

Assignment 2 Report Shengguo Zhou

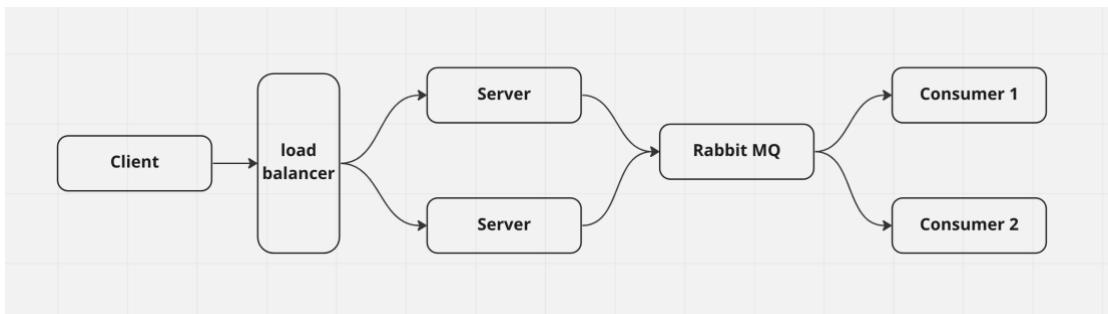
URL:

https://github.com/Shengguo-Zhou/CS6650_Distributed_System/tree/main/assignment-2

Description of design:

In this report, we will document the steps taken to complete Assignment 2, which builds on Assignment 1 by implementing simple business logic in the server and posting the result to a queue for subsequent consumption and processing.

Client would send 500k messages to the server, but the load balancer would deal with it, it will distribute the messages to 2 servers, to make it work in a balance way. And all the servers would deliver the messages to rabbit mq, and two consumers would consume the data inside the rabbitmq.



Client:

Main function is in “MultiThread” class. Inside this class:

First step, I created a queue and put 500k request inside it. And the queue is a global variable.

Nest step, I use a for loop to generate 200 threads to achieve multiply-thread. For each of thread, whenever this thread is started, it will pull a request from the queue. If the queue is empty, it will pull nothing, which means this thread would do nothing. So which thread is started first, which one may be tasked with more sending request tasks. Therefore, within this step, there would be 500k requests in total, and the 200 threads need to send it to the local or EC2 server. When all the requests are sent, which means the queue is empty, the 200 threads would then have a rest.

Last step, there would be some other global variable that would count the successful request number and the failed one, the lasting time and so on.

RunInOneThread.java: This class is designed for each one of the thread, I need to send the data through `api.swipeWithHttpInfo`. Inside the run function, the detail of each thread is implemented, queue is the global passed from the main function, when a new thread is created, it will get a request body from the queue. And it won't stop until the queue is empty. There are another two parameters to count the number of successful and failed sending tasks. In the sentEvent function, it will try at

most 5 times to sent one request, if it is successful within 5 times of try, success will plus one. When it is failed, this request will be abandoned and failure will plus one.

SwipeEvent.java inside the event file and BodyList.java and GenerateSingleBody.java is the helper function for main and singe-thread functions. For each of the request, I have packaged the info within the request and named it to a event class. When all the 500k request body is generated, it would be put into the list, to be specific, it is the queue that I have mentioned.

For this counting time part, I added nothing but result data processing class and a csv output file class. Inside the RunInOneThread.java class, when a new request is successfully sent, I will put the success info into another global list to store it. When all the requests are sent, I will get a list of 500k success info. And then I can sort this list by certain criteria. Like I can sort it by the delay time, then I can get the result of minimum, maximum, medium, 99%percentage response time. And like this, if I sort it by its start time, I can plot the chart by its beginning time to know for each of a certain second, how many requests would be sent. But I remove the csv output part from this client, because it won't be needed.

Server:

In general, the server for assignment2 is pretty similar to the one in assignment1, we still need to get messages from client, but we also need to deliver these messages to rabbit mq. And in order to achieve that, in the function “basicPublish”, I published the message to the rabbit mq. And the rabbit mq will build the channel to deal with these messages

To implement the doPost() method in the servlet to fully validate the URL and JSON payload, format the incoming Swipe data, and send it as a payload to a remote queue. I chose to use RabbitMQ as our queue technology, and deployed it on its own EC2 instance. I tested with a relatively small number of messages and cleared out the queues regularly using the RMQ management console.

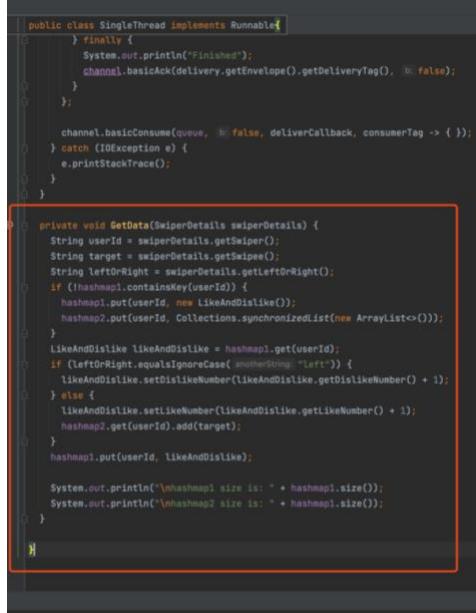
RabbitMQ:

The Rabbitmq is installed in my local PC Mac and AWS Ubuntu. For Mac PC I used homebrew to install it, I have to make sure that homebrew is installed in the correct path, which is inside the opt file, and this part took me a lot of part. And if I don't do that, the calling of the rabbitmq api interface would cause a segement error. To install it in AWS Ubuntu, I created a new instance to install the Rabbitmq, if I want to use the rabbitmq inside the aws ubuntu, I have to know the ip address of is, which would change each time I restart the virtual machine, it will need me to change the .war file of server each time.

RabbitMQ is a message broker that implements the Advanced Message Queuing Protocol (AMQP) to handle messaging between applications. In this assignment, RabbitMQ is used to post the swipe data payload from the server to a remote queue for subsequent consumption and processing by the consumers.

Consumer:

Use two ConcurrentHashMap to record all of the messages



```
public class SingleThread implements Runnable {
    ...
    > finally {
        System.out.println("finished");
        channel.basicAck(delivery.getEnvelope().getDeliveryTag(), false);
    }
}

channel.basicConsume(queue, false, deliverCallback, consumerTag -> { });
} catch (IOException e) {
    e.printStackTrace();
}
}

private void GetData(SwiperDetails swiperDetails) {
    String userId = swiperDetails.getSwiper();
    String target = swiperDetails.getSwipee();
    String leftOrRight = swiperDetails.getLeftOrRight();
    if (!hashmap1.containsKey(userId)) {
        hashmap1.put(userId, new LikeAndDislike());
        hashmap2.put(userId, Collections.synchronizedList(new ArrayList<>()));
    }
    LikeAndDislike likeAndDislike = hashmap1.get(userId);
    if (leftOrRight.equalsIgnoreCase("Left")) {
        likeAndDislike.setDislikeNumber(likeAndDislike.getDislikeNumber() + 1);
    } else {
        likeAndDislike.setLikeNumber(likeAndDislike.getLikeNumber() + 1);
    }
    hashmap2.get(userId).add(target);
    hashmap1.put(userId, likeAndDislike);

    System.out.println("\nhashmap1 size is: " + hashmap1.size());
    System.out.println("\nhashmap2 size is: " + hashmap2.size());
}
}
```

Consumers are separate Java programs that pull messages off the queue and store them in data structures that support the two different access patterns described in the requirements. Each consumer needs to consume every published message, which means that they need to be multithreaded to handle the messages as quickly as they are produced. It is important to make the data structures thread-safe to avoid any concurrency issues.

The consumers should run on separate or a single (not recommended for load testing) instance, connecting remotely to the broker queue. The aim of the consumers is to consume messages as quickly as they are produced to achieve the best throughput on RabbitMQ. The production and consumption rates need to be balanced to keep the queue length as close to zero as possible to ensure the system performs predictably and reliably. The RabbitMQ management console can be used to track the number of messages in the queue, as well as the producers and consumer rates.

The optimal number of consumer threads needed to keep the queue size as close to zero as possible will depend on the specific requirements of the assignment, and can be determined through load testing. It is important to ensure that the queue length doesn't continually grow and shrink, as this can lead to a 'pointy' queue length profile. A plateau in the queue length is fine, as long as it is less than around a 1000. The aim is to balance the production and consumption rates such that they are roughly equal, which will minimize queueing delays and maximize system throughput.

And it would be deployed on AWS by generating a .jar file.

Result:

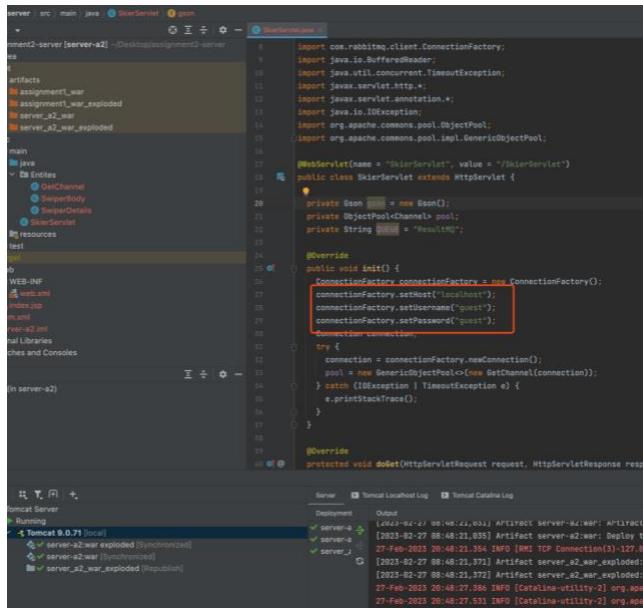
On the Local Side:

Start Local PC Rabbitmq:

```
shengguozhou -- zsh -- 80x24
Last login: Mon Feb 27 18:36:54 on ttys000
shengguozhou@Shengguo-MacBook-Air ~ % brew services start rabbitmq
==> Successfully started `rabbitmq` (label: homebrew.mxcl.rabbitmq)
shengguozhou@Shengguo-MacBook-Air ~ %
```

Server:

Start the local server: Local username: guest, password guest. Queue name: ResultMQ



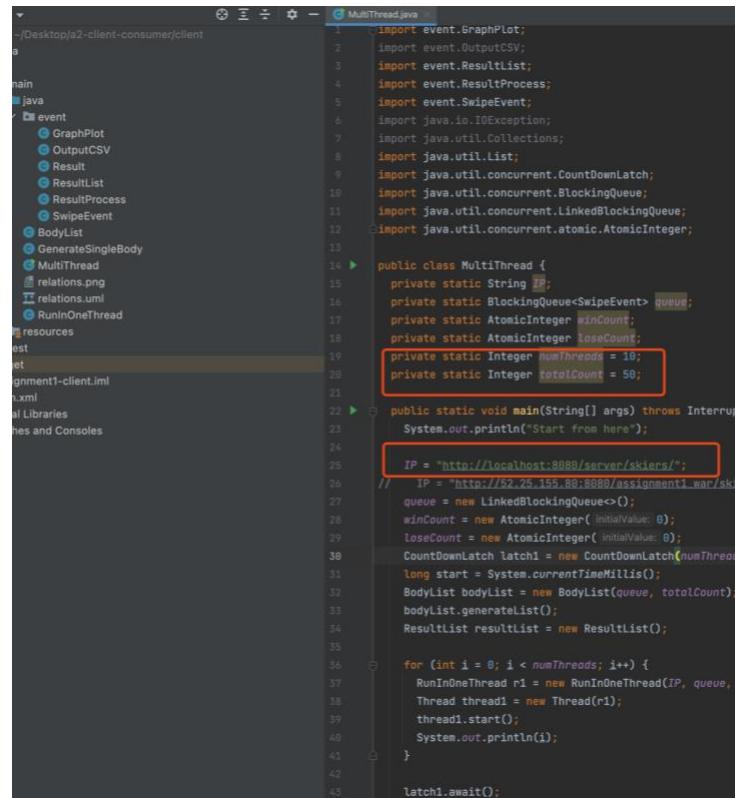
Postman is able to get the message back after sending it

```
POST http://localhost:8080/server/skiers/swipe/left
Params Authorization Headers (8) Body
none form-data x-www-form-urlencoded raw
1
2   "swiper": "123",
3   "swippee": "sdvfda",
4   "comment": "gwdfjbjuvvmk",
5   "leftOrRight": "left"
Body Cookies Headers (5) Test Results
Pretty Raw Preview Visualize JSON
```

```
1 Inside processRequest
2 {
3     "swiper": "123",
4     "swippee": "sdvfda",
5     "comment": "gwdfjbjuvvmk",
6     "leftOrRight": "left"
7 }
8 Assignment 2 doPost is working.
```

