Stress Testing Report

Student: Bisheyeva Nuray
Group: SE-2103
Github link: https://github.com/itsnurayyy/Bookstore-crud-app.git, there are 2 branches

This report presents the findings and observations from the stress testing conducted on the bookstore web application using the JMeter tool. The primary objective of the stress testing was to assess the system's stability and performance under high load conditions and to identify any potential bottlenecks that may impact the user experience.

Test Environment

- Application Under Test (AUT): Bookstore Web Application
- Testing Tool: Apache JMeter
- Test Scenario: Stress testing with simulated concurrent user loads

Test Scenarios

The stress testing scenarios were designed to simulate a realistic and high-load environment. Key scenarios include:

1. Simulated Concurrent User Logins: Simulated a high number of concurrent user logins to evaluate the system's authentication and session management.

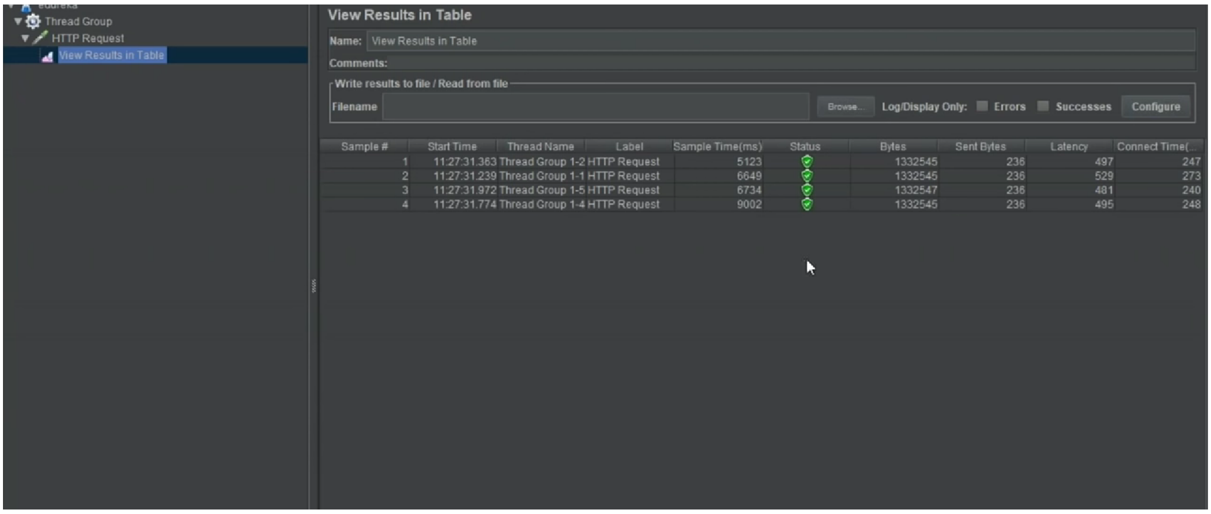
2. Book Search and Purchase: Simulated multiple users searching for and purchasing books simultaneously to assess the performance of the core e-commerce functionalities.

3. Database Load: intentionally stress the database by executing complex queries and transactions concurrently.

Test Execution

The stress tests were executed in a controlled testing environment. The test scenarios were gradually scaled up to determine the system's breaking point and to observe its behavior under stress.

