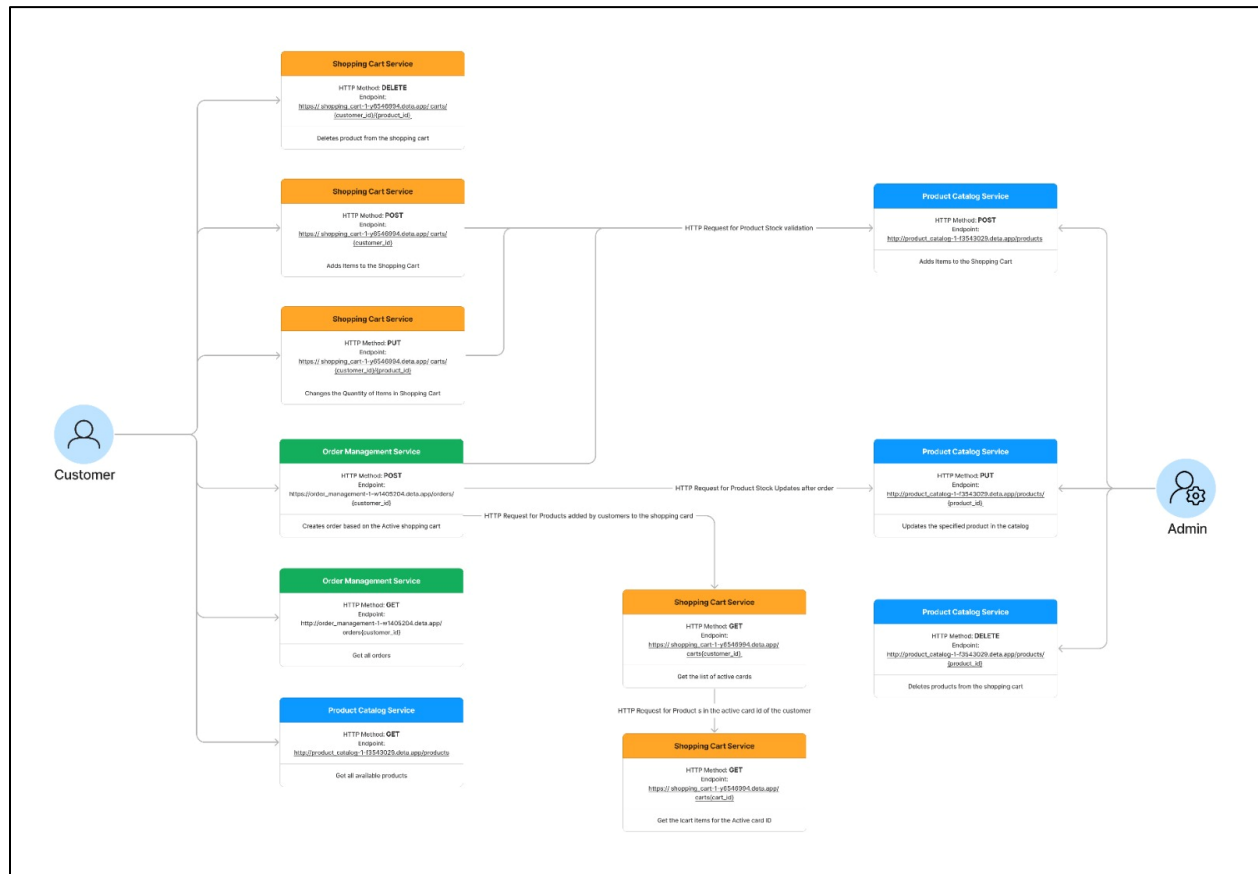## Architecture Description Diagram

Figure 1

Use Case Diagram

Figure 2

Order Flow Diagram

Figure 3

During this week, in addition to the revision of the architecture diagrams one of my main focus was on leveraging Apache JMeter as a tool to investigate and evaluate the application's performance. A series of test scenarios were formulated, executed, and data was systematically collected.

Formulation of Test Case Scenarios

Utilizing Apache JMeter, I have designed test scenarios that would provide valuable insights into the **effects of concurrent users on the application's response rate**. The primary objective was to understand how the application behaves under different loads and the corresponding response times.

Scenario 1:

Objective: Understand the application's behavior with a light load.

Start Threads Count: **10 users**

This signifies that 10 users will start simultaneously to send requests to the application.

Initial Delay: 0 seconds

The test begins without any initial delay.

