Systems in Cloud Computing - Project Report

Francisco Freitas 60313
Teresa Ribeiro 60756

November 2024

1 Introduction

The goal of this project was to port the existing TuKano web application to the Microsoft Azure Cloud platform, ensuring it meets the requirements of scalability, performance, and high availability. Our solution leveraged Azure's PaaS offerings, including Azure Cache for Redis, Azure Blob Storage, and Azure Cosmos DB. We explored both relational (PostgreSQL) and NoSQL backends for data storage, while also implementing Redis for caching to improve performance.

2 Implementation Choices

2.1 Caching Mechanism: Azure Redis Cache

To enhance the performance of read-heavy operations, we used Azure Cache for Redis. This caching mechanism was particularly effective in reducing latency for frequently accessed data, such as user feeds and metadata associated with shorts. Redis cache enabled us to serve data directly from memory and this way, reducing database load and improving response times.

2.2 Database Storage: PostgreSQL and NoSQL

According to the project requirements, we implemented both a relational and a NoSQL database backend, allowing us to evaluate their performance differences within the same application. We configured a system variable in our backend that allowed us to dynamically switch between using Azure CosmosDB for PostgreSQL and Azure CosmosDB NoSQL, as well as enabling or disabling Redis caching. This approach allowed us to perform a comparative tests across different database and caching configurations without altering the core logic of the application.

- **Azure CosmosDB for PostgreSQL**: This relational database backend was used to test scenarios where strong relational integrity and transactional support are essential.
- **Azure CosmosDB NoSQL**: The NoSQL backend allowed us to test scenarios optimized for high-speed data access without the overhead of relational constraints, suitable for operations with high insertion and retrieval demands.

2.3 Media Storage: Azure Blob Storage