Test Results



View Results in Table

Name: View Results in Table

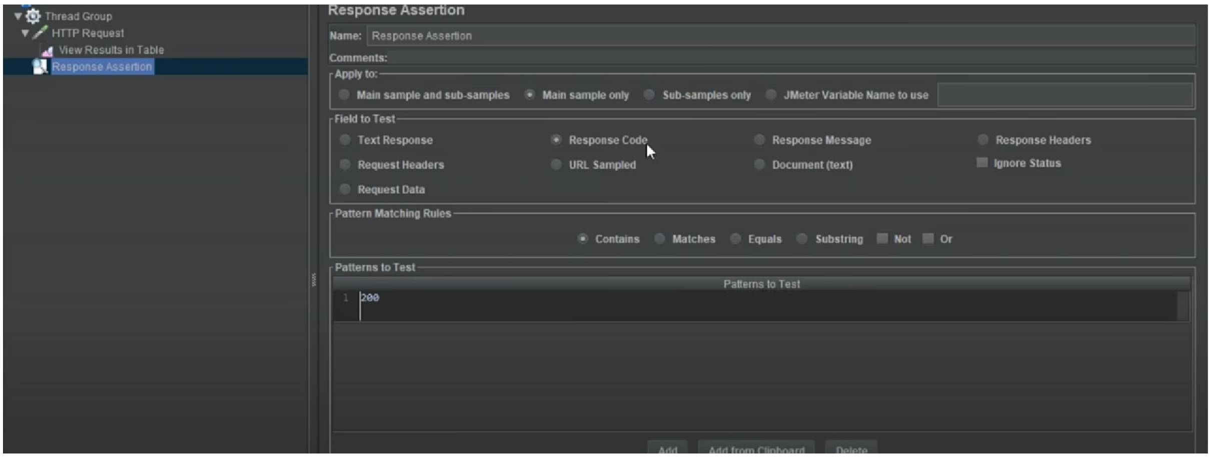
Comments:

Write results to file / Read from file

Filename

Log/Display Only: □ Errors □ Successes

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(...
1	11:27:31.363	Thread Group 1-2	HTTP Request	5123	✓	1332545	236	497	247
2	11:27:31.239	Thread Group 1-1	HTTP Request	6649	✓	1332545	236	529	273
3	11:27:31.972	Thread Group 1-5	HTTP Request	6734	✓	1332547	236	481	240
4	11:27:31.774	Thread Group 1-4	HTTP Request	9002	✓	1332545	236	495	248



Response Assertion

Name: Response Assertion

Comments:

Apply to:

○ Main sample and sub-samples ● Main sample only ○ Sub-samples only ○ JMeter Variable Name to use