Startup Time: 10 seconds

This parameter ensures that all users become active over a span of 10 seconds.
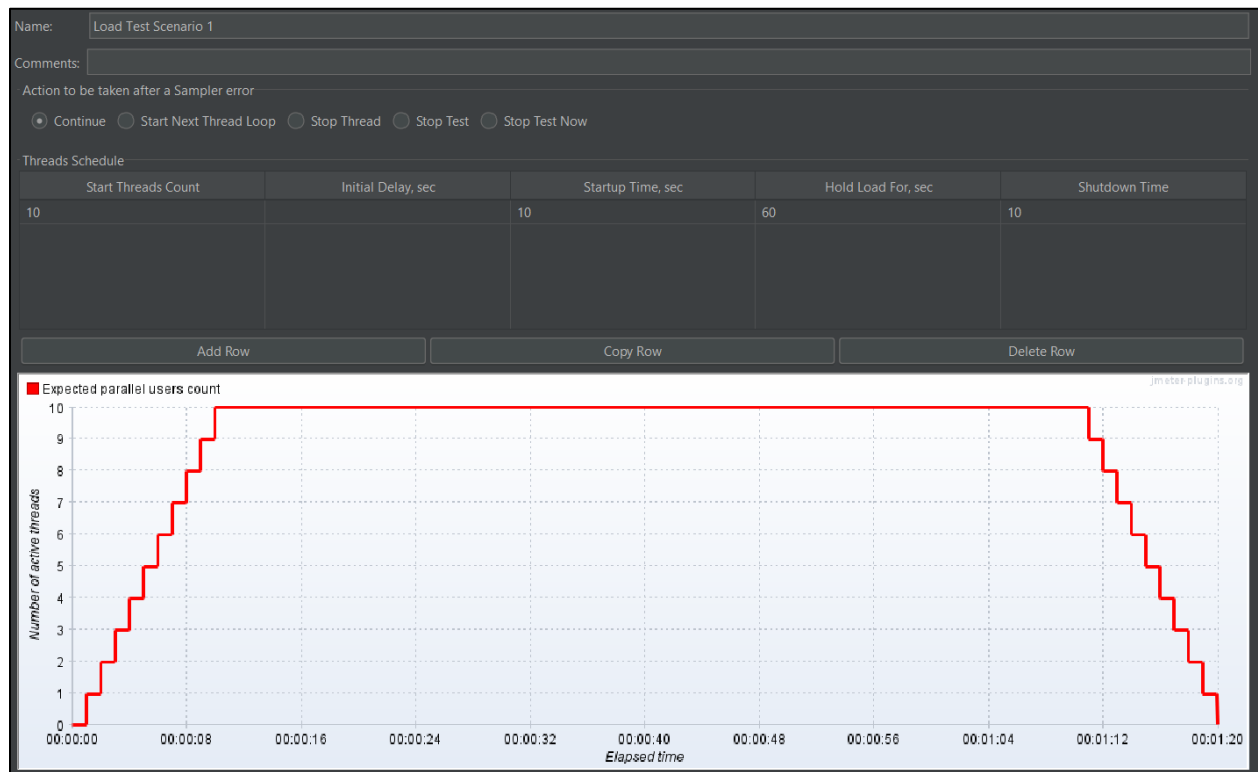
Figure 4

Scenario 2:

Objective: Assess the application's capability to manage increased load.

Start Threads Count: **50 users**

Initial Delay: 0 seconds

Startup Time: 10 seconds

Hold Load Time: 60 seconds

Shutdown Time: 10 seconds

Figure 5

Scenario 3:

Objective: Examine the application's robustness under a heavy user load.

Start Threads Count: **100 users**

Initial Delay: 0 seconds

Startup Time: 10 seconds
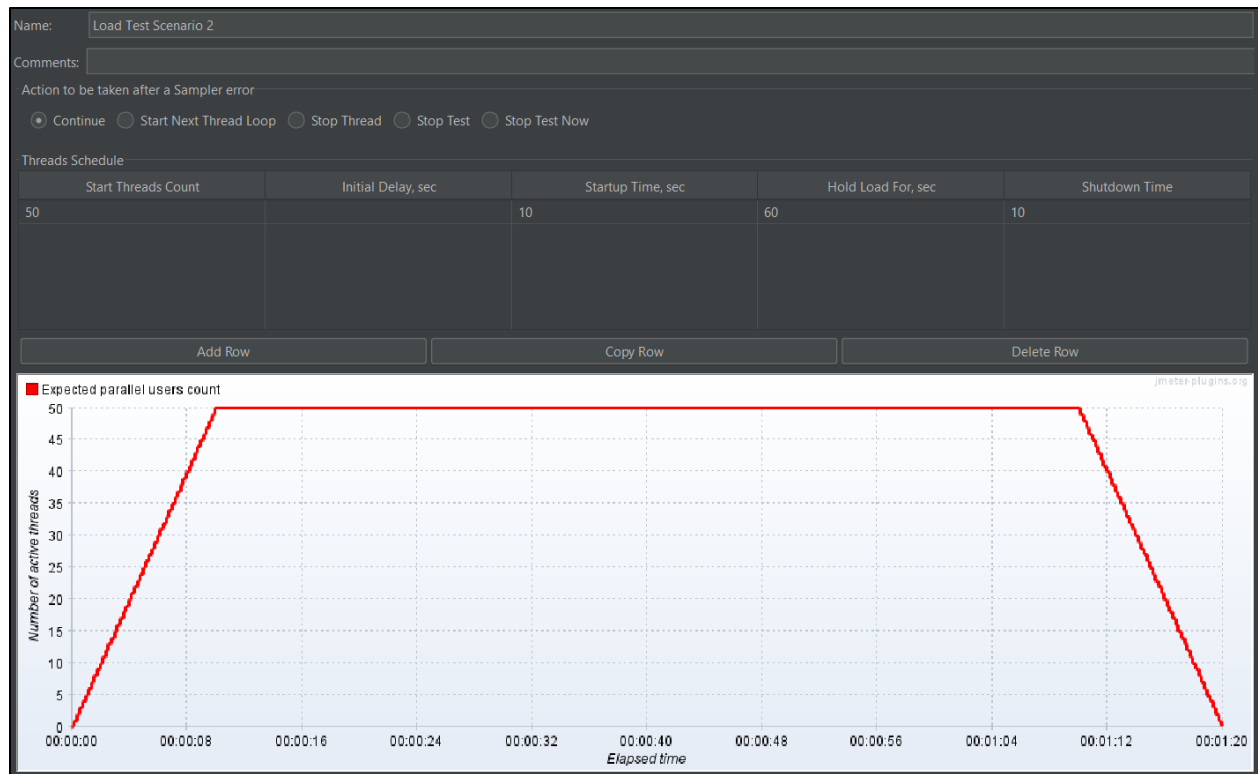
Hold Load Time: 60 seconds