Local host 10 threads and 50 messages

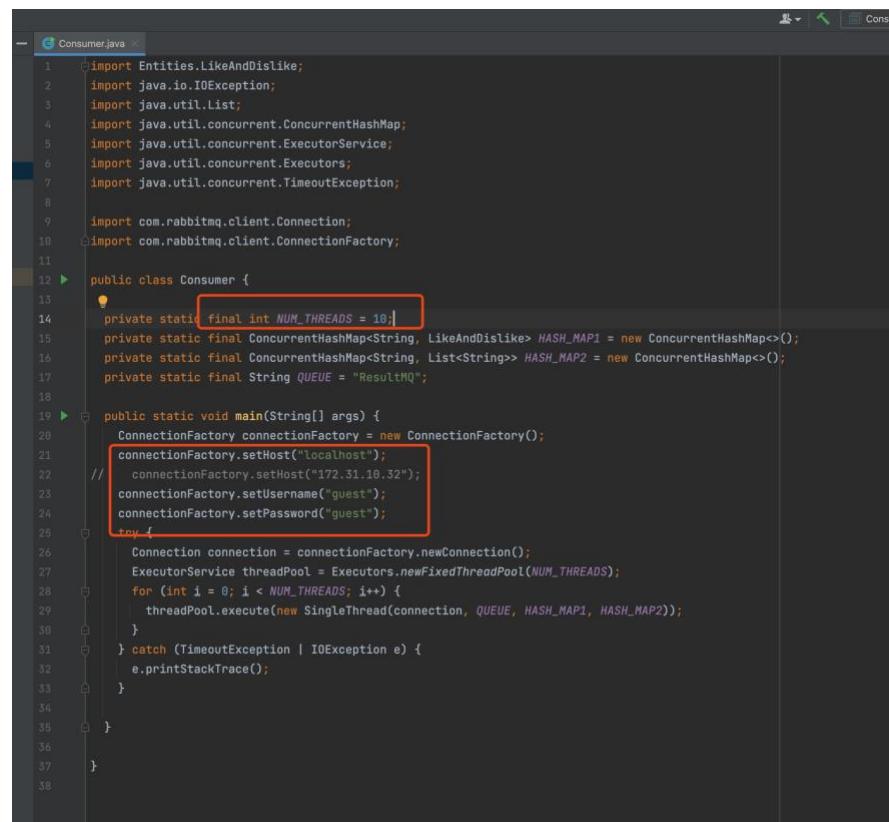
Client:



```
1 import event.GraphPlot;
2 import event.OutputCSV;
3 import event.ResultList;
4 import event.ResultProcess;
5 import event.SwipeEvent;
6 import java.io.IOException;
7 import java.util.Collections;
8 import java.util.List;
9 import java.util.concurrent.CountDownLatch;
10 import java.util.concurrent.BlockingQueue;
11 import java.util.concurrent.LinkedBlockingQueue;
12 import java.util.concurrent.atomic.AtomicInteger;
13
14 public class MultiThread {
15     private static String IP;
16     private static BlockingQueue<SwipeEvent> queue;
17     private static AtomicInteger winCount;
18     private static AtomicInteger loseCount;
19     private static Integer numThreads = 10;
20     private static Integer totalCount = 50;
21
22     public static void main(String[] args) throws InterruptedException {
23         System.out.println("Start from here");
24
25         IP = "http://localhost:8888/server/skiers/";
26         // IP = "http://52.45.155.80:8888/assignment1/war/skiers";
27         queue = new LinkedBlockingQueue<SwipeEvent>();
28         winCount = new AtomicInteger( initialValue: 0 );
29         loseCount = new AtomicInteger( initialValue: 0 );
30         CountdownLatch latch1 = new CountdownLatch(numThreads);
31         long start = System.currentTimeMillis();
32         BodyList bodyList = new BodyList(queue, totalCount);
33         bodyList.generateList();
34         ResultList resultList = new ResultList();
35
36         for (int i = 0; i < numThreads; i++) {
37             RunInOneThread r1 = new RunInOneThread(IP, queue, resultList);
38             Thread thread1 = new Thread(r1);
39             thread1.start();
40             System.out.println(i);
41         }
42
43         latch1.await();
44     }
45 }
```

Consumer: 10 corresponding threads, and to ping local host

Local Rabbitmq username: guest, password: guest



```
1 import Entities.LikeAndDislike;
2 import java.io.IOException;
3 import java.util.List;
4 import java.util.concurrent.ConcurrentHashMap;
5 import java.util.concurrent.ExecutorService;
6 import java.util.concurrent.Executors;
7 import java.util.concurrent.TimeoutException;
8
9 import com.rabbitmq.client.Connection;
10 import com.rabbitmq.client.ConnectionFactory;
11
12 public class Consumer {
13
14     private static final int NUM_THREADS = 10;
15     private static final ConcurrentHashMap<String, LikeAndDislike> HASH_MAP1 = new ConcurrentHashMap<>();
16     private static final ConcurrentHashMap<String, List<String>> HASH_MAP2 = new ConcurrentHashMap<>();
17     private static final String QUEUE = "ResultMQ";
18
19     public static void main(String[] args) {
20         ConnectionFactory connectionFactory = new ConnectionFactory();
21         connectionFactory.setHost("localhost");
22         // connectionFactory.setHost("172.31.10.32");
23         connectionFactory.setUsername("guest");
24         connectionFactory.setPassword("guest");
25         try {
26             Connection connection = connectionFactory.newConnection();
27             ExecutorService threadPool = Executors.newFixedThreadPool(NUM_THREADS);
28             for (int i = 0; i < NUM_THREADS; i++) {
29                 threadPool.execute(new SingleThread(connection, QUEUE, HASH_MAP1, HASH_MAP2));
30             }
31         } catch (TimeoutException | IOException e) {
32             e.printStackTrace();
33         }
34     }
35
36
37 }
```

At first rabbitmq is empty



And Send 50 requests through 10 threads:

```
src / main / java / MultiThread / main
```

```
client ~/Desktop/a2-client-consumer/client
  .idea
  src
  main
    java
      event
        GraphPlot
        OutputCSV
        Result
        ResultList
        ResultProcess
        SwipeEvent
        BodyList
        GenerateSingleBody
        MultiThread
        relations.png
        relations.uml
        RunInOneThread
        resources
    test
  target
```

```
MultiThread.java
```

```
public static void main(String[] args) throws InterruptedException {
    System.out.println("Start from here");
    IP = "http://localhost:8080/server/skiers/";
    // IP = "http://52.25.155.80:8080/assignment1_war/skiers/";
    queue = new LinkedBlockingQueue<>();
    winCount = new AtomicInteger( initialValue: 0 );
    loseCount = new AtomicInteger( initialValue: 0 );
    CountDownLatch latch1 = new CountDownLatch(numThreads);
    long start = System.currentTimeMillis();
    BodyList bodyList = new BodyList(queue, totalCount);
    bodyList.generateList();
    ResultList resultList = new ResultList();

    for (int i = 0; i < numThreads; i++) {
        RunInOneThread r1 = new RunInOneThread(IP, queue, winCount, loseCount, latch1, resultList);
        Thread thread1 = new Thread(r1);
        thread1.start();
        System.out.println(i);
    }
}
```

```
MultiThread
```

```
67
64
68
69
53
35
57
58
59
59
This is the End
Number of successful requests:50
Number of failed requests:0
Total lasting time: 1229
Throughput: 50 requests/second
Mean time: 192.08
Median response time: 68.0
99th percentile response time: 683.0
Max response time: 683.0
Min response time: 15.0

Process finished with exit code 0
```

Rabbitmq get the messages:

localhost:15672/#

RabbitMQ™ RabbitMQ 3.11.9 Erlang 25.2.3

Overview

Totals

Queued messages last minute ?

Ready	50
Unacked	0
Total	50

Message rates last minute ?

Publish	0.00/s
Publisher confirm	0.00/s
Deliver (auto ack)	0.00/s
Consumer ack	0.00/s
Redelivered	0.00/s
Get (manual ack)	0.00/s
Get (auto ack)	0.00/s
Get (empty)	0.00/s
Unroutable (return)	0.0/s
Unroutable (drop)	0.0/s
Disk read	0.0/s
Disk write	0.0/s

Global counts ?

Connections: 1 Channels: 8 Exchanges: 7 Queues: 1 Consumers: 0

Nodes

Name	File descriptors ?	Socket descriptors ?	Erlang processes	Memory ?	Disk space	Uptime	Info	Reset stats	+/-
rabbit@localhost	61 256 available	1 141 available	486 1048576 available	34 MiB 3.2 GiB high watermark 48 MiB low watermark	68 GiB	18m 10s	basic disc 5 rss	This node All nodes	

Churn statistics
Ports and contexts
Export definitions
Import definitions

localhost:15672/#

RabbitMQ™ RabbitMQ 3.11.9 Erlang 25.2.3

Overview

Totals

Queued messages last minute ?

Ready	50
Unacked	0
Total	50

Message rates last minute ?

Publish	0.00/s
Publisher confirm	0.00/s
Deliver (auto ack)	0.00/s
Consumer ack	0.00/s
Redelivered	0.00/s
Get (manual ack)	0.00/s
Get (auto ack)	0.00/s
Get (empty)	0.00/s
Unroutable (return)	0.0/s
Unroutable (drop)	0.0/s
Disk read	0.0/s
Disk write	0.0/s

Global counts ?

Connections: 1 Channels: 8 Exchanges: 7 Queues: 1 Consumers: 0

Nodes

Name	File descriptors ?	Socket descriptors ?	Erlang processes	Memory ?	Disk space	Uptime	Info	Reset stats	+/-
rabbit@localhost	61 256 available	1 141 available	486 1048576 available	33 MiB 3.2 GiB high watermark 48 MiB low watermark	68 GiB	18m 20s	basic disc 5 rss	This node All nodes	

Churn statistics
Ports and contexts
Export definitions
Import definitions

RabbitMQ™ RabbitMQ 3.11.9 Erlang 25.2.3 Refreshed 2

Overview

Totals

Queued messages last minute ?

Ready	40
Unacked	0
Total	40

Message rates last minute ?

Publish	0.00/s
Publisher confirm	0.00/s
Deliver (auto ack)	0.00/s
Consumer ack	2.0/s
Redelivered	0.00/s
Get (manual ack)	4.0/s
Get (auto ack)	0.00/s
Get (empty)	0.00/s
Unroutable (return)	0.00/s
Unroutable (drop)	0.00/s
Disk read	0.00/s
Disk write	0.00/s

Global counts ?

Connections: 2 Channels: 8 Exchanges: 7 Queues: 1 Consumers: 0

Nodes

Name	File descriptors ?	Socket descriptors ?	Erlang processes	Memory ?	Disk space	Uptime	Info	Reset stats
rabbit@localhost	62 236 available	2 141 available	493 1048576 available	36 MiB 3.2 GiB high watermark 8 MiB low watermark	68 GiB	19m 0s	basic disc 5 rss	+/- This node All nodes

Churn statistics

Ports and contexts

Export definitions

Import definitions

HTTP API Server Docs Tutorials Community Support Community Slack Commercial Support Plugins GitHub Changelog

Then Consumer get it and consumes it.

RabbitMQ Management x + localhost:15672/#/ RabbitMQ™ RabbitMQ 3.11.9 Erlang 25.2.3

Overview

Totals

Queued messages last minute ?

Ready	0
Unacked	0
Total	0

Message rates last minute ?

Publish	0.00/s
Publisher confirm	0.00/s
Deliver (auto ack)	0.00/s
Consumer ack	0.00/s
Redelivered	0.00/s
Get (manual ack)	0.00/s
Get (auto ack)	0.00/s
Get (empty)	0.00/s
Unroutable (return)	0.00/s
Unroutable (drop)	0.00/s
Disk read	0.00/s
Disk write	0.00/s

Global counts ?