Short videos uploaded by users were stored in Azure Blob Storage. Each short metadata entry contains a reference (URL) to its respective blob stored in Azure, facilitating scalable media storage without overloading our primary database.

2.4 Database Schema and SQL Table Creation

For the relational PostgreSQL database, we created the following tables: `users`, `shorts`, `follows`, and `likes`. These tables were created using SQL commands in the PostgreSQL console in the Azure portal, as shown below:

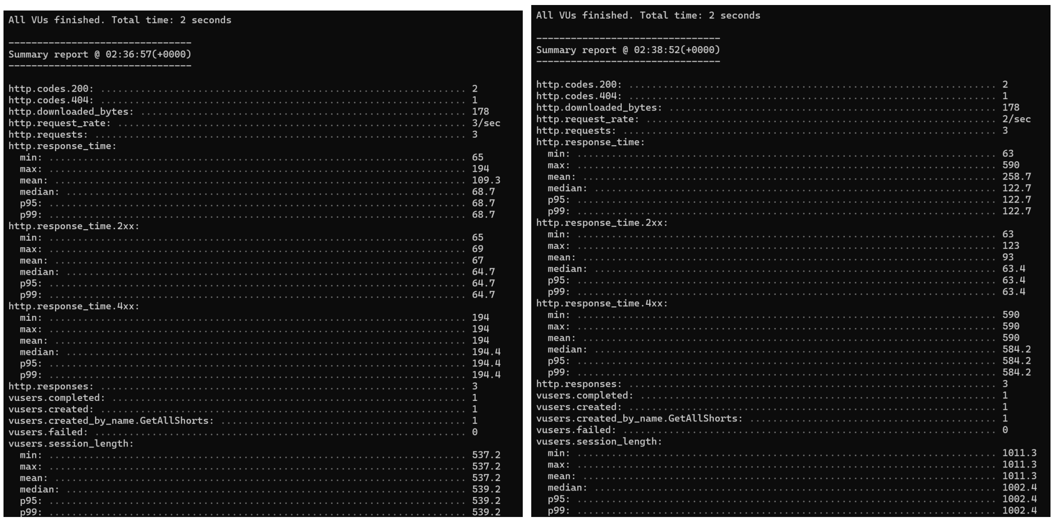
Listing 1: PostgreSQL Table Creation for TuKano

```sql
CREATE TABLE users (
    id VARCHAR(50) PRIMARY KEY,
    userId VARCHAR(50) UNIQUE NOT NULL,
    displayName VARCHAR(100),
    email VARCHAR(100),
    pwd VARCHAR(100)
);

CREATE TABLE shorts (
    id VARCHAR(50) PRIMARY KEY,
    shortId VARCHAR(50) UNIQUE NOT NULL,
    ownerId VARCHAR(50) NOT NULL REFERENCES users(id) ON DELETE CASCADE,
    blobUrl TEXT NOT NULL,
    timestamp BIGINT NOT NULL,
    totalLikes INTEGER DEFAULT 0
);

CREATE TABLE IF NOT EXISTS follows (
    follower VARCHAR(50) NOT NULL,
    followee VARCHAR(50) NOT NULL,
    id VARCHAR(50) NOT NULL,
    PRIMARY KEY (follower, followee),
    FOREIGN KEY (follower) REFERENCES users(id) ON DELETE CASCADE,
    FOREIGN KEY (followee) REFERENCES users(id) ON DELETE CASCADE
);

CREATE INDEX IF NOT EXISTS idx_follows_follower ON follows(follower);
CREATE INDEX IF NOT EXISTS idx_follows_followee ON follows(followee);

CREATE TABLE likes (
    id VARCHAR(50) PRIMARY KEY,
    userId VARCHAR(50) NOT NULL,
    shortId VARCHAR(50) NOT NULL,
    ownerId VARCHAR(50) NOT NULL,
    FOREIGN KEY (userId) REFERENCES users(id) ON DELETE CASCADE,
    FOREIGN KEY (shortId) REFERENCES shorts(id) ON DELETE CASCADE,
    FOREIGN KEY (ownerId) REFERENCES users(id) ON DELETE CASCADE
);

CREATE INDEX IF NOT EXISTS idx_likes_userId ON likes(userId);
CREATE INDEX IF NOT EXISTS idx_likes_shortId ON likes(shortId);
CREATE INDEX IF NOT EXISTS idx_likes_ownerId ON likes(ownerId);
```