Shutdown Time: 10 seconds
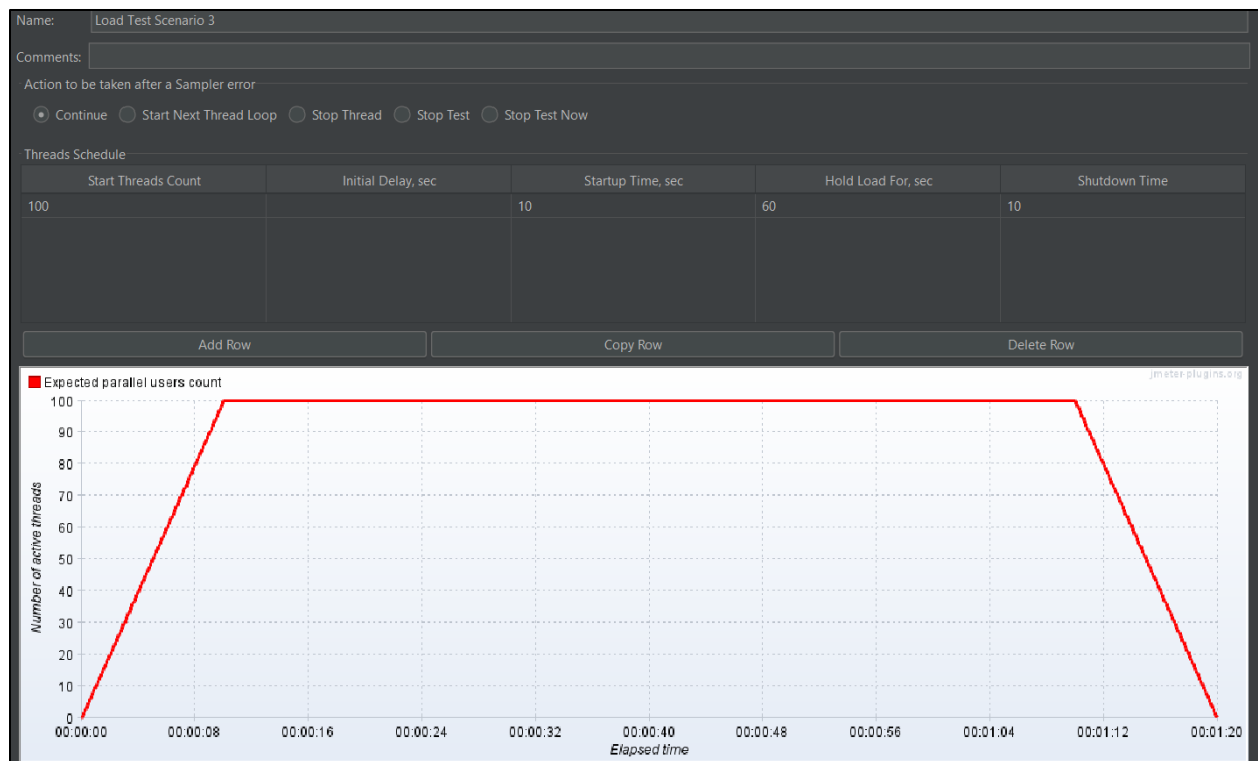
Figure 6

Data Collection

After the formulation and execution of the scenarios, Apache JMeter's listeners were employed to capture a wide array of data points. During my testing phase, I have employed many JMeter listeners that were instrumental in both data collection and visualization. Below I give descriptive information about some of them:

View Results Tree:

Detailed view of all request-response pairs, essential for debugging purposes. I have collected several data (**Sample time, Latency, Connect Time, Bytes sent/received, Success/error status**) about each request.

Figure 7

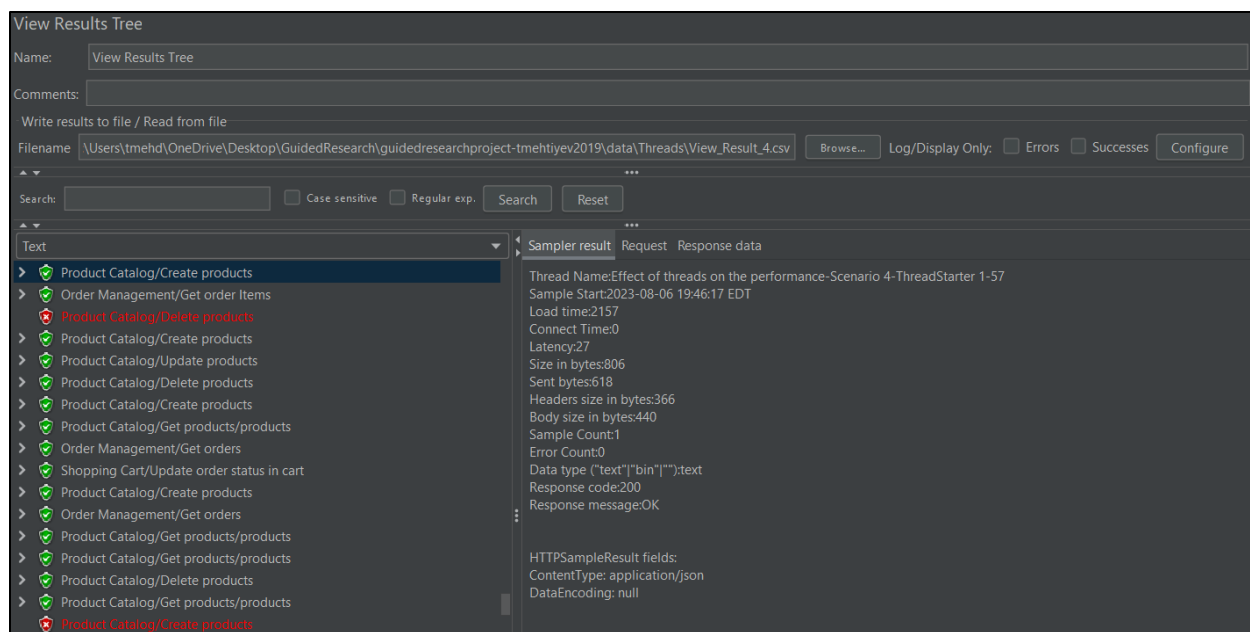## Summary Report & Aggregate Report:

Tabulated comprehensive statistics, including <u>Sample count, Average response time, Median response time, 90th percentile response time, Minimum and Maximum response times, Error percentage, Throughput (requests per second), Received KB/sec & Sent KB/sec,</u> providing a holistic overview of performance.