Connections: 2 Channels: 8 Exchanges: 7 Queues: 1 Consumers: 0

Nodes

Name	File descriptors ?	Socket descriptors ?	Erlang processes	Memory ?	Disk space	Uptime	Info	Reset stats
rabbit@localhost	64 256 available	2 141 available	493 1048576 available	29 MiB 3.2 GiB high watermark 8 MiB low watermark	68 GiB	21m 13s	basic disc 5 rss	+/- This node All nodes

Churn statistics

Ports and contexts

RabbitMQ Management

RabbitMQ 3.11.9 Erlang 25.2.3

Overview

Totals

Queued messages last minute:

Message rates last minute:

Global counts:

- Connections: 2
- Channels: 8
- Exchanges: 7
- Queues: 1
- Consumers: 0

Nodes

Name	File descriptors	Socket descriptors	Erlang processes	Memory	Disk space	Uptime	Info	Reset stats
rabbit@localhost	62 256 available	2 141 available	493 1048576 available	36 MiB 3.2 GiB high watermark 68 GiB low watermark		21m 28s	basic disc 5 rss	+/- This node All nodes

Churn statistics, Ports and contexts, Export definitions, Import definitions

At last, all the messages are consumed.

And we got all the data and stored inside the hashmap

```

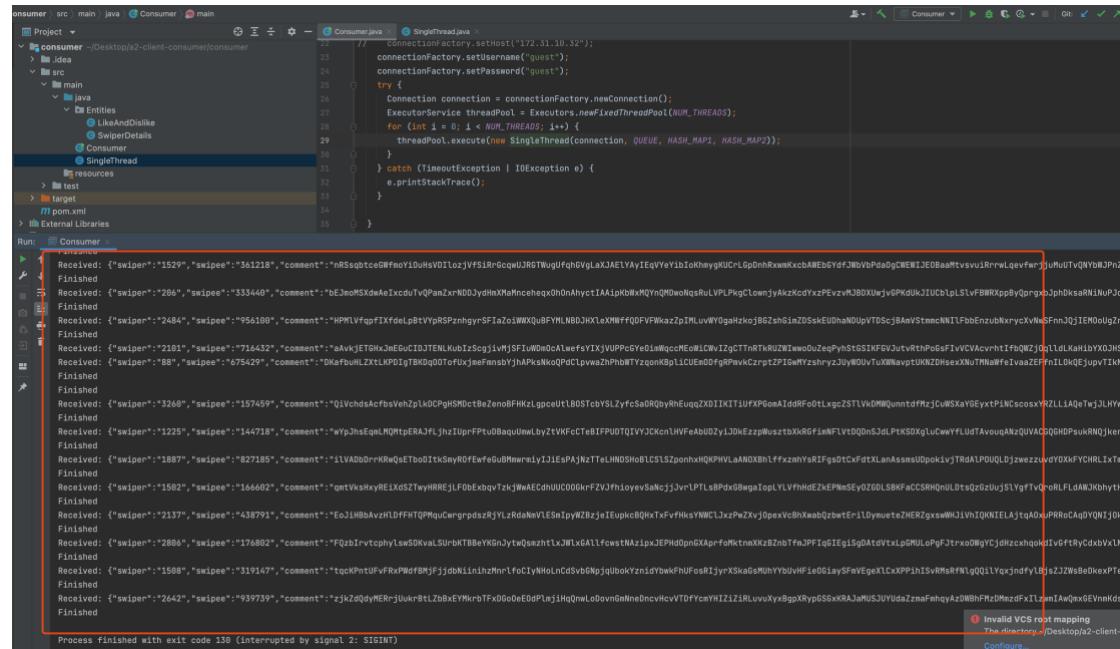
private static final int NUM_THREADS = 30;
private static final ConcurrentHashMap<String, LikeAndDislike> HASH_MAP1 = new ConcurrentHashMap<>();
private static final ConcurrentHashMap<String, List<String>> HASH_MAP2 = new ConcurrentHashMap<>();
private static final String QUEUE = "ResultMQ";
public static void main(String[] args) {
    ConnectionFactory connectionFactory = new ConnectionFactory();
    connectionFactory.setHost("localhost");
    // connectionFactory.setHost("172.31.10.32");
    connectionFactory.setUsername("guest");
    connectionFactory.setPassword("guest");
    try {
        Connection connection = connectionFactory.newConnection();
        ExecutorService threadPool = Executors.newFixedThreadPool(NUM_THREADS);
        for (int i = 0; i < NUM_THREADS; i++) {
            threadPool.execute(new SingleThread(connection, QUEUE, HASH_MAP1, HASH_MAP2));
        }
    } catch (TimeoutException | IOException e) {
        e.printStackTrace();
    }
}

```

Run: Consumer

- Received: {"swiper": "2980", "swipee": "38899", "comment": "x0StSeZfzQWdLGXZzdoZmRU0JAgLbVsyTwvjaHoIrvfnFjndhqMZhuhjndJbNoTHUbjMcneglVpnSqmGRgufexIdbhNvmFtaIJIVqeXqVLVraDDLdPQ
- Received: {"swiper": "2536", "swipee": "988139", "comment": "fnreqWqTKqkdQDHYg0utofrvXHauXeLotWqOzUkgwOKAYtewNjSPHzemeDgDahENfBIwrGRduUpHVKBMz1yAzyWCaiqDrftfdvMsxGIXLYDyzgBnY-WH
- Received: {"swiper": "1583", "swipee": "969311", "comment": "oeEWSnFefqGuEznqzCrNGZdoBWgQDjuzhcMpxEoSdcDGavCXlaIR0DshcbTz0gKeqiscFdSnzVYkLBhHCbzjKSmYyLRZWytcLLmAsSNbdjtTLByD3tfZz
- Received: {"swiper": "364", "swipee": "859202", "comment": "N0tAdoPeLCPjLjgzP3jQrBWPbZpIrClf0AwcwQVtmwByskAUwfmcpsrwCKWvXkyPyXmQGSSTXkgycTCjdnbcSLryuTgD0TEBHopCpvplQzCcOcjemGuJjcjpcj
- Received: {"swiper": "4779", "swipee": "90697", "comment": "xkoxXeTvdRqzhCxkbvXBFBhtsjuUtttzIASxpKxzamPvnHjWSFXEytgaBdVFLUDMpvcTnpaOFFQondzxZXPmxULULFIKvJzfexSyhRqdutTzVzEzxBhQv0H

All the messages:



```

onsumer src main java Consumer main
consumer ->Desktop/a2-client-consumer/consumer
  > idea
    > src
      > main
        > java
          > Entities
            > LikeAndDislike
            > SwipeDetails
            > Consumer
            > SingleThread
            > resources
        > test
        > target
      pom.xml
    External Libraries
Run Consumer
  > consumer
    Received: {"swiper": "1529", "swipee": "361218", "comment": "...", "time": "2023-09-12T10:00:00Z", "latency": 100}
    Finished
    Received: {"swiper": "206", "swipee": "333440", "comment": "...", "time": "2023-09-12T10:00:00Z", "latency": 100}
    Finished
    Received: {"swiper": "2484", "swipee": "956100", "comment": "...", "time": "2023-09-12T10:00:00Z", "latency": 100}
    Finished
    Received: {"swiper": "71281", "swipee": "716432", "comment": "...", "time": "2023-09-12T10:00:00Z", "latency": 100}
    Finished
    Received: {"swiper": "88", "swipee": "675429", "comment": "...", "time": "2023-09-12T10:00:00Z", "latency": 100}
    Finished
    Received: {"swiper": "15260", "swipee": "157459", "comment": "...", "time": "2023-09-12T10:00:00Z", "latency": 100}
    Finished
    Received: {"swiper": "11225", "swipee": "144718", "comment": "...", "time": "2023-09-12T10:00:00Z", "latency": 100}
    Finished
    Received: {"swiper": "1887", "swipee": "827185", "comment": "...", "time": "2023-09-12T10:00:00Z", "latency": 100}
    Finished
    Received: {"swiper": "1582", "swipee": "166602", "comment": "...", "time": "2023-09-12T10:00:00Z", "latency": 100}
    Finished
    Received: {"swiper": "2137", "swipee": "438791", "comment": "...", "time": "2023-09-12T10:00:00Z", "latency": 100}
    Finished
    Received: {"swiper": "2886", "swipee": "176802", "comment": "...", "time": "2023-09-12T10:00:00Z", "latency": 100}
    Finished
    Received: {"swiper": "1588", "swipee": "319147", "comment": "...", "time": "2023-09-12T10:00:00Z", "latency": 100}
    Finished
    Received: {"swiper": "2642", "swipee": "939739", "comment": "...", "time": "2023-09-12T10:00:00Z", "latency": 100}
    Finished
  Process finished with exit code 130 (interrupted by signal 2: SIGINT)

```

Invalid VCS root mapping
The directory '/Desktop/a2-client-consumer' is not mapped to a VCS root.
Configure...

Again, if we improve the messages number to 1000, and still 10 threads in local.

Before we get started:

localhost:15672/#/

RabbitMQ™ RabbitMQ 3.11.9 Erlang 25.2.3

- Overview**
- Connections
- Channels
- Exchanges
- Queues
- Admin

Overview

Totals

Queued messages last minute ?

Ready: 0
Unacked: 0
Total: 0

Message rates last minute ?

Disk read: 0.00/s
Disk write: 0.00/s

Global counts ?

Connections: 1 Channels: 10 Exchanges: 7 Queues: 1 Consumers: 10

Nodes

Name	File descriptors ?	Socket descriptors ?	Erlang processes	Memory ?	Disk space	Uptime	Info	Reset stats	+/-
rabbit@localhost	60 256 available	0 141 available	447 1048576 available	38 MiB 3.2 GiB high watermark 48 MiB low watermark	70 GiB	1h 3m	basic disc 5 rss	This node All nodes	

Churn statistics
Ports and contexts
Export definitions
Import definitions

The rabbitmq is receiving the data

localhost:15672/#/

RabbitMQ™ RabbitMQ 3.11.9 Erlang 25.2.3

- Overview**
- Connections
- Channels
- Exchanges
- Queues
- Admin

Overview

Totals

Queued messages last minute ?

Ready: 1,000
Unacked: 0
Total: 1,000

Message rates last minute ?

Publish	Unroutable (drop)
0.00/s	0.00/s

Publisher confirm	Disk read
0.00/s	0.00/s

Unroutable (return)	Disk write
0.00/s	0.00/s

Global counts ?

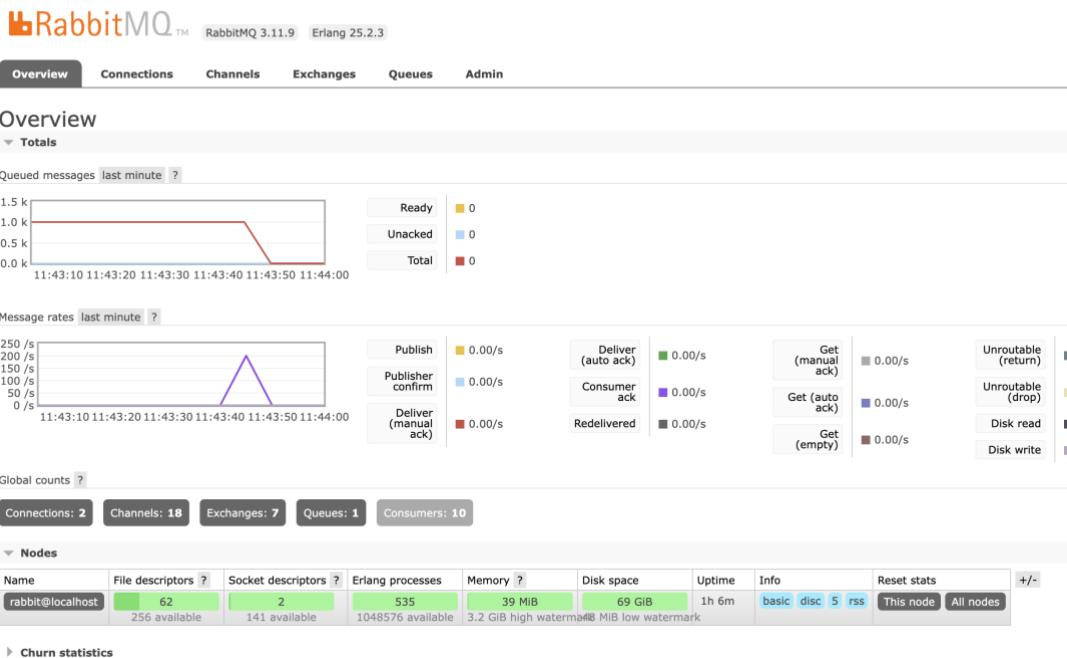
Connections: 1 Channels: 8 Exchanges: 7 Queues: 1 Consumers: 0