3 Performance Analysis

3.1 Impact of Caching with Redis

The use of Redis cache significantly improved the performance of read-heavy operations by caching frequently accessed data and serving it directly from memory. This resulted in a substantial reduction in response times, particularly for retrieval operations. Figure 1 illustrates the effect of caching on response time for the **Get All Shorts** operation on PostgreSQL, comparing results with and without Redis cache.



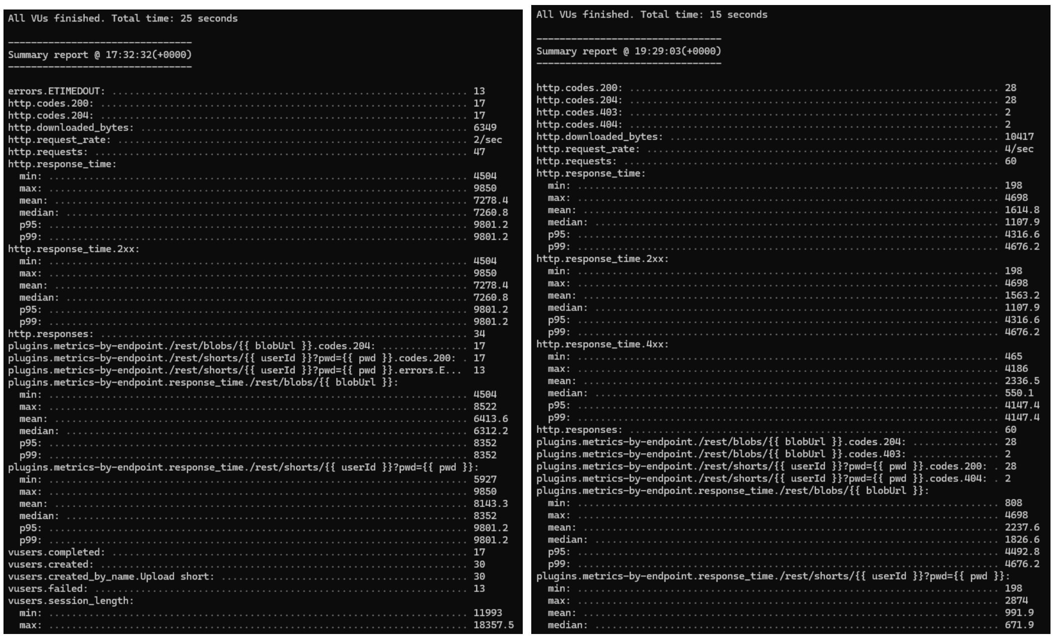
```
All VUs finished. Total time: 2 seconds

--------------------------------
Summary report @ 02:36:57(+0000)
--------------------------------

http.codes.200: ............................................................... 2
http.codes.404: ............................................................... 1
http.downloaded_bytes: ........................................................ 178
http.request_rate: ............................................................ 3/sec
http.requests: ................................................................ 3
http.response_time:
  min: ........................................................................ 65
  max: ........................................................................ 194
  mean: ....................................................................... 109.3
  median: ..................................................................... 68.7
  p95: ........................................................................ 68.7
  p99: ........................................................................ 68.7
http.response_time.2xx:
  min: ........................................................................ 65
  max: ........................................................................ 69
  mean: ....................................................................... 67
  median: ..................................................................... 64.7
  p95: ........................................................................ 64.7
  p99: ........................................................................ 64.7
http.response_time.4xx:
  min: ........................................................................ 194
  max: ........................................................................ 194
  mean: ....................................................................... 194
  median: ..................................................................... 194.4
  p95: ........................................................................ 194.4
  p99: ........................................................................ 194.4
http.responses: ............................................................... 3
vusers.completed: ............................................................. 1
vusers.created: ............................................................... 1
vusers.created_by_name.GetAllShorts: .......................................... 1
vusers.failed: ................................................................ 0
vusers.session_length:
  min: ........................................................................ 537.2
  max: ........................................................................ 537.2
  mean: ....................................................................... 537.2
  median: ..................................................................... 539.2
  p95: ........................................................................ 539.2
  p99: ........................................................................ 539.2
```

```
All VUs finished. Total time: 2 seconds

--------------------------------
Summary report @ 02:38:52(+0000)
--------------------------------

http.codes.200: ............................................................... 2
http.codes.404: ............................................................... 1
http.downloaded_bytes: ........................................................ 178
http.request_rate: ............................................................ 2/sec
http.requests: ................................................................ 3
http.response_time:
  min: ........................................................................ 63
  max: ........................................................................ 590
  mean: ....................................................................... 258.7
  median: ..................................................................... 122.7
  p95: ........................................................................ 122.7
  p99: ........................................................................ 122.7
http.response_time.2xx:
  min: ........................................................................ 63
  max: ........................................................................ 123
  mean: ....................................................................... 93
  median: ..................................................................... 63.4
  p95: ........................................................................ 63.4
  p99: ........................................................................ 63.4
http.response_time.4xx:
  min: ........................................................................ 590
  max: ........................................................................ 590
  mean: ....................................................................... 590
  median: ..................................................................... 584.2
  p95: ........................................................................ 584.2
  p99: ........................................................................ 584.2
http.responses: ............................................................... 3
vusers.completed: ............................................................. 1
vusers.created: ............................................................... 1
vusers.created_by_name.GetAllShorts: .......................................... 1
vusers.failed: ................................................................ 0
vusers.session_length:
  min: ........................................................................ 1011.3
  max: ........................................................................ 1011.3
  mean: ....................................................................... 1011.3
  median: ..................................................................... 1002.4
  p95: ........................................................................ 1002.4
  p99: ........................................................................ 1002.4
```

Figure 1: Response Time for Get All Shorts Operation: (Left) with Redis Cache, (Right) without Cache - PostgreSQL Backend

As shown in the left image, enabling Redis cache lowers response times by serving data directly from memory, avoiding repeated access to PostgreSQL. In the right image, where caching is disabled, response times are noticeably higher due to direct database calls for every request.