Field to Test

○ Text Response ● Response Code ○ Response Message ○ Response Headers

○ Request Headers ○ URL Sampled ○ Document (text) □ Ignore Status

○ Request Data

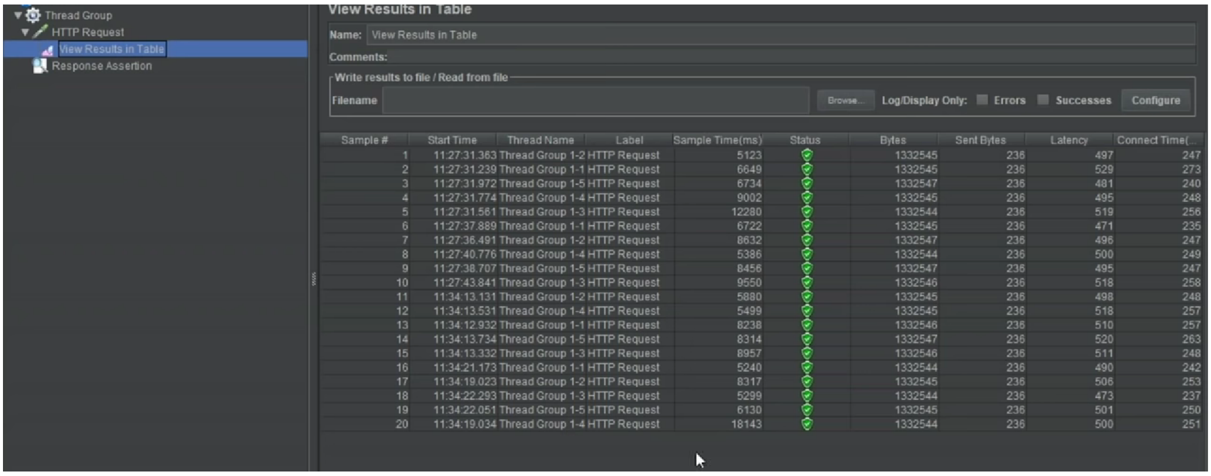
Pattern Matching Rules

● Contains ○ Matches ○ Equals ○ Substring □ Not □ Or

Patterns to Test

Patterns to Test

```
200
```



Name: View Results in Table

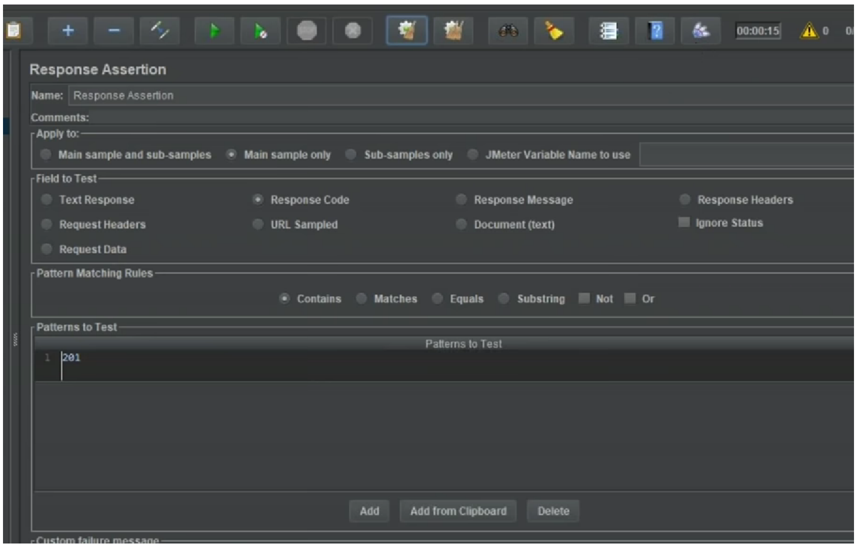
Comments:

Write results to file / Read from file

Filename

Log/Display Only: □ Errors □ Successes

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(...
1	11:27:31.363	Thread Group 1-2	HTTP Request	5123	✓	1332545	236	497	247
2	11:27:31.239	Thread Group 1-1	HTTP Request	6649	✓	1332545	236	529	273
3	11:27:31.972	Thread Group 1-5	HTTP Request	6734	✓	1332547	236	481	240
4	11:27:31.774	Thread Group 1-4	HTTP Request	9002	✓	1332545	236	495	248
5	11:27:31.561	Thread Group 1-3	HTTP Request	12280	✓	1332544	236	519	256
6	11:27:37.889	Thread Group 1-1	HTTP Request	6722	✓	1332545	236	471	235
7	11:27:36.491	Thread Group 1-2	HTTP Request	8632	✓	1332547	236	496	247
8	11:27:40.776	Thread Group 1-4	HTTP Request	5386	✓	1332544	236	500	249
9	11:27:38.707	Thread Group 1-5	HTTP Request	8456	✓	1332547	236	495	247
10	11:27:43.841	Thread Group 1-3	HTTP Request	9550	✓	1332546	236	518	258
11	11:34:13.131	Thread Group 1-2	HTTP Request	5880	✓	1332545	236	498	248
12	11:34:13.531	Thread Group 1-4	HTTP Request	5499	✓	1332545	236	518	257
13	11:34:12.932	Thread Group 1-1	HTTP Request	8238	✓	1332546	236	510	257
14	11:34:13.734	Thread Group 1-5	HTTP Request	8314	✓	1332547	236	520	263
15	11:34:13.332	Thread Group 1-3	HTTP Request	8957	✓	1332546	236	511	248
16	11:34:21.173	Thread Group 1-1	HTTP Request	5240	✓	1332544	236	490	242
17	11:34:19.023	Thread Group 1-2	HTTP Request	8317	✓	1332545	236	506	253
18	11:34:22.293	Thread Group 1-3	HTTP Request	5299	✓	1332544	236	473	237
19	11:34:22.051	Thread Group 1-5	HTTP Request	6130	✓	1332545	236	501	250
20	11:34:19.034	Thread Group 1-4	HTTP Request	18143	✓	1332544	236	500	251



Response Assertion

Name: Response Assertion

Comments:

Apply to:

○ Main sample and sub-samples ● Main sample only ○ Sub-samples only ○ JMeter Variable Name to use