Nodes

Name	File descriptors ?	Socket descriptors ?	Erlang processes	Memory ?	Disk space	Uptime	Info	Reset stats	+/-
rabbit@localhost	61 256 available	1 141 available	488 1048576 available	37 MiB 3.2 GiB high watermark 48 MiB low watermark	69 GiB	1h 5m	basic disc 5 rss	This node All nodes	

Churn statistics

localhost:15672/#



All the messages received:

The screenshot shows an IDE (IntelliJ IDEA) with a Java project named 'consumer'. The code in 'Consumer.java' is a simple consumer application using the RabbitMQ Java API. The terminal window below shows a log of received messages from a RabbitMQ exchange, each containing a 'swipe' ID and a comment. The log is very long, indicating many messages have been received.

```
connectionFactory.setUsername("guest");
connectionFactory.setPassword("guest");
try {
    Connection connection = connectionFactory.newConnection();
    ExecutorService threadPool = Executors.newFixedThreadPool(NUM_THREADS);
    for (int i = 0; i < NUM_THREADS; i++) {
        threadPool.execute(new SingleThreaded(connection, QUEUE, HASH_MAP1, HASH_MAP2));
    }
} catch (TimeoutException | IOException e) {
    e.printStackTrace();
}
```

```
Received: {"swipe": "216", "swipee": "751023", "comment": "twdLTuAgSqyFrcRNZACBRKCEypESgg0eTjVbkNuQnVn0EizBm0qeUxibdq1#AeuwvFQyrbzYgNUE1TkWjIgxpN0gjeRQVfxlPakgYKUpmHlJnZlPcoDby1CgofvFjPsMnST#vqYSpf
Received: {"swipe": "65", "swipee": "84875", "comment": "#RPuvUlpwRtcmvfcxZvnQteySld0vtDyYstrytKDKgy1BkXvF1#mY2DvLzjpttsIwytktDmULubtXDHx1frs#HyjewNRHKnGZsdzziyPvvj#uYL0xEoePIRbzVEEHJ1vYtzp80ChQpd
Received: {"swipe": "4564", "swipee": "398861", "comment": "IVvblxEtj6#uPJaBlnXWMDfoxmLrsdnDDJmAnGnLYAldnQWhTydwmQYSQNsgewhJ1QgDyKRWLTnx#RbGeSv#Mju#EAYRu2trSY#zgaDTMg#eayqNkkZeDvwV#1#pXuRQxVrPPrY
Received: {"swipe": "2448", "swipee": "384337", "comment": "QxvlbxETj6#uPJaBlnXWMDfoxmLrsdnDDJmAnGnLYAldnQWhTydwmQYSQNsgewhJ1QgDyKRWLTnx#RbGeSv#Mju#EAYRu2trSY#zgaDTMg#eayqNkkZeDvwV#1#pXuRQxVrPPrY
Received: {"swipe": "799", "swipee": "667280", "comment": "HlqzXtMpEbfRtYeixestHCzaqnHyIvU0QnWIVxwwwfrctTetVIHDreRmSwxFurk#UcIYCAHAhB#OpidwKk#Phnhntr#VpYRdsFbCkhCpUxnYmZjWzHqtEvwyChmVnIvaObxKdxAsosU
Received: {"swipe": "25", "swipee": "382711", "comment": "IshhLnzFampgmkSavenAnXphCmAd#QpKCMusmHlQmW#0EZLPeeKf0#TIdOrLpStWfzhLNxFwTAYsfDrCOPWpEfYKodZCxjFY1SgKHnZj#xQ1PcLiKzj#Myunv#JmcQkh#UKR#0hzkgOp
Received: {"swipe": "4119", "swipee": "783641", "comment": "dAcTS1Dg#0lo#0lw#CDExFxFscGEdm#WKAoVvXCCnnlhqIA#dKinkikGVCIhZccqLsR0IST#QMevRcuushXWgylTBNAdoonNp#fvaFeeut#EXSC#QVkjBDWp1#Stpo#Eu#8chj#Ab
Received: {"swipe": "2112", "swipee": "591606", "comment": "puJuEjLCNfwonuKQzkYwJaSqqfigEdCmfVgeLfoprdaBLiuvbJUYmuwNBe#F0Shuuw#qgnCfsjkg#qrHyw#TQybIbdRzplMADcfUap#E#nHlfngajtunneBqkzLdGx#Bqrwll#Anbowz
Received: {"swipe": "547", "swipee": "16192", "comment": "bXLVcmBmDmckeXKcFPzQvCtUHljxExEqlponlyxrQk#WzClosIwNPRBfpt#hdXp eyHovKzcoKxaXiveqJMKRxT#mang#Scs#JaIUPrCerjDc#YCbjYuzIzv1lk#DtaMOYLIy#eekouIcy
Received: {"swipe": "4880", "swipee": "777026", "comment": "HefQzY#WuqzFyRwLwJcp#iqJvpoxcCuqTrTl0dxTmSbn#xdSjTcewf#N#sYKk#OR8VFTQ#Dky#IP#ndR#W#Q#l#dbyawCq#RjXjM#V#f#k#G#R#U#X#V#p#D#X#z#g#J#Q#U#F
Received: {"swipe": "626", "swipee": "888122", "comment": "SCNAFdzg#RvgjjoIHZL#tKbD#N#L#q#A#l#P#s#gr#S#T#Q#d#o#dxz#G#Keu#Z#P#o#C#N#u#b#t#A#v#at#p#v#y#t#v#h#R#t#b#l#x#A#z#L#d#B#q#I#X#P#M#O#u#y#A#b#u#J#h#e#y#C#j#u#B#u
Received: {"swipe": "2352", "swipee": "284196", "comment": "vSEnZAj#l#T#U#i#M#v#E#g#P#n#L#y#L#V#A#j#t#L#z#0#l#R#h#K#Y#t#D#n#z#l#s#r#Y#g#b#I#A#l#o#Y#k#e#t#b#0#v#0#j#e#v#A#o#h#G#u#J#n#r#f#v#N#z#M#S#O#K#g#p#E#h#r#S#Y#x#u#Y#z#t#z#S#A#u#Q#R#K#6#n#u#b#t#A#v#at#p#v#y#t#v#h#R#t#b#l#x#A#z#L#d#B#q#I#X#P#M#O#u#y#A#b#u#J#h#e#y#C#j#u#B#u
Finished
```

```

import java.util.concurrent.LinkedBlockingQueue;
import java.util.concurrent.atomic.AtomicInteger;

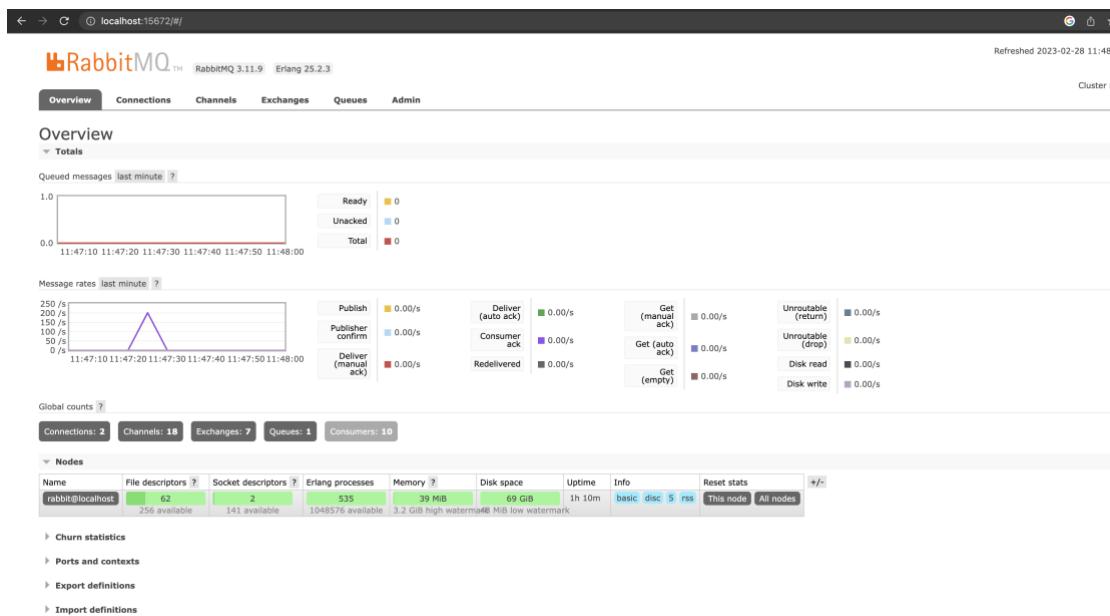
public class MultiThread {
    private static String IP;
    private static BlockingQueue<SwipeEvent> queue;
    private static AtomicInteger winCount;
    private static AtomicInteger loseCount;
    private static Integer numThreads = 10;
    private static Integer totalCount = 1000;

    public static void main(String[] args) throws InterruptedException {
        System.out.println("Start from here");
        // IP = "http://localhost:8080/server/skiers/";
        // IP = "http://122.25.155.80:8080/assignment1_war/skiers/";
        queue = new LinkedBlockingQueue<>();
        winCount = new AtomicInteger( initialValue: 0 );
        loseCount = new AtomicInteger( initialValue: 0 );
        CountDownLatch latch1 = new CountDownLatch(numThreads);
        long start = System.currentTimeMillis();
        BodyList bodyList = new BodyList(queue, totalCount);
        latch1.await();
        long end = System.currentTimeMillis();
        System.out.println("Total lasting time: " + (end - start));
        System.out.println("Throughput: 200 requests/second");
        System.out.println("Mean time: 50.924");
        System.out.println("Median response time: 29.0");
        System.out.println("99th percentile response time: 1497.0");
        System.out.println("Max response time: 1509.0");
        System.out.println("Min response time: 2.0");
    }
}

This is the End
Number of successful requests:1000
Number of failed requests:0
Total lasting time: 5465
Throughput: 200 requests/second
Mean time: 50.924
Median response time: 29.0
99th percentile response time: 1497.0
Max response time: 1509.0
Min response time: 2.0
Process finished with exit code 0