However, when we are creating a short or a user, using cache will make response times higher because we have to create these objects both in the cache and in the database. Figure 2 illustrates the effect of caching on response time for the **upload_shorts** operation on NoSQL, comparing results with and without Redis cache.



```
All VUs finished. Total time: 25 seconds

--------------------------------
Summary report @ 17:32:32(+0000)
--------------------------------

errors.ETIMEDOUT: .......................................................... 13
http.codes.200: ............................................................ 17
http.codes.204: ............................................................ 17
http.downloaded_bytes: ..................................................... 6349
http.request_rate: ......................................................... 2/sec
http.requests: ............................................................. 47
http.response_time:
  min: ..................................................................... 4504
  max: ..................................................................... 9850
  mean: .................................................................... 7278.4
  median: .................................................................. 7260.8
  p95: ..................................................................... 9801.2
  p99: ..................................................................... 9801.2
http.response_time.2xx:
  min: ..................................................................... 4504
  max: ..................................................................... 9850
  mean: .................................................................... 7278.4
  median: .................................................................. 7260.8
  p95: ..................................................................... 9801.2
  p99: ..................................................................... 9801.2
http.responses: ............................................................ 34
plugins.metrics-by-endpoint./rest/blobs/{{ blobUrl }}.codes.204: ........... 17
plugins.metrics-by-endpoint./rest/shorts/{{ userId }}?pwd={{ pwd }}.codes.200: . 17
plugins.metrics-by-endpoint./rest/shorts/{{ userId }}?pwd={{ pwd }}.errors.E...  13
plugins.metrics-by-endpoint.response_time./rest/blobs/{{ blobUrl }}:
  min: ..................................................................... 4504
  max: ..................................................................... 8522
  mean: .................................................................... 6413.6
  median: .................................................................. 6312.2
  p95: ..................................................................... 8352
  p99: ..................................................................... 8352
plugins.metrics-by-endpoint.response_time./rest/shorts/{{ userId }}?pwd={{ pwd }}:
  min: ..................................................................... 5927
  max: ..................................................................... 9850
  mean: .................................................................... 8143.3
  median: .................................................................. 8352
  p95: ..................................................................... 9801.2
  p99: ..................................................................... 9801.2
vusers.completed: .......................................................... 17
vusers.created: ............................................................ 30
vusers.created_by_name.Upload short: ....................................... 30
vusers.failed: ............................................................. 13
vusers.session_length:
  min: ..................................................................... 11993
  max: ..................................................................... 18357.5
```

```
All VUs finished. Total time: 15 seconds

--------------------------------
Summary report @ 19:29:03(+0000)
--------------------------------

http.codes.200: ............................................................ 28
http.codes.204: ............................................................ 28
http.codes.403: ............................................................ 2
http.codes.404: ............................................................ 2
http.downloaded_bytes: ..................................................... 10417
http.request_rate: ......................................................... 4/sec
http.requests: ............................................................. 60
http.response_time:
  min: ..................................................................... 198
  max: ..................................................................... 4698
  mean: .................................................................... 1614.8
  median: .................................................................. 1107.9
  p95: ..................................................................... 4316.6
  p99: ..................................................................... 4676.2
http.response_time.2xx:
  min: ..................................................................... 198
  max: ..................................................................... 4698
  mean: .................................................................... 1563.2
  median: .................................................................. 1107.9
  p95: ..................................................................... 4316.6
  p99: ..................................................................... 4676.2
http.response_time.4xx:
  min: ..................................................................... 465
  max: ..................................................................... 4186
  mean: .................................................................... 2336.5
  median: .................................................................. 550.1
  p95: ..................................................................... 4147.4
  p99: ..................................................................... 4147.4
http.responses: ............................................................ 60
plugins.metrics-by-endpoint./rest/blobs/{{ blobUrl }}.codes.204: ........... 28
plugins.metrics-by-endpoint./rest/blobs/{{ blobUrl }}.codes.403: ........... 2
plugins.metrics-by-endpoint./rest/shorts/{{ userId }}?pwd={{ pwd }}.codes.200: . 28
plugins.metrics-by-endpoint./rest/shorts/{{ userId }}?pwd={{ pwd }}.codes.404: . 2
plugins.metrics-by-endpoint.response_time./rest/blobs/{{ blobUrl }}:
  min: ..................................................................... 808
  max: ..................................................................... 4698
  mean: .................................................................... 2237.6
  median: .................................................................. 1826.6
  p95: ..................................................................... 4492.8
  p99: ..................................................................... 4676.2
plugins.metrics-by-endpoint.response_time./rest/shorts/{{ userId }}?pwd={{ pwd }}:
  min: ..................................................................... 198
  max: ..................................................................... 2874
  mean: .................................................................... 991.9
  median: .................................................................. 671.9
```

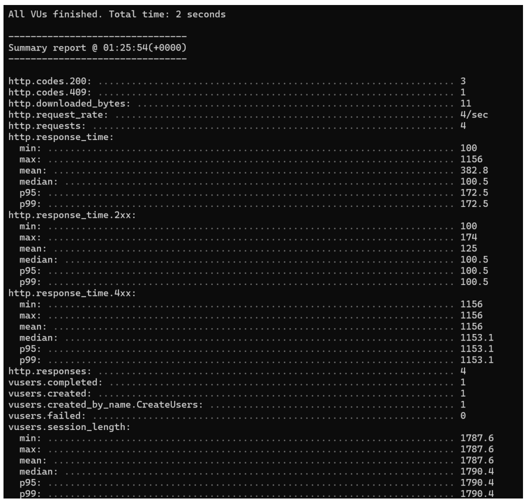
Figure 2: Response Time for upload_shorts Operation: (Left) with Redis Cache, (Right) without Cache - NoSQL Backend