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/s…	Sent KB/sec	Avg. Bytes
Product Catalog/Get products/products	487	945	0	2620	582.34	2.05%	4.7/sec	3.69	2.11	796.0
Product Catalog/Create products	476	878	0	3033	654.56	5.04%	4.6/sec	3.55	2.74	781.5
Product Catalog/Update products	461	767	0	2803	578.20	7.38%	4.5/sec	2.32	2.85	526.6
Product Catalog/Delete products	448	772	0	3305	626.89	9.60%	4.4/sec	2.23	2.26	521.7
Shopping Cart/Get carts	439	1412	0	6110	1075.49	1.82%	4.3/sec	5.93	1.87	1411.8
Shopping Cart/Get cart items	425	1155	0	7054	911.33	4.71%	4.5/sec	7.08	1.96	1625.6
Shopping Cart/Add product to shopping cart	421	1068	0	5705	863.12	9.26%	4.4/sec	2.30	2.41	537.9
Shopping Cart/Update order status in cart	416	1064	0	6220	889.65	12.50%	4.4/sec	2.26	2.30	529.2
Order Management/Create orders	407	1328	0	8254	945.07	0.00%	4.3/sec	2.38	2.52	569.0
Order Management/Get orders	400	1192	0	5897	823.57	0.50%	4.3/sec	3.33	1.91	797.6
Order Management/Get order Items	393	1209	0	7077	879.23	1.27%	4.2/sec	3.17	1.91	774.2
TOTAL	4773	1063	0	8254	836.98	4.97%	46.4/sec	36.56	23.78	806.3

Figure 8

Aggregate Graph:

A visual representation of key metrics like <u>Average, Median, 90th percentile, and Min/Max response times, Throughput, Standard deviation</u>, providing graphical insights for quick trend analysis.

Figure 9

Response Time Graph:

Graphical portrayal of <u>Response time against time or sample number, Deviation, Throughput, Median, Average, and 90th percentile values, aiding in visual detection of patterns or anomalies in response time</u>.
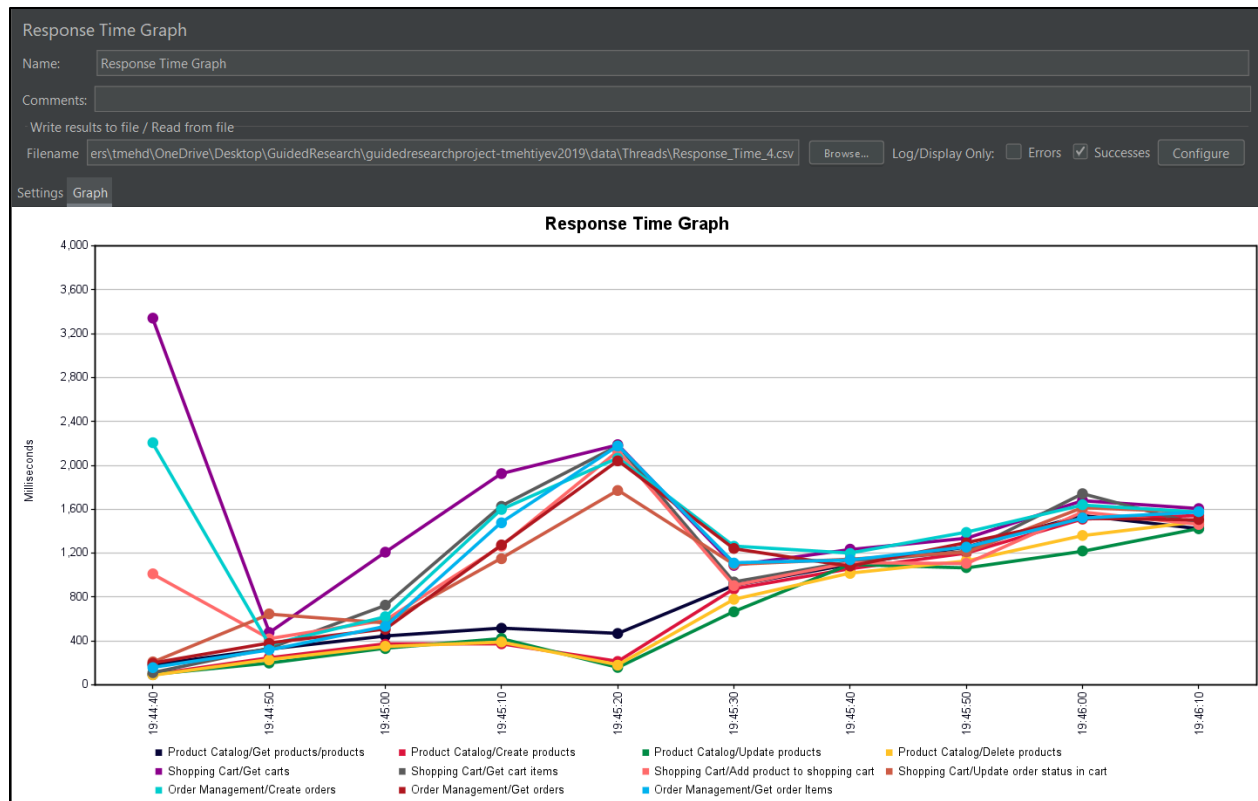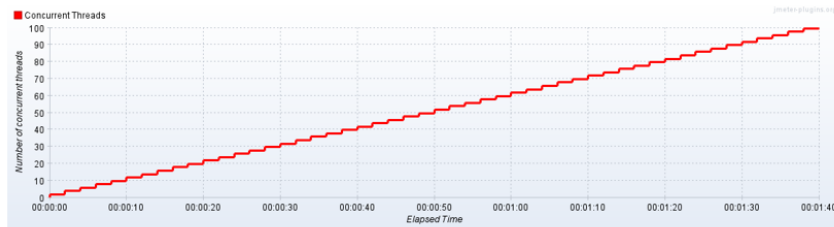
Figure 10