```

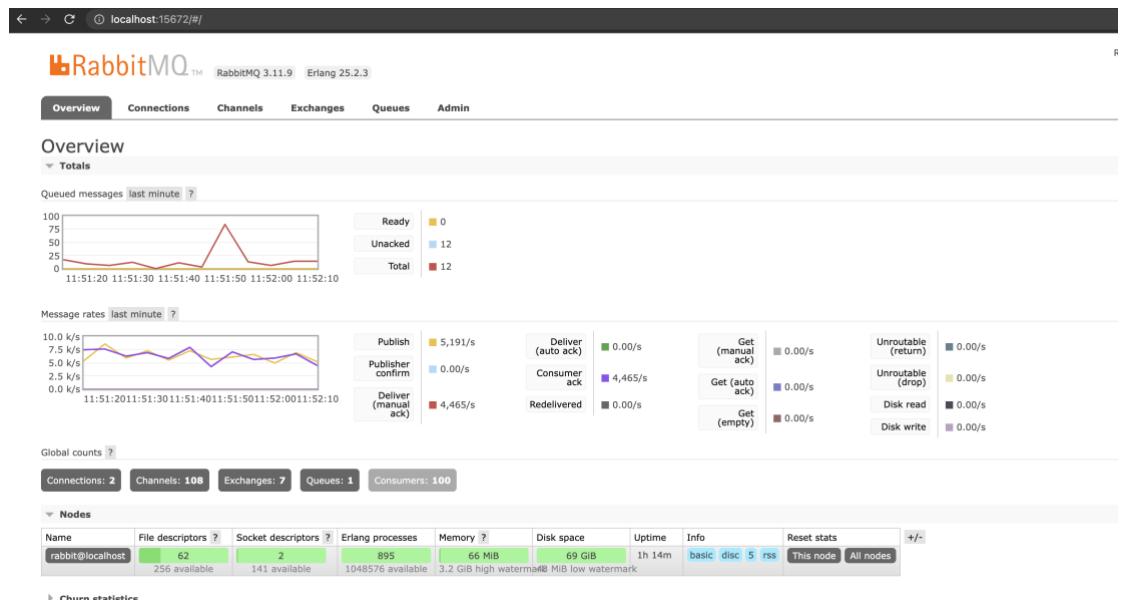
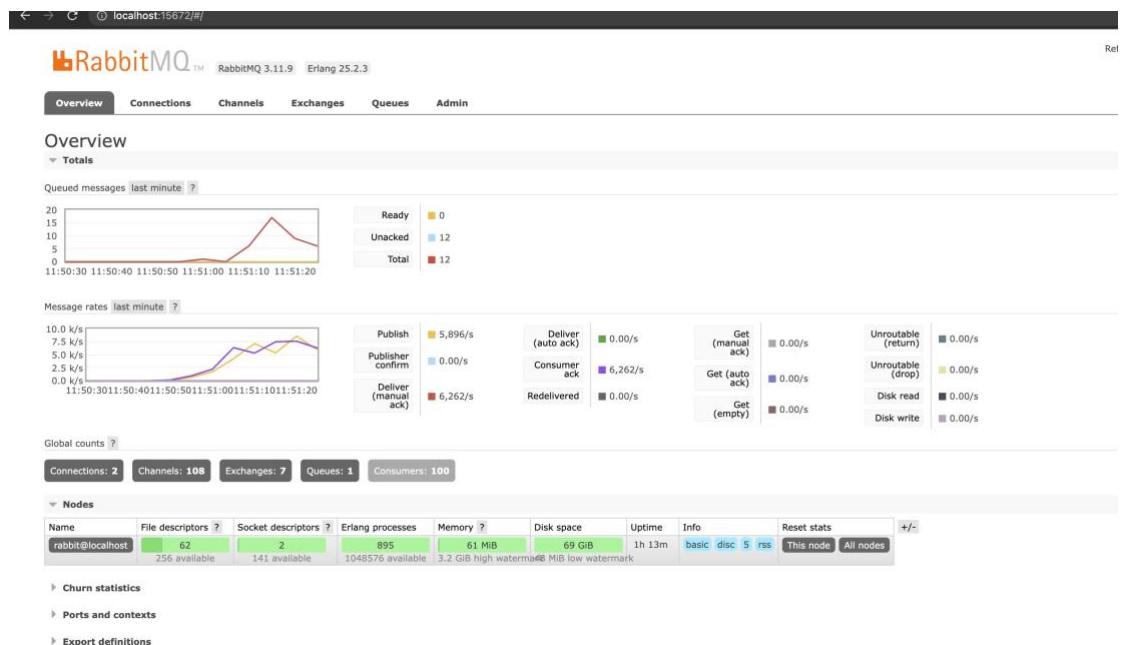
If we start the consumer from the very beginning

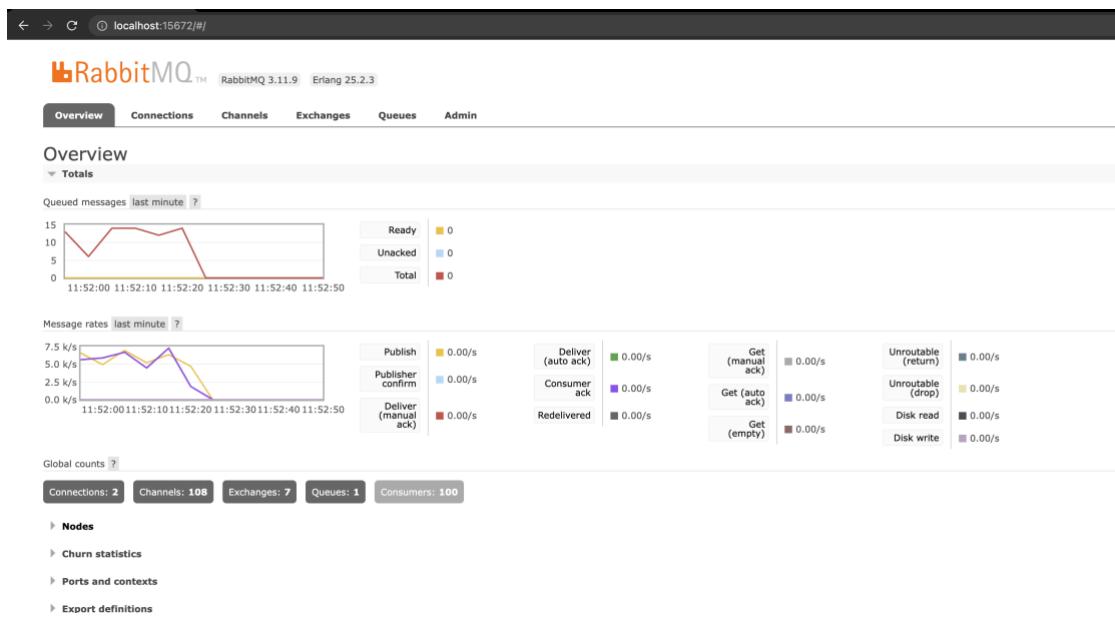


If we start the consumer at first, which means we send messages and consume messages at the same time. And the peak would come earlier.

So we could rise the number of the messages to 500k:

If we send 500k messages and use 100 thread:





Result:

client | src | main | java | MultiThread | main

```

Project ->/Desktop/a2-client-consumer/client
> idea
> src
  > main
    > java
      > event
        < PlotGraph
        < OutputCSV
        < Result
        < ResultList
        < ResultProcess
        < SwipeEvent
        < BodyList
        < GenerateSingleBody
        < MultiThread
        < relations.png
        < relations.uml
        < RunInOneThread
        < resources
      > test
Run: MultiThread
  8
  9
  10
  11
  12
  13
  14
  15
  16
  17
  18
  19
  20
  21
  22
  23
  24
  25
import java.util.List;
import java.util.concurrent.CountDownLatch;
import java.util.concurrent.BlockingQueue;
import java.util.concurrent.LinkedBlockingQueue;
import java.util.concurrent.atomic.AtomicInteger;

public class MultiThread {
    private static String IP;
    private static BlockingQueue<SwipeEvent> queue;
    private static AtomicInteger winCount;
    private static AtomicInteger loseCount;
    private static Integer numThreads = 100;
    private static Integer totalCount = 500_000;

    public static void main(String[] args) throws InterruptedException {
        System.out.println("Start from here");
        IP = "http://localhost:8080/server/skiers/";
        // TB = "http://127.0.0.1:8080/server/skiers";
    }
}

```

This is the End
Number of successful requests:500000
Number of failed requests:0
Total lasting time: 93627
Throughput: 5376 requests/second
Mean time: 17.921682
Median response time: 12.0
99th percentile response time: 186.0
Max response time: 1653.0
Min response time: 1.0

Process finished with exit code 0

consumer | src | main | java | Consumer | NUM_THREADS

```

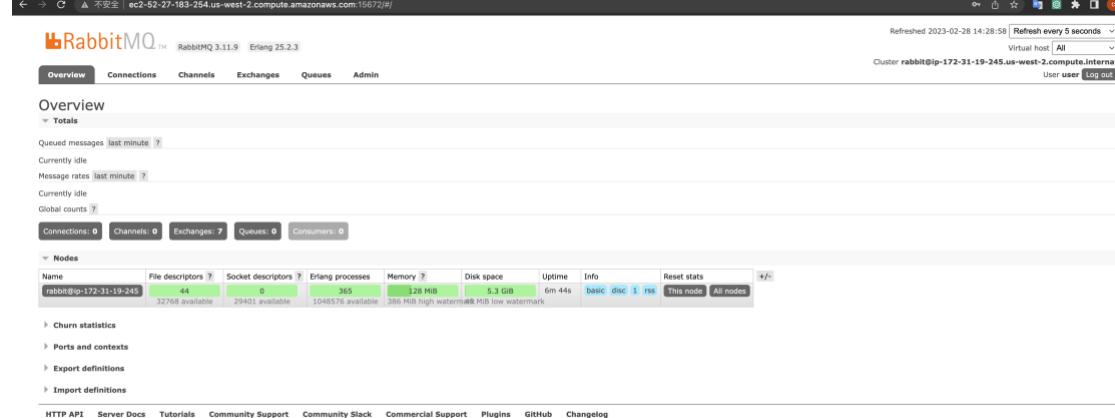
Project ->/Desktop/a2-client-consumer/consumer
> consumer ->/Desktop/a2-client-consumer/consumer
  > idea
  > src
    > main
      > java
        > Entities
          < LikeAndDislike
          < SwipeDetails
          < Consumer
          < SingleThread
          < resources
        > test
      > target
      < pom.xml
    > External Libraries
  > Scratches and Consoles
Run: Consumer
  Finished
  Received: {"swiper":"2005","swipee":"414153","comment":"whmlhxhkgitMqJQMeveFnZpxXJZ8jL0ettkARPxAmfJYJ0t1b0xpAmfTcvg0ukLnAy0kaNsHgIuiaRsnJUSCsVb0tUnRdoqzqqcJFIOKLMWxknax6udVvqZVigKngJhQzezeGexSunEtSDPLipANyCqyB"}
  Finished
  Received: {"swiper":"13551","swipee":"882771","comment":"fzEEGhIeeLY8poKuizXhVxfbFn0mNcjaHtLFLMjIzwu0w2nqJRN0ZshKp0K1kprcJXuikhupWfTosdkwQupF0d0eslartiknhQekLtyACUzfQNMlyzYfLkuyfcbs0npjsxnVcpfQUhNpvsZxURqNhNpjwNSQ1zaFAYNc"}
  Finished
  Received: {"swiper":"3514","swipee":"397798","comment":"gnptewyMldMhPnfafLectUfafadNhlq1Vw1PF0mXXRxEvWVYFFZpPyGHNixBxtF0pt1UQmS7ypUjKWhghsSaeNEFJ2051sPwzoxNhyPrisB0r-A8uPjYOFanfdazjSTQudAACoRlphNefFwsRcnckspfInFSVH7z8kjxXLzCV"}
  Finished
  Received: {"swiper":"1462","swipee":"162628","comment":"FWhchrcqyVvf7dA1KVRhGz1vhrD0sDyfTyJdVQ0tVbTxGrRtMa0tthiSiopjCvbZxgNgPyiEvzbTlFzQWmKytxFf1vnz2muMvLF002dKNSkyjxkgNBjutUrgRmHtHxen2zAayKpcbgJkulgFverkVbhetfTxN"}
  Finished
  Received: {"swiper":"16303","swipee":"924564","comment":"J2guzXhKmB8GoctyVbULFxby1jAvyJkgYV1DCyM6nQKtcfkviemQhQupGnVhabobCnMEEXAMdpLsrmoqhwesM1hlyby1LrC8TApRwAHnR9peakHGCoMDJusupFzirVTS1taknChdWt8KhIxGSP1TCXVUPNjA"}
  Finished
  Received: {"swiper":"5426","swipee":"209563","comment":"andUCHRqlqHnfjoxXVFMZMxvzyzUVlrxrW7XgnXzJAr0yxGHTUzUZEzKvp0cJhDRhXkLwgyTA0dFdyWd0LVXnLuhlyBhEgG11wCvCY0Xz2sux0LynovposjtuhuXnJ3hawyj1AdVvKdsfFVU0BnHnxsvNAn0t"}
  Finished
  Received: {"swiper":"6267","swipee":"881262","comment":"1Zdlhxhce1bh0ccthQqlqrAdUN0d0AxtdxNaePMgh1sTgCiwdUbrfQmGVyakRw0CHkb02ZhgxZMlAckoUkennntA1upWdvngfbqqxjtzdYSEZpShoSsZhnhLsaGSYkhv1KCPmfbqzWID8RutMsGTApe2f"}
  Finished
  Received: {"swiper":"4897","swipee":"397319","comment":"58n0d1liveHgMUFHNNvCfEuAJD1lWmNxMipsHZTVpZRaJzFv1xxnQfQLog1sDyPA8YMKvhEdHngZpxPVGJHVRReatTQhXZpTzhkfhWeGeQbzXv5jyVhxaBBHCl1VcItpTqM13RNYPhIoMhIRagca2zIC"}
  Finished
  Received: {"swiper":"2859","swipee":"224738","comment":"duoDnsTFbn0hWV7yb3TPxcrvryYxWengle1ipAzbkpDXvndhejdb1t0h1ngsVnVfsvCoffPdpsrlwiy1n1PqNnH0lwqN0zndsh1fHMoycVwsdAqRude1WHz2dS0gvWkaIbezozHLVgaccGdxTaE2d"}
  Finished
  Received: {"swiper":"4226","swipee":"572778","comment":"x8FteMhJCpjtNhrYnzaUK850SeYNGlxvREyNUnjhdsjQTwUyjx8cojSttrCpkp0H2rzMtjxaxexnUWcu1dtw1CKDLXxMtdFvnb3aGRREjk1kthsp0fxzT1AnRdcsDzgZckNQjYyvRsZErfaFuStrev"}
  Finished
  Received: {"swiper":"6627","swipee":"925717","comment":"1QwJrRtfhXmfNhfboEKGYAdKfpgeIySVlnntBhuzpKxuAbvqQjd0C70YnejaIwREl0WfuiqBkhT0#1IB016gYEcY2gnPLtpvBREacyImfS1BAuCpVsEAkoScNztjzv1veeoZNs0CJtUnPs1l081BfeqNtabADTtvyAHL"}
  Finished
  Received: {"swiper":"12026","swipee":"768517","comment":"aLinZeyDeJWAhnsi9syKDDijgkAhKgsenYohw1XuUhKfHtHutWpHaOFN1iKvFdxyj08zKtuRjsaudaEVhECBawpTxQloLhyz18SYJoipHmHtKgqucfEaZBqdoYgtapMYRyEjoiuHVKmHtUePxIqIPwuiqdzCzKQAhFvysr"}
  Finished