3.2 Comparison of PostgreSQL vs. NoSQL Backends

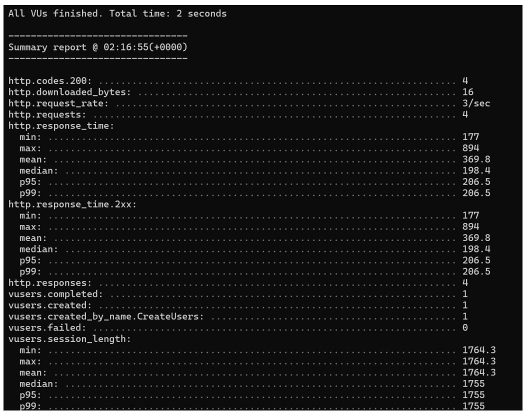
To compare the differences in performance between the PostgreSQL and NoSQL backends, we conducted tests using the **CreateUsers** and **LikeShorts** scenarios. The **CreateUsers** test measures performance during user creation, while **LikeShorts** involves high-frequency write and retrieval operations for the "like" functionality across different users and shorts creation.

3.2.1 CreateUsers and user_register Tests

Figure 3 presents the response times for the **CreateUsers** operation, comparing NoSQL and PostgreSQL backends without caching.



```
All VUs finished. Total time: 2 seconds

--------------------------------
Summary report @ 01:25:54(+0000)
--------------------------------

http.codes.200: ........................................................... 3
http.codes.409: ........................................................... 1
http.downloaded_bytes: .................................................... 11
http.request_rate: ........................................................ 4/sec
http.requests: ............................................................ 4
http.response_time:
  min: .................................................................... 100
  max: .................................................................... 1156
  mean: ................................................................... 382.8
  median: ................................................................. 100.5
  p95: .................................................................... 172.5
  p99: .................................................................... 172.5
http.response_time.2xx:
  min: .................................................................... 100
  max: .................................................................... 174
  mean: ................................................................... 125
  median: ................................................................. 100.5
  p95: .................................................................... 100.5
  p99: .................................................................... 100.5
http.response_time.4xx:
  min: .................................................................... 1156
  max: .................................................................... 1156
  mean: ................................................................... 1156
  median: ................................................................. 1153.1
  p95: .................................................................... 1153.1
  p99: .................................................................... 1153.1
http.responses: ........................................................... 4
vusers.completed: ......................................................... 1
vusers.created: ........................................................... 1
vusers.created_by_name.CreateUsers: ....................................... 1
vusers.failed: ............................................................ 0
vusers.session_length:
  min: .................................................................... 1787.6
  max: .................................................................... 1787.6
  mean: ................................................................... 1787.6
  median: ................................................................. 1790.4
  p95: .................................................................... 1790.4
  p99: .................................................................... 1790.4
```

