the URL for your git repo.

<https://github.com/jerryyummy/distributed_system_2(for> client)

https://github.com/jerryyummy/assignment2\_server

A short description of your data model (5 points) - Please state size of image used if not using the stock image, and also Database/File storage solution.

Size of image:12kb

The object in database includes a primary key named album\_id, blob imagedata, string artist, string title, string year. When I try to save my file to database, I use the code below:

preparedStatement.setBlob(1, new FileInputStream(file));, which can convert imagefile to blob

use this statement to create table:

CREATE TABLE album (  
 album\_id INT AUTO\_INCREMENT PRIMARY KEY,  
 image\_data LONGBLOB,  
 artist varchar(255),  
 year varchar(255),  
 title varchar(255)  
);

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

Output windows for the 3 client configuration tests run against a single server/DB (5 points)

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | time | throughput | mean | 99% time | Succeed request | Fail  request |
| 10,10,2 | 883605 | 226 | 393 ms | 88 ms | 92506 | 7494 |
| 10,20,2 | 1027410 | 389 | 306 ms | 75 ms | 100170 | 99830 |
| 10,30,2 | 1147122 | 523 | 328 ms | 93 ms | 193473 | 106527 |

For post

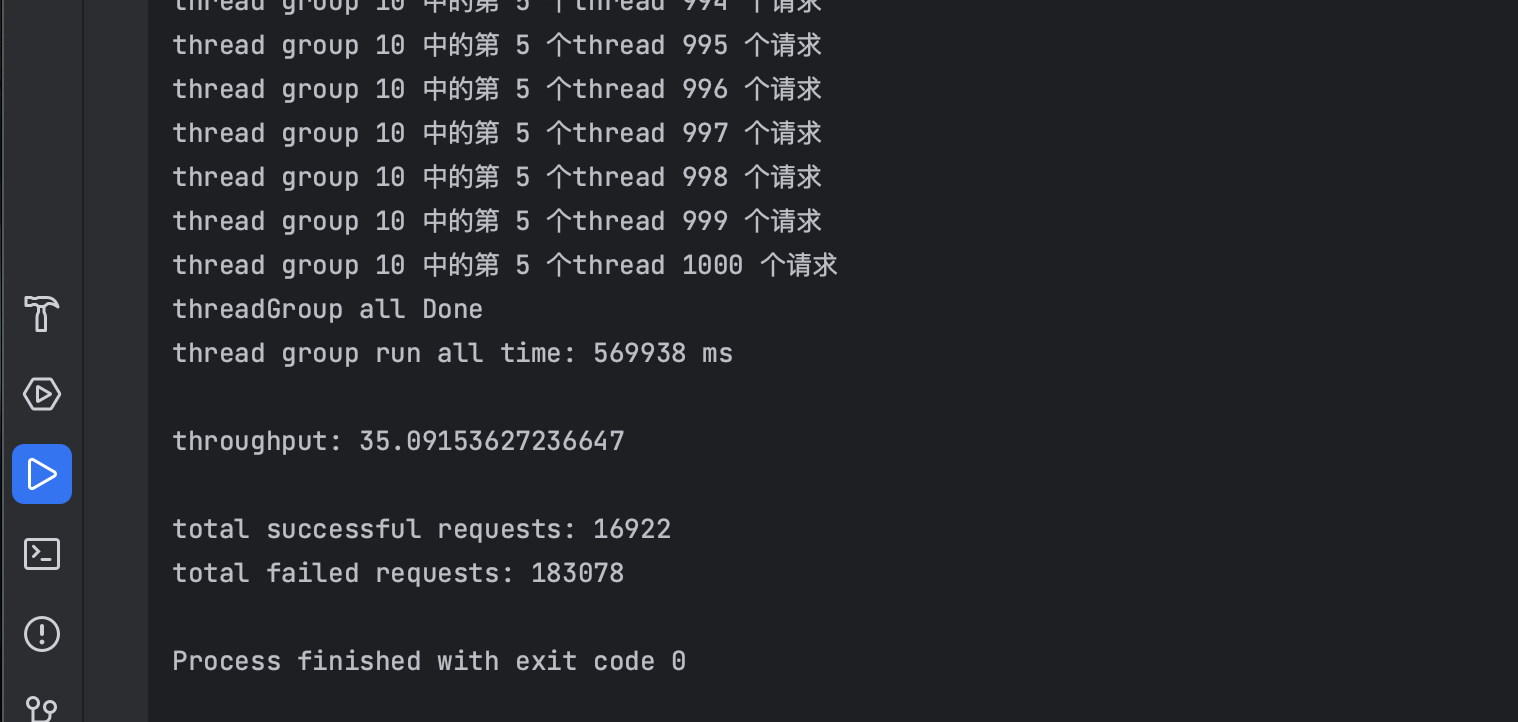
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | time | throughput | mean | 99% time | Succeed request | Fail  request |
| 10,10,2 | 883605 | 226 | 90 ms | 78 ms | 92739 | 7261 |
| 10,20,2 | 1027410 | 389 | 160 ms | 69 ms | 100117 | 99830 |
| 10,30,2 | 1147212 | 523 | 148 ms | 82 ms | 193473 | 106529 |