Visual Description | Effects of Ramp-Up Steps Count on Response Time
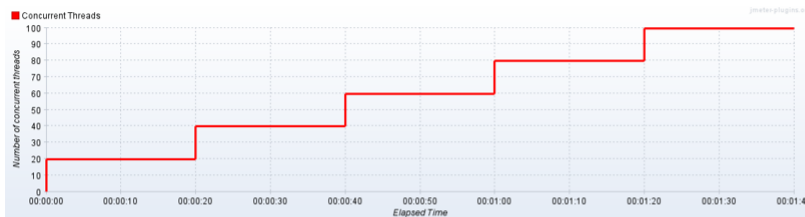
Figure 11

Figure 12

Figure 13

Statistical Analysis | Effects of Threads Count on Response Rate

With the refined data, I have applied various statistical techniques to extract meaning from the numbers. By leveraging the **Central Limit Theorem**, **hypothesis testing**, I have gained a deeper understanding of the application's performance metrics (mainly response rate) under different user loads.

Below are the details of the response rate distribution for each test scenario:
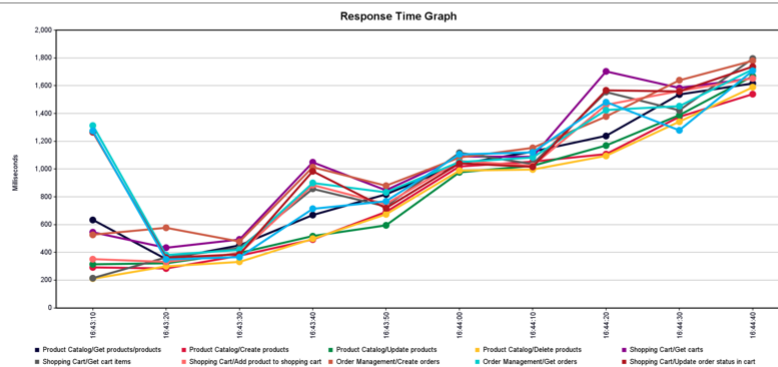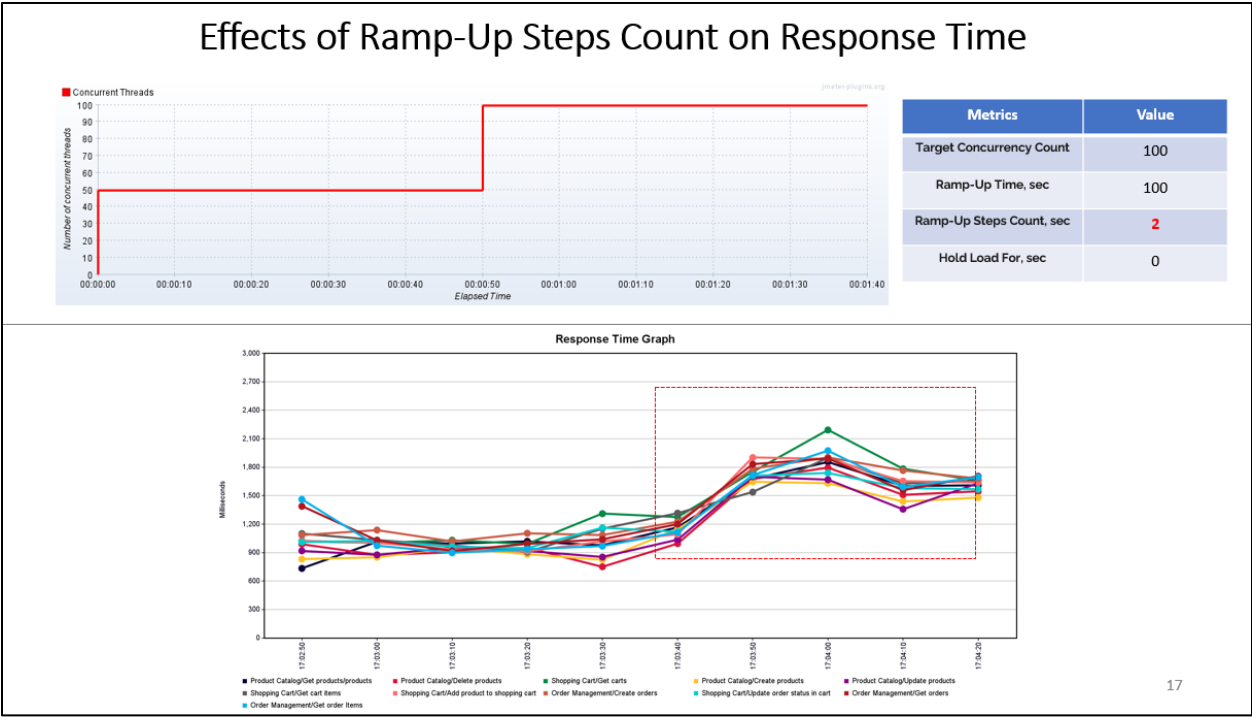
Figure 14

Figure 15

Figure 16

Application of the Central Limit Theorem

The Central Limit Theorem (CLT):