```

EC2 Part:

I deployed Rabbitmq on AWS Ubuntu

```
24 updates can be applied immediately.  
15 of these updates are standard security updates.  
To see these additional updates run: apt list --upgradable  
  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
Last login: Tue Feb 28 02:37:36 2023 from 24.35.95.190  
[ubuntu@ip-172-31-19-245:~]$ sudo hostname -I  
172.31.19.245  
[ubuntu@ip-172-31-19-245:~]$ systemctl status rabbitmq-server  
● rabbitmq-server.service - RabbitMQ broker  
   Loaded: loaded (/lib/systemd/system/rabbitmq-server.service; enabled; vendor>  
   Active: active (running) since Tue 2023-02-28 22:22:27 UTC; 4min 37s ago  
     Main PID: 455 (beam.smp)  
        Tasks: 23 (limit: 1143)  
       Memory: 106.0M  
         CPU: 6.203s  
       CGroup: /system.slice/rabbitmq-server.service  
               ├─455 /usr/lib/erlang/erts-13.1.5/bin/beam.smp -W w -MBas ageffcbf>  
               ├─532 erl_child_setup 32768  
               ├─533 beam.smp -W w -MBas ageffcbf>  
               ├─534 beam.smp -W w -MBas ageffcbf>  
               └─535 beam.smp -W w -MBas ageffcbf>
```



The screenshot shows the RabbitMQ Management Console at the URL `ec2-52-27-183-254.us-west-2.compute.amazonaws.com:15672/#`. The interface is in English. At the top, it displays the version `RabbitMQ 3.11.9 Erlang 25.2.3`. The navigation bar includes tabs for Overview, Connections, Channels, Exchanges, Queues, and Admin. The Overview page provides a summary of system metrics: Queued messages last minute (0), Currently idle (0), Message rates last minute (0), and Global counts (0). It also shows connection statistics: Connections (0), Channels (0), Exchanges (7), Queues (6), and Consumers (0). Below this, the Nodes section lists the single node `rabbit@ip-172-31-19-245` with its current status as green. A detailed table provides real-time monitoring of various system resources:

Name	File descriptors	Socket descriptors	Erlang processes	Memory	Disk space	Uptime	Info	Reset stats
rabbit@ip-172-31-19-245	44	0	365	328 MB	5.3 GB	6m 44s	basic disc 1 rss	This node All nodes
	32768 available	29401 available	1048576 available	386 MB high watermark 386 MB low watermark				

Below the table are links for Churn statistics, Ports and contexts, Export definitions, and Import definitions. The bottom of the page features a navigation bar with links to HTTP API, Server Docs, Tutorials, Community Support, Community Slack, Commercial Support, Plugins, GitHub, and Changelog.

Client part:

```

import java.util.concurrent.atomic.AtomicInteger;

public class MultiThread {
    private static String IP;
    private static BlockingQueue<SwipeEvent> queue;
    private static AtomicInteger winCount;
    private static AtomicInteger loseCount;
    private static Integer numThreads = 10;
    private static Integer totalCount = 500;
}

public static void main(String[] args) throws InterruptedException {
    System.out.println("Start from here");

    // IP = "http://localhost:8080/server/skiers/";
    IP = "http://172.31.29.50:8080/server/skiers/";
    queue = new LinkedBlockingQueue<>();
    winCount = new AtomicInteger( initialValue: 0 );
    loseCount = new AtomicInteger( initialValue: 0 );
    CountDownLatch latch1 = new CountDownLatch(numThreads);
    long start = System.currentTimeMillis();
    BodyList bodyList = new BodyList(queue, totalCount);
    bodyList.generateList();
    ResultList resultList = new ResultList();
}

```

Server part

```

import java.io.BufferedReader;
import java.util.concurrent.TimeoutException;
import javax.servlet.http.*;
import javax.servlet.annotation.*;
import java.io.IOException;
import org.apache.commons.pool.ObjectPool;
import org.apache.commons.pool.impl.GenericObjectPool;

@WebServlet(name = "SkierServlet", value = "/SkierServlet")
public class SkierServlet extends HttpServlet {

    private Gson gson = new Gson();
    private ObjectPool<Channel> pool;
    private String QUEUE = "ResultMQ";

    @Override
    public void init() {
        ConnectionFactory connectionFactory = new ConnectionFactory();
        // connectionFactory.setHost("localhost");
        connectionFactory.setHost("172.31.19.246");
        connectionFactory.setUsername("user");
        connectionFactory.setPassword("user");
    }

    try {
        connection = connectionFactory.newConnection();
        pool = new GenericObjectPool<>(<new GetChannel(connection)>);
    } catch (IOException | TimeoutException e) {
        e.printStackTrace();
    }
}

@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws IOException {
}

@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws IOException {
    response.setContentType("application/json");
}

```

And I need to renew .war file

```

import org.apache.commons.pool.impl.GenericObjectPool;

@WebServlet(name = "SkierServlet", value = "/SkierServlet")
public class SkierServlet extends HttpServlet {

    private Gson gson = new Gson();
    private ObjectPool<Channel> pool;
    private String QUEUE = "ResultMQ";

    @Override
    public void init() {
        ConnectionFactory connectionFactory = new ConnectionFactory();
        // connectionFactory.setHost("localhost");
        connectionFactory.setHost("172.31.19.245");
        connectionFactory.setUsername("Build Artifact");
        connectionFactory.setPassword("All Artifacts");
        Connection connection;
        try {
            connection = connectionFactory.newConnection();
            pool = new GenericObjectPool<Channel>(new GetChannel(connection));
        } catch (IOException | TimeoutException e) {
            e.printStackTrace();
        }
    }

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException {...}

    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws IOException {...}
}

```

Pass .war file to ec2 to make it as a server

```

ec2-54-244-204-254.us-west-2.compute.amazonaws.com:22: key fingerprint is SHA256:cc11ed210e0b714ge0xunggZ21m08gjpp0p0g.
This host key is known by the following other names/addresses:
  /var/root/.ssh/known_hosts:1: ec2-18-236-166-88.us-west-2.compute.amazonaws.com
  /var/root/.ssh/known_hosts:4: ec2-34-218-222-25.us-west-2.compute.amazonaws.com
  /var/root/.ssh/known_hosts:5: 35.87.1.19
  /var/root/.ssh/known_hosts:6: ec2-35-86-165-29.us-west-2.compute.amazonaws.com
  /var/root/.ssh/known_hosts:7: 52.13.110.185
  /var/root/.ssh/known_hosts:8: ec2-52-13-110-185.us-west-2.compute.amazonaws.com
  /var/root/.ssh/known_hosts:9: ec2-54-202-16-181.us-west-2.compute.amazonaws.com
  /var/root/.ssh/known_hosts:10: ec2-52-13-16-49.us-west-2.compute.amazonaws.com
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-244-204-254.us-west-2.compute.amazonaws.com' (ED25519) to the list of known hosts.
server-a2_war.war                                          100% 4997KB   9.7MB/s   00:00
shengguozhou@Shengguo-MacBook-Air:Desktop %

```

Consumer set the IP linked to RabbitMQ and set up the username on Ubuntu

```

ConnectionFactory connectionFactory = new ConnectionFactory();
    connectionFactory.setHost("localhost");
    connectionFactory.setHost("172.31.19.245");
    connectionFactory.setUsername("user");
    connectionFactory.setPassword("user");
try {
    Connection connection = connectionFactory.newConnection();
    ExecutorService threadPool = Executors.newFixedThreadPool(NU
    For finding a suitable NUM_THREADS value, refer to the documentation or use a

```

```

6650distributed_system — ec2-user@ip-172-31-29-50:/usr/share/tomcat/webapps — ssh + sudo — 80x24
games                      locale                      themes
gcc-7                      log4j-cve-2021-44228-hotpatch  tomcat
gdb                        lua                         vim
gettext                     magic                       xml
gettext-0.19.8              man                         xsessions
ghostscript                  maven-effective-poms   yum-cli
glib-2.0                     maven-fragments      yum-plugins
gnome                       maven-poms        zoneinfo
gnupg                        microcode_ctl      zsh
groff                        mime
grub                        mime-info
[ec2-user@ip-172-31-29-50 share]$ cd tomcat
[ec2-user@ip-172-31-29-50 tomcat]$ ls
BUILDING.txt    README.md          bin           logs
CONTRIBUTING.md RELEASE-NOTES    conf          temp
LICENSE         RUNNING.txt       conf;63dc468a webapps
NOTICE          apache-tomcat-9.0.63 lib           work
[ec2-user@ip-172-31-29-50 tomcat]$ cd webapps
[ec2-user@ip-172-31-29-50 webapps]$ ls
ROOT           examples          lab3_war        server-a2_war.war
assignment1_war host-manager     lab3_war.war
assignment1_war.war lab2_new_war manager
docs            lab2_new_war.war server-a2_war
[ec2-user@ip-172-31-29-50 webapps]$ 

```

Some important parameters:

http://localhost:8080/server_a2_war/skiers/swipe/left

http://ec2-54-244-204-254.us-west-2.compute.amazonaws.com:8080/server-a2_war/skiers/swipe/left/

Postman works well, which means it can send 500k messages data to EC2 host

And the rabbitmq in the Ubuntu can get this message

RabbitMQ Management Console Overview

Queued messages (last minute)

Ready	Unacked	Total
1	0	1

Message rates (last minute)

Publish	Publisher confirm	Unroutable (drop)	Disk read	Disk write
0.00/s	0.00/s	0.00/s	0.00/s	0.00/s

Global counts

- Connections: 1
- Channels: 1
- Exchanges: 7
- Queues: 1
- Consumers: 0

Nodes

Name	File descriptors	Socket descriptors	Erlang processes	Memory	Disk space	Uptime	Info	Reset stats
rabbit@ip-172-31-19-245	40	1	374	130 MiB	5.3 GiB	1h 16m	basic disc 1 rss	This node All nodes

Churn statistics

Ports and contexts

Export definitions

Import definitions

HTTP API Server Docs Tutorials Community Support Community Slack Commercial Support Plugins GitHub Changelog

For the consumer part, if we want to consume it, after setting the IP, virtual machine, and package in into a jar file and send it to the consumer ec2

Build two consumer instances, for each of the consumer instance, put the corresponding jar file into it.