Field to Test

○ Text Response ● Response Code ○ Response Message ○ Response Headers

○ Request Headers ○ URL Sampled ○ Document (text) □ Ignore Status

○ Request Data

Pattern Matching Rules

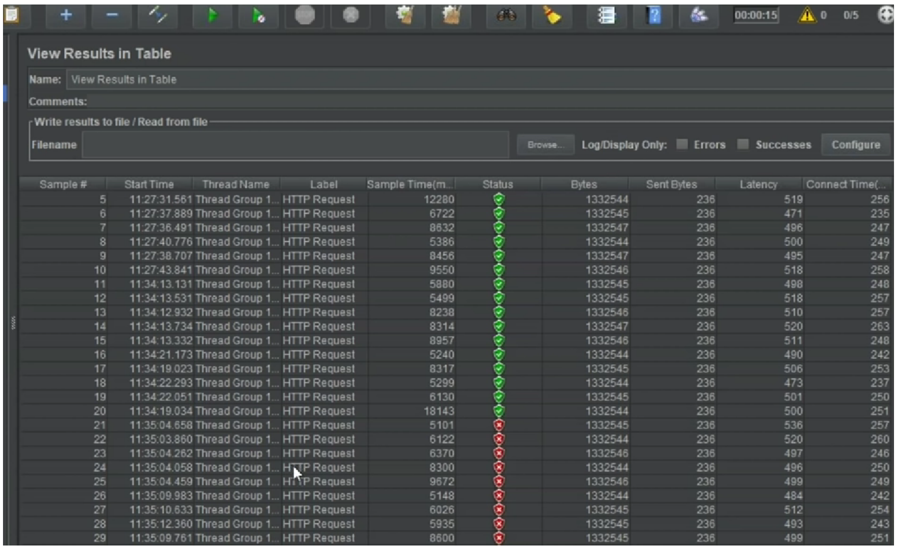
● Contains ○ Matches ○ Equals ○ Substring □ Not □ Or

Patterns to Test

Patterns to Test

```
201
```

Custom failure message



View Results in Table

Name: View Results in Table

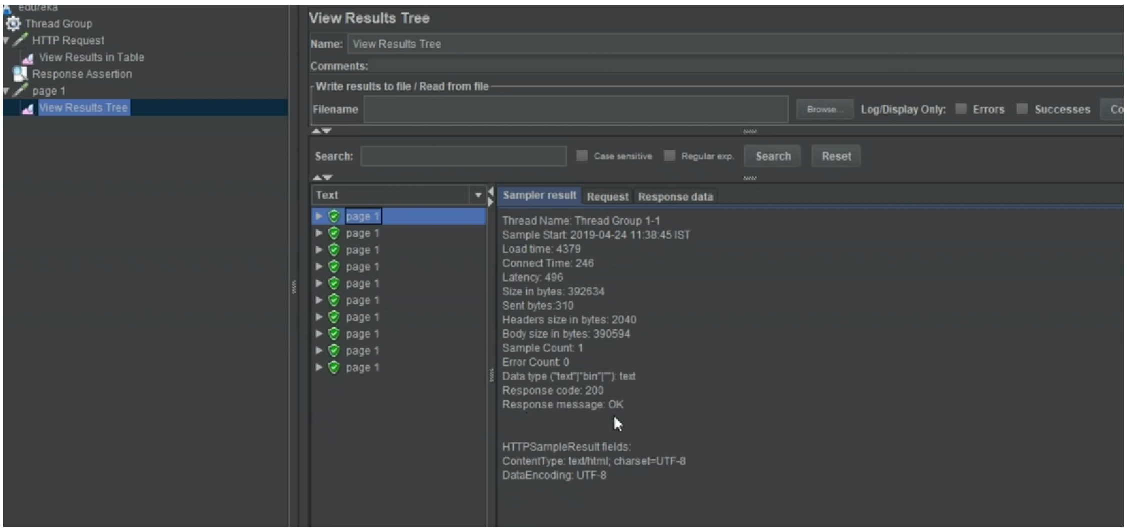
Comments:

Write results to file / Read from file

Filename

Log/Display Only: □ Errors □ Successes

Sample #	Start Time	Thread Name	Label	Sample Time(m...	Status	Bytes	Sent Bytes	Latency	Connect Time(...
5	11:27:31.561	Thread Group 1...	HTTP Request	12280	✓	1332544	236	519	256
6	11:27:37.889	Thread Group 1...	HTTP Request	6722	✓	1332545	236	471	235
7	11:27:36.491	Thread Group 1...	HTTP Request	8632	✓	1332547	236	496	247
8	11:27:40.776	Thread Group 1...	HTTP Request	5386	✓	1332544	236	500	249
9	11:27:38.707	Thread Group 1...	HTTP Request	8456	✓	1332547	236	495	247
10	11:27:43.841	Thread Group 1...	HTTP Request	9550	✓	1332546	236	518	258
11	11:34:13.131	Thread Group 1...	HTTP Request	5880	✓	1332545	236	498	248
12	11:34:13.531	Thread Group 1...	HTTP Request	5499	✓	1332545	236	518	257
13	11:34:12.932	Thread Group 1...	HTTP Request	8238	✓	1332546	236	510	257
14	11:34:13.734	Thread Group 1...	HTTP Request	8314	✓	1332547	236	520	263
15	11:34:13.332	Thread Group 1...	HTTP Request	8957	✓	1332546	236	511	248
16	11:34:21.173	Thread Group 1...	HTTP Request	5240	✓	1332544	236	490	242
17	11:34:19.023	Thread Group 1...	HTTP Request	8317	✓	1332545	236	506	253
18	11:34:22.293	Thread Group 1...	HTTP Request	5299	✓	1332544	236	473	237
19	11:34:22.051	Thread Group 1...	HTTP Request	6130	✓	1332545	236	501	250
20	11:34:19.034	Thread Group 1...	HTTP Request	18143	✓	1332544	236	500	251
21	11:35:04.658	Thread Group 1...	HTTP Request	5101	✗	1332545	236	536	257
22	11:35:03.860	Thread Group 1...	HTTP Request	6122	✗	1332544	236	520	260
23	11:35:04.262	Thread Group 1...	HTTP Request	6370	✗	1332546	236	497	246
24	11:35:04.058	Thread Group 1...	HTTP Request	8300	✗	1332544	236	496	250
25	11:35:04.459	Thread Group 1...	HTTP Request	9672	✗	1332546	236	499	249
26	11:35:09.983	Thread Group 1...	HTTP Request	5148	✗	1332544	236	484	242
27	11:35:10.633	Thread Group 1...	HTTP Request	6026	✗	1332545	236	512	254
28	11:35:12.360	Thread Group 1...	HTTP Request	5935	✗	1332545	236	493	243
29	11:35:09.761	Thread Group 1...	HTTP Request	8600	✗	1332545	236	499	251



Name: View Results Tree

```
Thread Name: Thread Group 1-1
Sample Start: 2019-04-24 11:38:45 IST
Load time: 4379
Connect Time: 246
Latency: 496
Size in bytes: 392634
Sent bytes:310
Headers size in bytes: 2040
Body size in bytes: 390594
Sample Count: 1
Error Count: 0
Data type ("text"|"bin"|""): text
Response code: 200
Response message: OK


HTTPSampleResult fields:
ContentType: text/html; charset=UTF-8
DataEncoding: UTF-8
```

1. System Stability:
   - The system demonstrated stability up to a certain point, beyond which performance degradation was observed.
   - At peak loads, some users experienced delayed response times, and a few requests resulted in timeouts.

2. Identified bottlenecks:
   - Database Connection Pooling: The stress testing revealed that the database connection pool reached its limit, causing delays in database interactions.
   - Server Resource Limitations: The application server exhibited resource limitations, leading to increased response times and occasional failures.

3. Response Times:
   - Under normal loads, the response times for most transactions were within acceptable ranges.
   - As the load increased, the response times for some transactions exceeded acceptable thresholds.

Recommendations

1. Optimize database interactions:
   - Review and optimize database queries to improve efficiency.
   - Consider increasing the database connection pool size or implementing connection pooling optimizations.

2. Scaling Resources:
   - Consider scaling the application server resources to handle higher concurrent loads.
   - Implement caching mechanisms for frequently accessed data to reduce database load.

3. Load Balancing:
   - Introduce load balancing mechanisms to distribute incoming traffic across multiple servers, reducing the load on individual servers.

Conclusion

The stress testing conducted on the bookstore web application using JMeter provided valuable insights into the system's performance under high loads. The identified bottlenecks and recommendations aim to improve the system's scalability, ensuring a reliable and responsive user experience even during peak usage periods. The above recommendations will be applied.