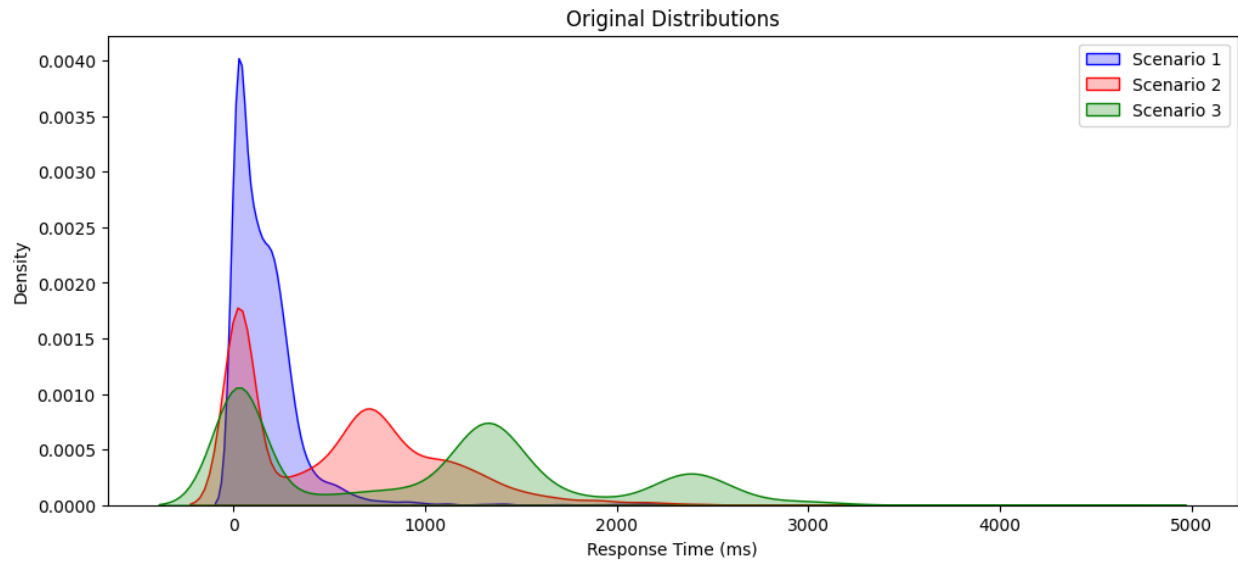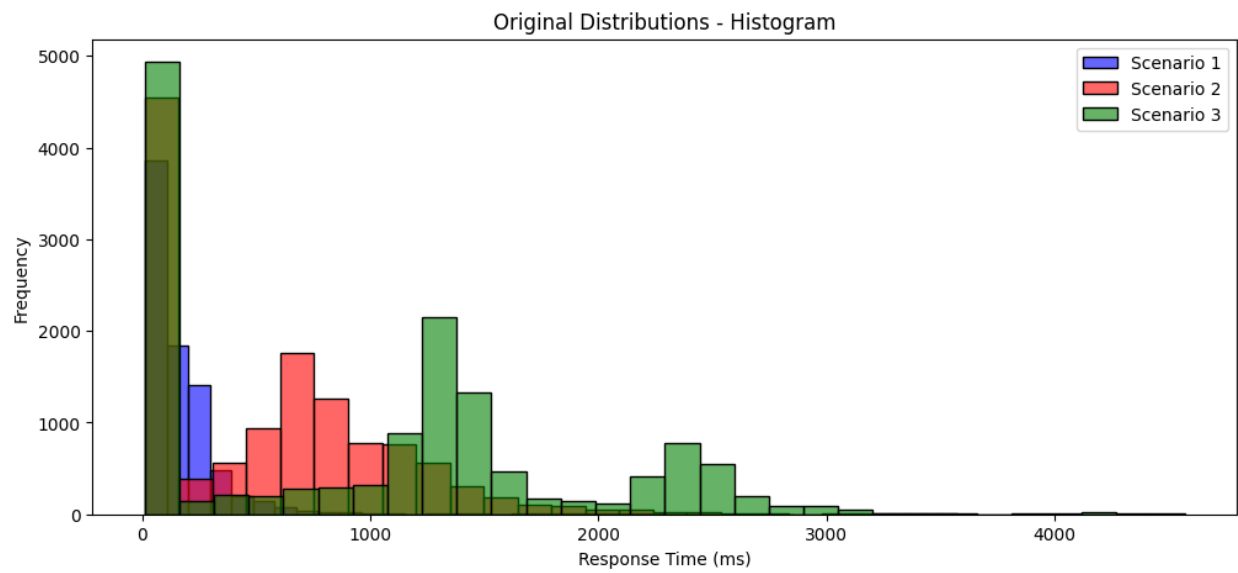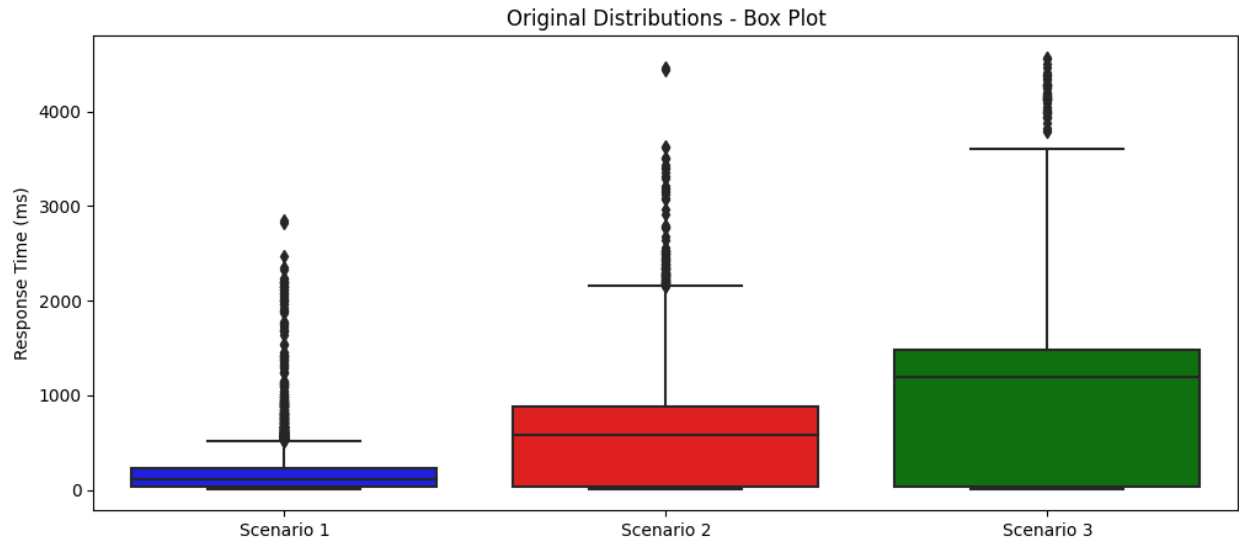The Central Limit Theorem is a statistical principle stating that, given a sufficiently large sample size, the distribution of the sample means of independent and identically distributed random variables will be approximately normal, irrespective of the original distribution of the variables.