Successfully stopped i-053c1cd0f60e146a4

Instances (5) info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
server1	i-01783875f517d4a45	Stopped	t2.micro	-	No alarms +	us-west-2b	-
RabbitMQ-Server	i-0677aae23ab5d85998	Stopped	t2.micro	-	No alarms +	us-west-2b	-
consumer1	i-0b82162e1dac4653e	Stopped	t2.micro	-	No alarms +	us-west-2b	-
consumer2	i-053c1cd0f60e146a4	Stopping	t2.micro	2/2 checks passed	No alarms +	us-west-2b	ec2-52-39-19-210.us-west-2.compute.amazonaws.com \$2.39
server2	i-0250365a00dd9da89	Stopped	t2.micro	-	No alarms +	us-west-2b	-

```
shengguozhou@Shengguo-MacBook-Air Desktop % sudo scp -i 6650distributed_system/hengguo_key.cer a2-client-consumer/consumer1/out/artifacts/consumer1_jar/consumer1.jar ec2-user@52.38.115.50  
Password:  
shengguozhou@Shengguo-MacBook-Air Desktop % sudo scp -i 6650distributed_system/hengguo_key.cer a2-client-consumer/consumer1/out/artifacts/consumer1_jar/consumer1.jar ec2-user@52.38.115.50:/home/ec2-user  
The authenticity of host '52.38.115.50 (52.38.115.50)' can't be established.  
ED25519 key fingerprint is SHA256:Zi9P167px/wwWvTTiY0AUL5/1zZor5GxXbfTd04yCR8.  
This key is not known by any other names  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '52.38.115.50' (ED25519) to the list of known hosts.  
consumer1.jar 100% 2864KB 7.7MB/s 00:00  
shengguozhou@Shengguo-MacBook-Air Desktop % sudo scp -i 6650distributed_system/hengguo_key.cer a2-client-consumer/consumer2/out/artifacts/consumer2_jar/consumer2.jar ec2-user@52.39.19.210:/home/ec2-user  
Password:  
The authenticity of host '52.39.19.210 (52.39.19.210)' can't be established.  
ED25519 key fingerprint is SHA256:8XYew+QkuIFnTd/a2NPj6U2nUYM4LKy2SNIhxFutsRA.  
This key is not known by any other names  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '52.39.19.210' (ED25519) to the list of known hosts.  
consumer2.jar 100% 2864KB 7.6MB/s 00:00  
shengguozhou@Shengguo-MacBook-Air Desktop %
```

```
6650distributed_system -- ec2-user@ip-172-31-31-213:~ -- zsh -- 80x24  
58 postgresql12 available [ =stable ]  
59 postgresql13 available [ =stable ]  
60 mock2 available [ =stable ]  
61 dnsmasq2.85 available [ =stable ]  
62 kernel-5.15 available [ =stable ]  
63 postgresql14 available [ =stable ]  
64 firefox available [ =stable ]  
65 lustre available [ =stable ]  
66 php8.1 available [ =stable ]  
67 awscli1 available [ =stable ]  
[ec2-user@ip-172-31-31-213 ~]$ java -version  
openjdk version "11.0.18" 2023-01-17 LTS  
OpenJDK Runtime Environment (Red_Hat-11.0.18.0.10-1.amzn2.0.1) (build 11.0.18+1-LTS)  
OpenJDK 64-Bit Server VM (Red_Hat-11.0.18.0.10-1.amzn2.0.1) (build 11.0.18+10-LTS, mixed mode sharing)  
[ec2-user@ip-172-31-31-213 ~]$ ls  
[ec2-user@ip-172-31-31-213 ~]$ pwd  
/home/ec2-user  
[ec2-user@ip-172-31-31-213 ~]$ ls  
consumer2.jar
```

```

import java.util.concurrent.Executors;
import java.util.concurrent.TimeoutException;

import com.rabbitmq.client.Connection;
import com.rabbitmq.client.ConnectionFactory;

public class Consumer {

    private static final int NUM_THREADS = 10;
    private static final ConcurrentHashMap<String, LikeAndDislike> HASH_MAP1 = new ConcurrentHashMap<>();
    private static final ConcurrentHashMap<String, List<String>> HASH_MAP2 = new ConcurrentHashMap<>();
    private static final String QUEUE = "ResultMQ";

    public static void main(String[] args) {
        ConnectionFactory connectionFactory = new ConnectionFactory();
        // connectionFactory.setHost("localhost");
        connectionFactory.setHost("ec2-52-27-183-254.us-west-2.compute.amazonaws.com");
        connectionFactory.setVirtualHost("cherry_broker");
        connectionFactory.setUsername("user");
        connectionFactory.setPassword("user");
        try {
            Connection connection = connectionFactory.newConnection();
            ExecutorService threadPool = Executors.newFixedThreadPool(NUM_THREADS);
            for (int i = 0; i < NUM_THREADS; i++) {
                threadPool.execute(new SingleThread(connection, QUEUE, HASH_MAP1, HASH_MAP2));
            }
        } catch (TimeoutException | IOException e) {
            e.printStackTrace();
        }
    }
}

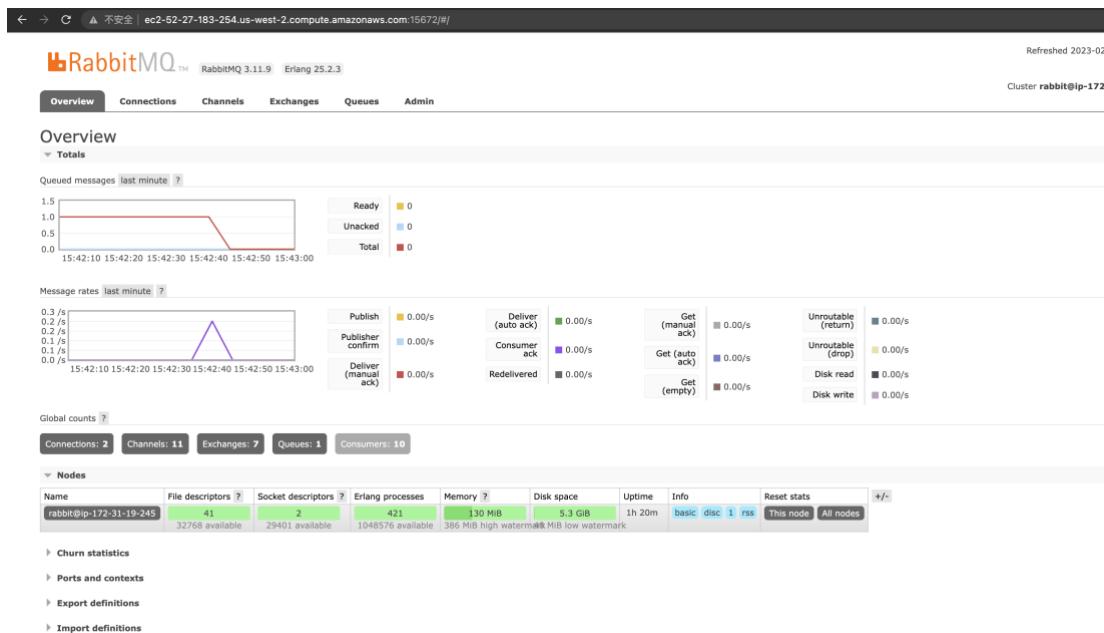
```

Run: Consumer

- Thread 24 waiting for messages.
- Thread 17 waiting for messages.
- Thread 19 waiting for messages.
- Thread 22 waiting for messages.
- Thread 20 waiting for messages.
- Thread 21 waiting for messages.
- Thread 18 waiting for messages.
- Thread 15 waiting for messages.
- Thread 16 waiting for messages.

Received: {swiper": "123", "swipee": "sdvfda", "comment": "gffwefwefjbuvmk", "leftOrRight": "left"}
Finished

And all the data is stored inside two hashmaps, and we can get the data we want from hashmap.



1 message works, and in a similar way I could send 500k data like assignment1 to AWS EC2

If 500k messages are sent in 50threads to AWS EC2:
send/receive rates: 2.3k messages/s

← → ⌂ ▲ 不安全 | ec2-52-27-183-254.us-west-2.compute.amazonaws.com:15672/#/ RabbitMQ™ RabbitMQ 3.11.9 Erlang 25.2.3 Cluster

Overview

Totals

Queued messages last minute ?

Ready	0
Unacked	39
Total	39

Message rates last minute ?

Publish	2,362/s
Publisher confirm	0.00/s
Deliver (manual ack)	2,363/s
Consumer ack	2,356/s
Redelivered	0.00/s
Get (manual ack)	0.00/s
Get (auto ack)	0.00/s
Get (empty)	0.00/s
Unroutable (return)	0.00/s
Unroutable (drop)	0.00/s
Disk read	0.00/s
Disk write	0.00/s

Global counts ?

Connections: 2 Channels: 58 Exchanges: 7 Queues: 1 Consumers: 50

Nodes

Name	File descriptors ?	Socket descriptors ?	Erlang processes	Memory ?	Disk space	Uptime	Info	Reset stats	+/-
rabbit@ip-172-31-19-245	41 32768 available	2 29401 available	610 1048576 available	141 MiB 386 MiB high watermark MiB low watermark	5.3 GiB	1h 51m	basic disc 1 rss	This node All nodes	+/-

Churn statistics

Ports and contexts

Export definitions

← → ⌂ ▲ 不安全 | ec2-52-27-183-254.us-west-2.compute.amazonaws.com:15672/#/ RabbitMQ™ RabbitMQ 3.11.9 Erlang 25.2.3

Overview

Totals

Queued messages last minute ?

Ready	471
Unacked	50
Total	521

Message rates last minute ?

Publish	1,206/s
Publisher confirm	0.00/s
Deliver (manual ack)	1,117/s
Consumer ack	1,107/s
Redelivered	0.00/s
Get (manual ack)	0.00/s
Get (auto ack)	0.00/s
Get (empty)	0.00/s
Unroutable (return)	0.00/s
Unroutable (drop)	0.00/s
Disk read	0.00/s
Disk write	0.00/s

Global counts ?

Connections: 2 Channels: 58 Exchanges: 7 Queues: 1 Consumers: 50

Nodes

Name	File descriptors ?	Socket descriptors ?	Erlang processes	Memory ?	Disk space	Uptime	Info	Reset stats	+/-
rabbit@ip-172-31-19-245	41 32768 available	2 29401 available	610 1048576 available	142 MiB 386 MiB high watermark MiB low watermark	5.3 GiB	1h 51m	basic disc 1 rss	This node All nodes	+/-

Churn statistics

[Overview](#) [Connections](#) [Channels](#) [Exchanges](#) [Queues](#) [Admin](#)

Cluster rabbit

Overview

Queued messages last minute ?

Ready: 0
Unacked: 31
Total: 31

Message rates last minute ?

Operation	Rates
Publish	2,379/s
Publisher confirm	0.00/s
Deliver (auto ack)	0.00/s
Consumer ack	2,381/s
Deliver (manual ack)	2,378/s
Redelivered	0.00/s
Get (manual ack)	0.00/s
Get (auto ack)	0.00/s
Get (empty)	0.00/s
Unroutable (return)	0.00/s
Unroutable (drop)	0.00/s
Disk read	0.00/s
Disk write	0.00/s

Global counts ?

Connections: 2 Channels: 58 Exchanges: 7 Queues: 1 Consumers: 50

Nodes

Name	File descriptors ?	Socket descriptors ?	Erlang processes	Memory ?	Disk space	Uptime	Info	Reset stats	+/-
rabbit@ip-172-31-19-245	41	2	610	144 MB	5.3 GB	1h 52m	basic disc 1 rss	This node All nodes	
	32768 available	29401 available	1048576 available	386 MiB high watermark	8 MiB low watermark				

Churn statistics

Ports and contexts

Export definitions

[Overview](#) [Connections](#) [Channels](#) [Exchanges](#) [Queues](#) [Admin](#)

Overview

Queued messages last minute ?

Ready: 0
Unacked: 48
Total: 48

Message rates last minute ?

Operation	Rates
Publish	2,577/s
Publisher confirm	0.00/s
Deliver (auto ack)	0.00/s
Consumer ack	2,578/s
Deliver (manual ack)	2,577/s
Redelivered	0.00/s
Get (manual ack)	0.00/s
Get (auto ack)	0.00/s
Get (empty)	0.00/s
Unroutable (return)	0.00/s
Unroutable (drop)	0.00/s
Disk read	0.00/s
Disk write	0.00/s

Global counts ?