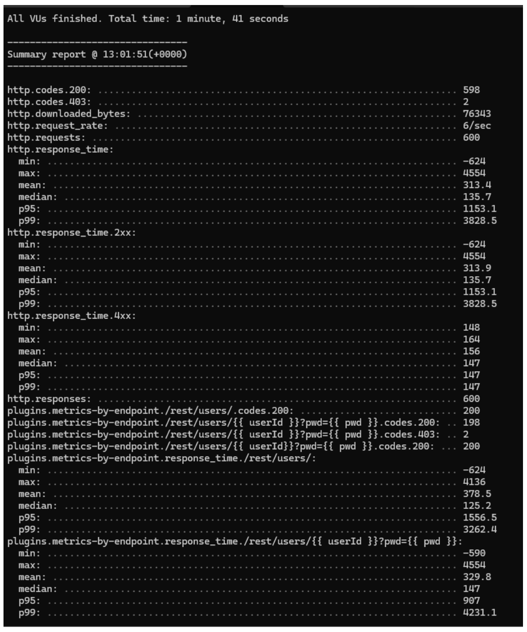


```
All VUs finished. Total time: 2 seconds

--------------------------------
Summary report @ 02:16:55(+0000)
--------------------------------

http.codes.200: ........................................................... 4
http.downloaded_bytes: .................................................... 16
http.request_rate: ........................................................ 3/sec
http.requests: ............................................................ 4
http.response_time:
  min: .................................................................... 177
  max: .................................................................... 894
  mean: ................................................................... 369.8
  median: ................................................................. 198.4
  p95: .................................................................... 206.5
  p99: .................................................................... 206.5
http.response_time.2xx:
  min: .................................................................... 177
  max: .................................................................... 894
  mean: ................................................................... 369.8
  median: ................................................................. 198.4
  p95: .................................................................... 206.5
  p99: .................................................................... 206.5
http.responses: ........................................................... 4
vusers.completed: ......................................................... 1
vusers.created: ........................................................... 1
vusers.created_by_name.CreateUsers: ....................................... 1
vusers.failed: ............................................................ 0
vusers.session_length:
  min: .................................................................... 1764.3
  max: .................................................................... 1764.3
  mean: ................................................................... 1764.3
  median: ................................................................. 1755
  p95: .................................................................... 1755
  p99: .................................................................... 1755
```

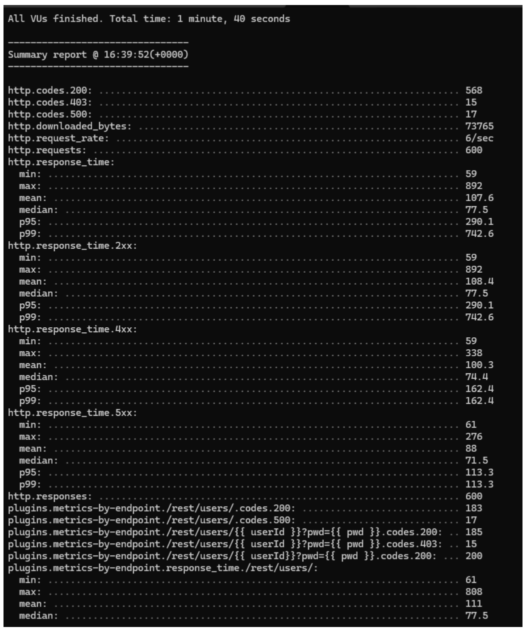
Figure 3: Response Time for Create Users Operation: (Left) NoSQL Backend (No Cache), (Right) PostgreSQL Backend (No Cache)

In Figure 3, both NoSQL and PostgreSQL achieved similar response times for user creation. Although NoSQL is typically faster, PostgreSQL offered competitive response times.

Figure 4 presents the response times for the **user_register** operation, comparing NoSQL and PostgreSQL backends with caching.



```
All VUs finished. Total time: 1 minute, 41 seconds

--------------------------------
Summary report @ 13:01:51(+0000)
--------------------------------

http.codes.200: ........................................................... 598
http.codes.403: ........................................................... 2
http.downloaded_bytes: .................................................... 76343
http.request_rate: ........................................................ 6/sec
http.requests: ............................................................ 600
http.response_time:
  min: .................................................................... -624
  max: .................................................................... 4554
  mean: ................................................................... 313.4
  median: ................................................................. 135.7
  p95: .................................................................... 1153.1
  p99: .................................................................... 3828.5
http.response_time.2xx:
  min: .................................................................... -624
  max: .................................................................... 4554
  mean: ................................................................... 313.9
  median: ................................................................. 135.7
  p95: .................................................................... 1153.1
  p99: .................................................................... 3828.5
http.response_time.4xx:
  min: .................................................................... 148
  max: .................................................................... 164
  mean: ................................................................... 156
  median: ................................................................. 147
  p95: .................................................................... 147
  p99: .................................................................... 147
http.responses: ........................................................... 600
plugins.metrics-by-endpoint./rest/users/.codes.200: ....................... 200
plugins.metrics-by-endpoint./rest/users/{{ userId }}?pwd={{ pwd }}.codes.200: .. 198
plugins.metrics-by-endpoint./rest/users/{{ userId }}?pwd={{ pwd }}.codes.403: .. 2
plugins.metrics-by-endpoint./rest/users/{{ userId}}?pwd={{ pwd }}.codes.200: ... 200
plugins.metrics-by-endpoint.response_time./rest/users/:
  min: .................................................................... -624
  max: .................................................................... 4136
  mean: ................................................................... 378.5
  median: ................................................................. 125.2
  p95: .................................................................... 1556.5
  p99: .................................................................... 3262.4
plugins.metrics-by-endpoint.response_time./rest/users/{{ userId }}?pwd={{ pwd }}:
  min: .................................................................... -590
  max: .................................................................... 4554
  mean: ................................................................... 329.8
  median: ................................................................. 147
  p95: .................................................................... 907
  p99: .................................................................... 4231.1
```