Why Did I Apply the Central Limit Theorem?

➢ **Normality Assumption**: Many statistical techniques and tests assume the data to be normally distributed. If the original data is not normal, applying the CLT helps meet this assumption by working with the distribution of sample means instead.

➢ **Statistical Robustness**: By working with a normally distributed dataset (the distribution of sample means), my hypothesis tests and statistical inferences become more reliable and robust.

➢ **Simplification**: The normal distribution is well-understood, and its properties are widely used in statistics. By ensuring that my dataset adheres to a normal distribution (via the CLT), analysis and interpretation become more straightforward.

Details from the Application:

<u>Sample Size</u>: In the sampling process, the sample size (number of observations in each sample) is set as 50.

<u>Number of Samples</u>: I've collected 1,000 samples from each scenario. This means that I've drawn 1,000 separate samples, each of 50 observations, and then calculated their means.

<u>Random Sampling</u>: I've drawn random samples from the original data, ensuring that each sample is drawn independently.

The application of the Central Limit Theorem allowed me to make statistical inferences using the normal distribution, which has desirable properties. By applying the CLT, I'm aiming to leverage these properties to make more reliable decisions and interpretations based on my JMeter test results.

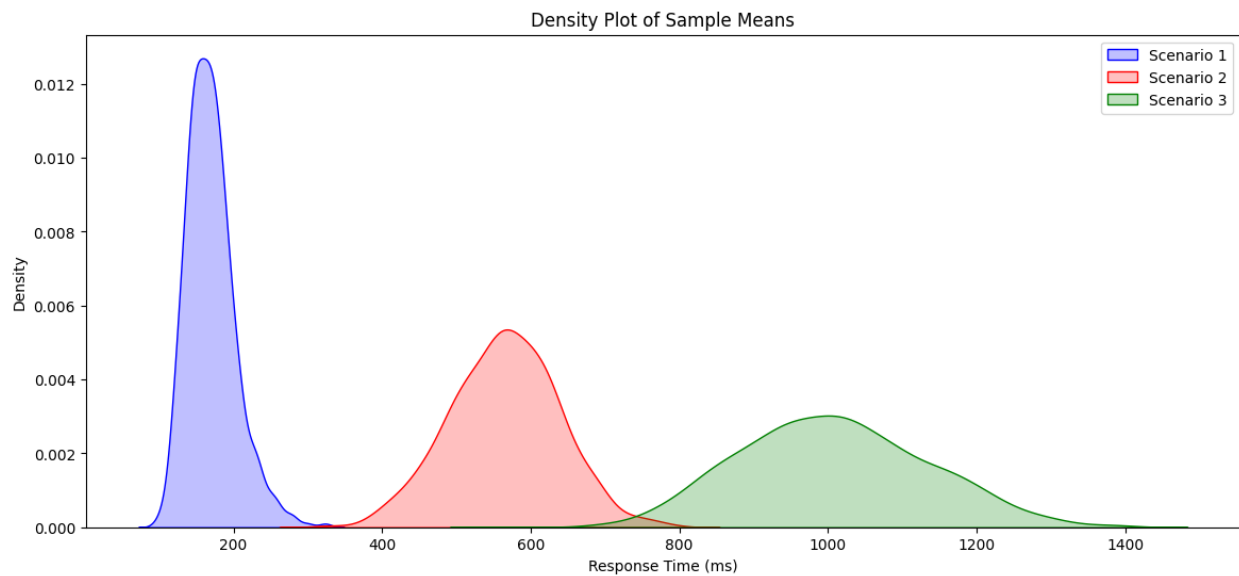Below you can find the distribution for each test scenario after the application of CLT.
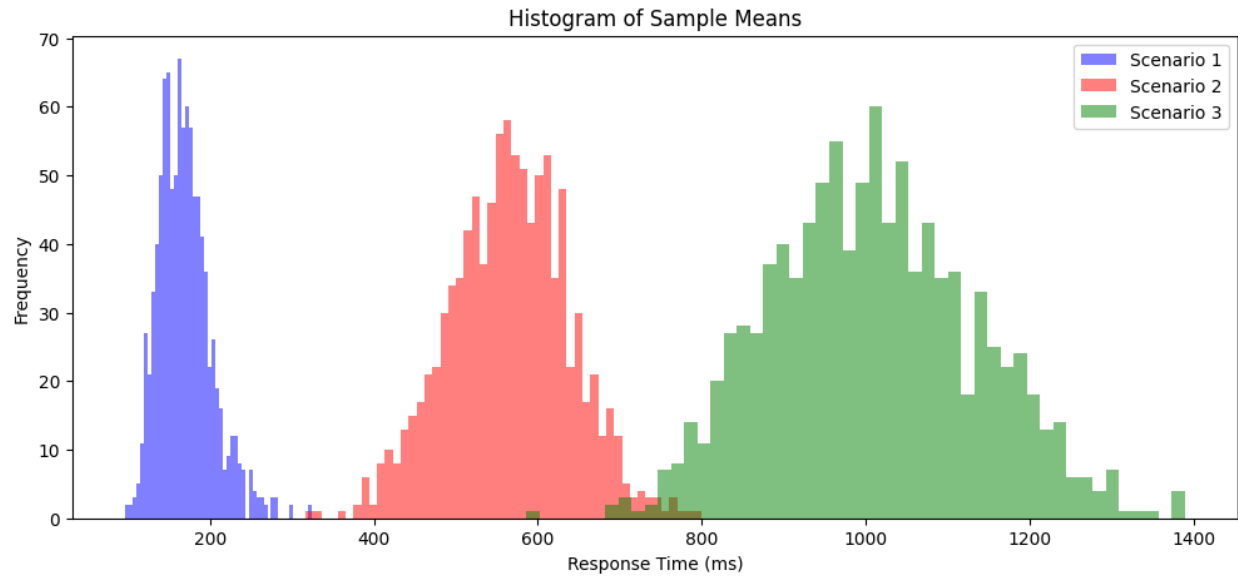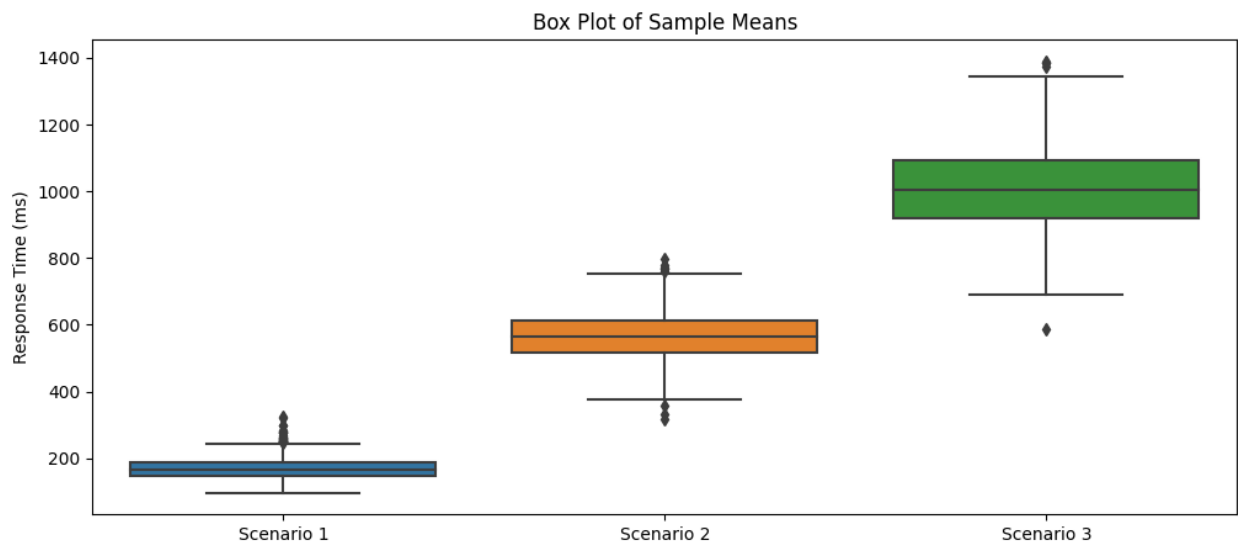