For get  
a plot comparing the throughout

Output windows for the 3 client configuration tests run against a two load balanced servers/DB (15 points)

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Mean time** | **Median time** | **99% time** | **Min time** | **Max time** | **Wall time** | **throughput** | **Succeed request** |
| **10 10 2** | **614** | **533** | **146** | **125** | **17592** | **1335427** | **150** | **9118** |
| **10 20 2** | **520** | **452** | **137** | **116** | **4814** | **418706** | **955** | **16264** |
| **10 30 2** | **621** | **517** | **138** | **122** | **7686** | **1132342** | **530** | **27297** |

For post

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Mean time** | **Median time** | **99% time** | **Min time** | **Max time** | **Wall time** | **throughput** | **Succeed request** |
| **10 10 2** | **304** | **171** | **64** | **52** | **6295** | **1335427** | **150** | **6835** |
| **10 20 2** | **181** | **151** | **59** | **42** | **4764** | **418706** | **955** | **14332** |
| **10 30 2** | **172** | **147** | **61** | **46** | **5594** | **1132342** | **530** | **21582** |

For get

Output window for optimized server configuration for client with 30 Thread Groups. Briefly describe what configuration changes you made and what % throughput improvement you achieved (15 points)

I made two replicate for rds so that we can separate writing and reading , I also change the number of max connections in database, and I add one more instance in the load balancer for testing.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Wall time is 1038116ms, throughput is 578 per second, the succeed request is 166768, fail request is 433232 the total throughput improve about 9%, the performance improve a bit because we still use one database to write, and the max\_connection in RDS is limited to 100.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Mean response | Wall time | 99% time | Min response | Max response | throughput | succeed |
| Two instance | 621 | **1132342** | 138 | 122 | 7686 | 530 | 272970 |
| optimal | 489 | 960989 | 133 | 120 | 4348 | 578 | 270960 |

For post

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Mean response | Wall time | 99% time | Min response | Max response | throughput | succeed |
| Two instance | 172 | **1132342** | 147 | 46 | 5594 | 530 | 215820 |
| optimal | 167 | 960989 | 78 | 57 | 4373 | 578 | 349150 |

For get

Other optional but highly recommended to have in your submission:

A screenshot of a computer

Description automatically generated

What database looks like

A graph showing a line

Description automatically generated with medium confidence

A graph showing a line

Description automatically generated

Before optimalization

A screenshot of a computer

Description automatically generated

After optimalization

As we can see , we can decrease the cpu utilization from about 40% to 20%, and the maximum connections in every is about 60, which is less than 70.

A screenshot of a computer

Description automatically generated

Set up load balancer and target group

A screenshot of a graph

Description automatically generated

Target group after testing

A screenshot of a graph

Description automatically generated

Alb after testing