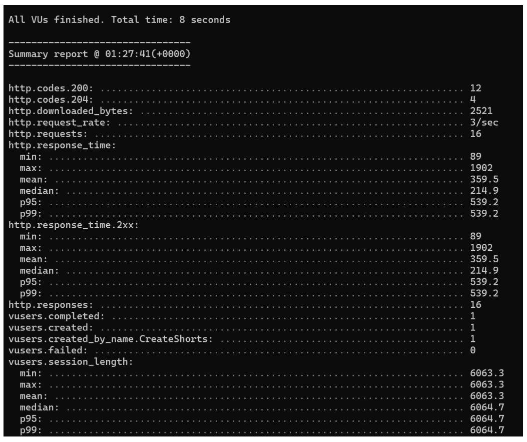
```
All VUs finished. Total time: 1 minute, 40 seconds

--------------------------------
Summary report @ 16:39:52(+0000)
--------------------------------

http.codes.200: ........................................................... 568
http.codes.403: ........................................................... 15
http.codes.500: ........................................................... 17
http.downloaded_bytes: .................................................... 73765
http.request_rate: ........................................................ 6/sec
http.requests: ............................................................ 600
http.response_time:
  min: .................................................................... 59
  max: .................................................................... 892
  mean: ................................................................... 107.6
  median: ................................................................. 77.5
  p95: .................................................................... 290.1
  p99: .................................................................... 742.6
http.response_time.2xx:
  min: .................................................................... 59
  max: .................................................................... 892
  mean: ................................................................... 108.4
  median: ................................................................. 77.5
  p95: .................................................................... 290.1
  p99: .................................................................... 742.6
http.response_time.4xx:
  min: .................................................................... 59
  max: .................................................................... 338
  mean: ................................................................... 100.3
  median: ................................................................. 74.4
  p95: .................................................................... 162.4
  p99: .................................................................... 162.4
http.response_time.5xx:
  min: .................................................................... 61
  max: .................................................................... 276
  mean: ................................................................... 88
  median: ................................................................. 71.5
  p95: .................................................................... 113.3
  p99: .................................................................... 113.3
http.responses: ........................................................... 600
plugins.metrics-by-endpoint./rest/users/.codes.200: ....................... 183
plugins.metrics-by-endpoint./rest/users/.codes.500: ....................... 17
plugins.metrics-by-endpoint./rest/users/{{ userId }}?pwd={{ pwd }}.codes.200: .. 185
plugins.metrics-by-endpoint./rest/users/{{ userId }}?pwd={{ pwd }}.codes.403: .. 15
plugins.metrics-by-endpoint./rest/users/{{ userId}}?pwd={{ pwd }}.codes.200: ... 200
plugins.metrics-by-endpoint.response_time./rest/users/:
  min: .................................................................... 61
  max: .................................................................... 808
  mean: ................................................................... 111
  median: ................................................................. 77.5
```

Figure 4: Response Time for user_register Operation: (Left) NoSQL Backend (With Cache), (Right) PostgreSQL Backend (With Cache)

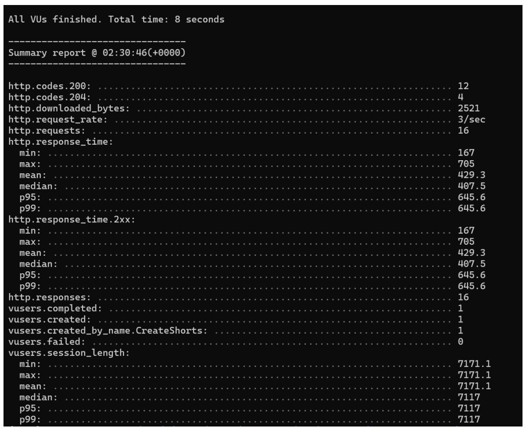
In Figure 4, although NoSQL is typically faster, PostgreSQL ended up being 1 second faster for the **user_register** operation, this might have happened because in NoSQL (Left) we have more successful operations: 598 http codes 200 vs 568 http codes 200 on PostgreSQL (Right).