Figure 17

Figure 18

Figure 19

Hypothesis Testing:

Methodology: Hypothesis Testing

1

Null Hypothesis (H0): The means of Scenario 1 and Scenario 2 are equal.

Alternative Hypothesis (Ha): The means of Scenario 1 and Scenario 2 are not equal.

- **Mean after CLT for Scenario 1:** 168.2088
- **Standard Error:** 2.4462

- **Mean after CLT for Scenario 2:** 563.1481
- **Standard Error:** 4.7189

- **Alpha Score:** 5%
- **T-statistic (t-score):** -155.4008
- **P-value:** 0.0000

- **Result:** Reject H0. There's a significant difference between Scenario 1 and Scenario 2.

2

Null Hypothesis (H0): The means of Scenario 2 and Scenario 3 are equal.

Alternative Hypothesis (Ha): The means of Scenario 2 and Scenario 3 are not equal.

- **Mean after CLT for Scenario 2:** 563.1481
- **Standard Error:** 4.7189

- **Mean after CLT for Scenario 3:** 1015.5730
- **Standard Error:** 7.6377

- **Alpha Score:** 5%
- **T-statistic (t-score):** -94.1281
- **P-value:** 0.0000

- **Result:** Reject H0. There's a significant difference between Scenario 2 and Scenario 3.

3

Null Hypothesis (H0): The means of Scenario 1 and Scenario 3 are equal.

Alternative Hypothesis (Ha): The means of Scenario 1 and Scenario 3 are not equal.

- **Mean after CLT for Scenario 1:** 168.2088
- **Standard Error:** 2.4462

- **Mean after CLT for Scenario 3:** 1015.5730
- **Standard Error:** 7.6377

- **Alpha Score:** 5%
- **T-statistic (t-score):** -196.2082
- **P-value:** 0.0000

- **Result:** Reject H0. There's a significant difference between Scenario 1 and Scenario 3.

Figure 20

This week marked a significant advancement in my project, primarily in the realms of performance testing and statistical analysis. Through the formulation of test case scenarios in Apache JMeter and the collection of data using various listeners, I managed to preprocess the data, apply statistical techniques, and finally derive essential results via hypothesis testing.

I carried out hypothesis testing to assess whether the mean response times of the three scenarios differed significantly, and the results indicated a significant difference in mean response times across all three scenarios.

The implication of finding a significant difference in mean response times across the three scenarios is that the application's performance varies under different user loads. Therefore, as the number of concurrent users changes, it has a noticeable impact on the application's responsiveness. This knowledge is crucial as it points to areas that might require optimization to ensure consistent performance across varying user loads.