3.2.2 LikeShorts Test

For a heavier load scenario, we ran the **LikeShorts** test, which involved creating shorts and performing numerous ”like” operations across multiple users. Figure 5 shows the response times for this test on NoSQL and PostgreSQL without caching.



```
All VUs finished. Total time: 8 seconds

--------------------------------
Summary report @ 01:27:41(+0000)
--------------------------------

http.codes.200: ................................................................ 12
http.codes.204: ................................................................ 4
http.downloaded_bytes: ......................................................... 2521
http.request_rate: ............................................................. 3/sec
http.requests: ................................................................. 16
http.response_time:
  min: ......................................................................... 89
  max: ......................................................................... 1902
  mean: ........................................................................ 359.5
  median: ...................................................................... 214.9
  p95: ......................................................................... 539.2
  p99: ......................................................................... 539.2
http.response_time.2xx:
  min: ......................................................................... 89
  max: ......................................................................... 1902
  mean: ........................................................................ 359.5
  median: ...................................................................... 214.9
  p95: ......................................................................... 539.2
  p99: ......................................................................... 539.2
http.responses: ................................................................ 16
vusers.completed: .............................................................. 1
vusers.created: ................................................................ 1
vusers.created_by_name.CreateShorts: ........................................... 1
vusers.failed: ................................................................. 0
vusers.session_length:
  min: ......................................................................... 6063.3
  max: ......................................................................... 6063.3
  mean: ........................................................................ 6063.3
  median: ...................................................................... 6064.7
  p95: ......................................................................... 6064.7
  p99: ......................................................................... 6064.7
```



```
All VUs finished. Total time: 8 seconds

--------------------------------
Summary report @ 02:30:46(+0000)
--------------------------------

http.codes.200: ................................................................ 12
http.codes.204: ................................................................ 4
http.downloaded_bytes: ......................................................... 2521
http.request_rate: ............................................................. 3/sec
http.requests: ................................................................. 16
http.response_time:
  min: ......................................................................... 167
  max: ......................................................................... 705
  mean: ........................................................................ 429.3
  median: ...................................................................... 407.5
  p95: ......................................................................... 645.6
  p99: ......................................................................... 645.6
http.response_time.2xx:
  min: ......................................................................... 167
  max: ......................................................................... 705
  mean: ........................................................................ 429.3
  median: ...................................................................... 407.5
  p95: ......................................................................... 645.6
  p99: ......................................................................... 645.6
http.responses: ................................................................ 16
vusers.completed: .............................................................. 1
vusers.created: ................................................................ 1
vusers.created_by_name.CreateShorts: ........................................... 1
vusers.failed: ................................................................. 0
vusers.session_length:
  min: ......................................................................... 7171.1
  max: ......................................................................... 7171.1
  mean: ........................................................................ 7171.1
  median: ...................................................................... 7117
  p95: ......................................................................... 7117
  p99: ......................................................................... 7117
```

Figure 5: Response Time for Like Shorts Operation: (Left) NoSQL Backend (No Cache), (Right) PostgreSQL Backend (No Cache)

In Figure 5, we observed a noticeable performance difference, with NoSQL achieving significantly better results. The NoSQL backend completed the **LikeShorts** test with an average response time of approximately 6 seconds, while PostgreSQL took around 7.1 seconds. This advantage for NoSQL can be attributed to its schema flexibility and reduced overhead in write-heavy operations, which allows it to handle frequent data changes more efficiently. PostgreSQL, while providing strong data consistency, obtained slightly higher latency results due to its transactional nature, making it considerably slower in this high-frequency write scenario.

4 Results and Conclusion

Figures 1, 3, 4 and 5 highlight that Redis caching effectively reduces response times for both backends in read-heavy scenarios, particularly with PostgreSQL. For write operations, the performance difference between NoSQL and PostgreSQL was generally minor, with NoSQL demonstrating a slight advantage in high-frequency operations. Specifically, in the **LikeShorts** test, NoSQL completed the operation approximately 1.1 seconds faster than PostgreSQL, showcasing its efficiency in handling write-heavy tasks with minimal overhead.

Also, for write-heavy tasks like in Figure 2, where it shows the **upload_shorts** operation, Redis caching is usually slower than no caching.