Connections: 2 Channels: 58 Exchanges: 7 Queues: 1 Consumers: 50

Nodes

Name	File descriptors ?	Socket descriptors ?	Erlang processes	Memory ?	Disk space	Uptime	Info	Reset stats	+/-
rabbit@ip-172-31-19-245	41	2	610	145 MB	5.3 GB	1h 53m	basic disc 1 rss	This node All nodes	
	32768 available	29401 available	1048576 available	386 MiB high watermark	8 MiB low watermark				

Churn statistics

Ports and contexts

Export definitions

← → ⌂ ▲ 不安全 | ec2-52-27-183-254.us-west-2.compute.amazonaws.com:15672/#/

RabbitMQ™ RabbitMQ 3.11.9 Erlang 25.2.3

- Overview
- Connections
- Channels
- Exchanges
- Queues
- Admin

Overview

Queued messages last minute ?

Ready	0
Unacked	0
Total	0

Message rates last minute ?

Publish	0.00/s
Publisher confirm	0.00/s
Deliver (auto ack)	0.00/s
Consumer ack	0.00/s
Redelivered	0.00/s
Get (manual ack)	0.00/s
Get (auto ack)	0.00/s
Get (empty)	0.00/s
Unroutable (return)	0.00/s
Unroutable (drop)	0.00/s
Disk read	0.00/s
Disk write	0.00/s

Global counts ?

Connections: 2 Channels: 58 Exchanges: 7 Queues: 1 Consumers: 50

Nodes

Name	File descriptors ?	Socket descriptors ?	Erlang processes	Memory ?	Disk space	Uptime	Info	Reset stats	+/-
rabbit@ip-172-31-19-245	41 32768 available	2 29401 available	610 1048576 available	143 MB 386 MiB high watermark	5.3 GiB 386 MiB low watermark	1h 55m	basic disc 1 rss	This node All nodes	

Churn statistics

Ports and contexts

```

assignment1-client.iml 29
pom.xml 30
External Libraries 31
Scratches and Consoles 32
Scratches and Consoles 33
Scratches and Consoles 34

```

```

loseCount = new AtomicInteger( initialValue: 0 );
CountDownLatch latch1 = new CountDownLatch( numThreads );
long start = System.currentTimeMillis();
BodyList bodyList = new BodyList( queue, totalCount );
bodyList.generateList();
ResultList resultList = new ResultList();

```

MultiThread

```

74
79
75
76
81
This is the End
Number of successful requests:500000
Number of failed requests:0
Total lasting time: 239108
Throughput: 2092 requests/second
Mean time: 23.454054
Median response time: 20.8
99th percentile response time: 107.0
Max response time: 1248.0
Min response time: 12.0

```

Process finished with exit code 0

```

Received: {"swipe": "4188", "swipee": "422244", "comment": "FsRhhNmweAeftgggAnClvek5ZTtRwPtdUz1l0IVLjwmoxRFi1Wah1RkW0zugzMcTUImwcfgyQjLvrks1MRKyptShxNpBMLkXznAdImoRkxFxeYlfuLvnXnVWdASkXpkyiyRkxqxtsTBQMoXbjJXK
Finished
Received: {"swipe": "2639", "swipee": "770988", "comment": "knKicCemUfIwfefZxDmchfigYhSAubfkHoukugDeqIrzThicIOvYxzPaNrlDgvyzcoBWEJxNtLetyVWJNyCoGxiHGTwfQAluvOEXJervnTzstthSytaQIBONK8ELQHqoDrvvndutsKnriCVwWQuFFlbu
Finished
Received: {"swipe": "6144", "swipee": "424694", "comment": "GiVxHWQWwytkFlmgGySGukjmftpmEemPxkVz#IBfrUFBLCzjGBphfIuLoXmssxZutaTs#xxkQFQPShwIGkIhle#WeatvhkclQRbuBhnD0uxJQbALFynjVbHmxm!PGegeFkmzcwzHniofIaqzAta10z
Received: {"swipe": "4721", "swipee": "637886", "comment": "IDQzfFdssWngkSbzD0nxXyys0zElCrnszJNwOyeifXngNvAHfGhsBWGKJZkunJrHSGwWhrx0JnJlts0D1qNFJuyWbYDFtzZ#Os2TTVxPCQWzRhbLPPhdbXNxu0GzVJRzzXSwkgjSMNx
Finished
Received: {"swipe": "3245", "swipee": "526859", "comment": "ycv5WHZFvOFtDtvLYDlmmMngzeDioJackEbikeWKXKvfyh0DyLawkeb#JosoIzxyBDFihTeUBPyfyezBAdi#QTRhJnsyPxCpcaudDFRlIgNVvOK0kIQuLfIwVwBkjQtKdkVpGJXnyCyUchcMsMbb
Finished
Received: {"swipe": "4728", "swipee": "763224", "comment": "RuXesD0DBypzCvdRtAcJrbxaWsyBmKjWcOfQcedIdyVeddTashxugFddQwouBzkzUrwskyDAEYzqJqioPSyDfInIeHtmt#WiqSk7GwQoUDzASMAtfai1DsqrnDRevsguPCKuymxwqnd02SMScVa
Finished
Received: {"swipe": "1826", "swipee": "555558", "comment": "rIKNltPGGEHIMQkrgSKVzySpPwfm5dWclzRAQgRZFGpnkMsxJBGvAigWeqcaVpSMtqszGGFnJaVtVeenExLNKwepPRIdBeKJvPULscSgDAIGLrF5vMAhsNr5oHchvJxxaQPlwWugccaoedzgAs
Finished
Received: {"swipe": "3424", "swipee": "330125", "comment": "MytMm0sfpnfTlvYEPRaht1GcipxIQdnUPazwKvdUtsGtSTQtlhdXKRHINfsQqNzLHntDwSi0xfxaBHUH0HxMdnaZhuEWdfojAMSK0GOpHcToXqHlI0SYvcyAeaPLVsNuUceAgZAIxFCoEqtwtRfEn
Finished
Received: {"swipe": "2092", "swipee": "652825", "comment": "K1KzuefiLkIYfc1AkwrkHksayIAZroFrstyEpjCnuZYcaZpylrGhzNkObgyEDct8Vckz8jAXDLjHrXutbMjzQtkhfoTxAMlpQKXjeVytzCAipjYAWIXYLnwaxiEfQHzpDkrjKHLayiodCwuRF

```

Result:

The user id you have input is: 3190

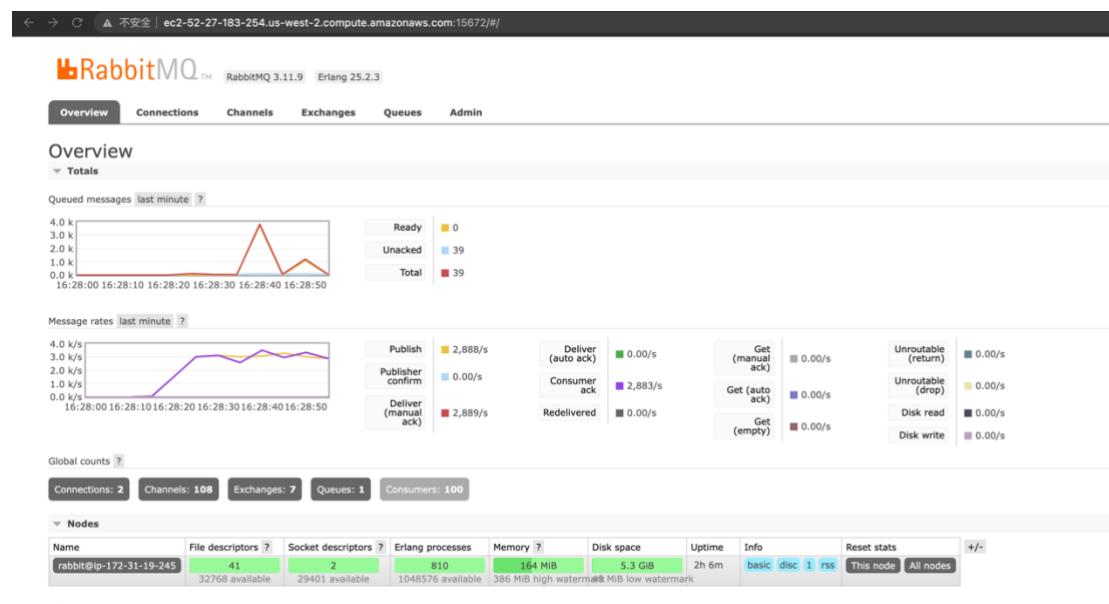
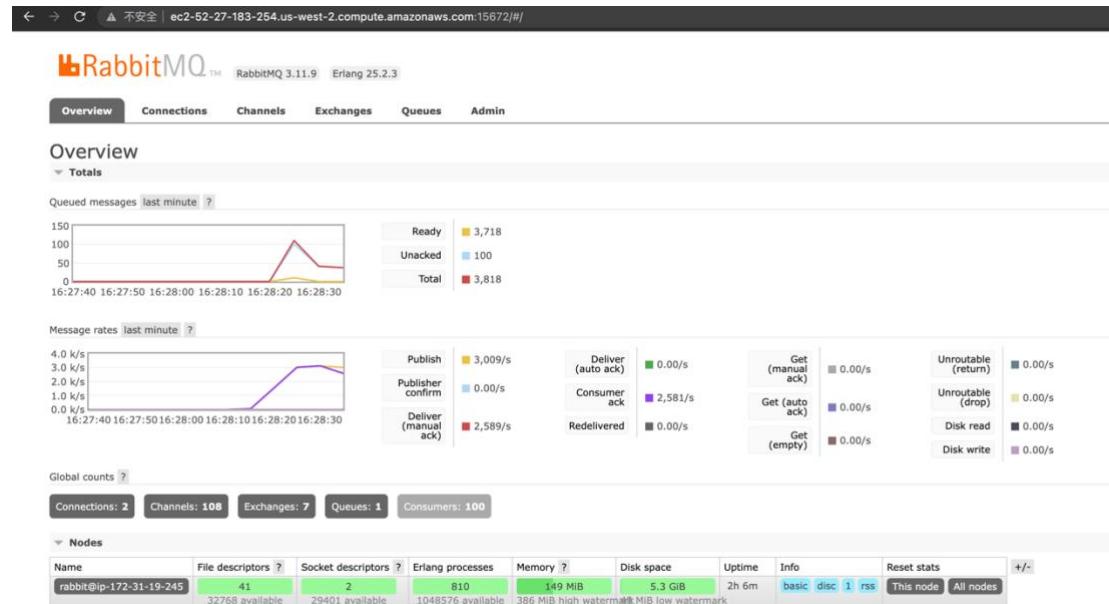
This user's like number is:
41

This user's dislike number is:
55

This user id has swiped right on:
[15283 761654 652694 951660 280281 314110 607168 434596 818046 738096 748096 4460442 536373 171704 211203 122822 417683 827244 253735 426852 626447 753647 317434 783441 141112 889994 285876 299811 880235 855847]

End

Sending 500k messages data in 100threads:
send/receive rates 3.05k messages/s



Overview

Totals

Queued messages last minute ?

Ready	5,886
Unacked	100
Total	5,986

Message rates last minute ?

Publish	2,753/s
Publisher confirm	0.0/s
Deliver (auto ack)	0.00/s
Consumer ack	1,441/s
Redelivered	0.00/s
(Get) (manual ack)	1,444/s
Get (auto ack)	0.00/s
Get (empty)	0.00/s
Get (return)	0.00/s
Unroutable (drop)	0.00/s
Disk read	0.00/s
Disk write	0.00/s

Global counts ?

Connections: 2 Channels: 108 Exchanges: 7 Queues: 1 Consumers: 100

Nodes

Name	File descriptors	Socket descriptors	Erlang processes	Memory	Disk space	Uptime	Info	Reset stats	+/-
rabbit@ip-172-31-19-245	41	2	810	164 MiB	5.3 GiB	2h 7m	basic disc 1 rss	This node All nodes	
	32768 available	29401 available	1048576 available	386 MiB high watermark	512 MiB low watermark				

Churn statistics

Ports and contexts

Overview

Totals

Queued messages last minute ?

Ready	0
Unacked	0
Total	0

Message rates last minute ?

Publish	0.00/s
Publisher confirm	0.00/s
Deliver (auto ack)	0.00/s
Consumer ack	0.00/s
Redelivered	0.00/s
(Get) (manual ack)	0.00/s
Get (auto ack)	0.00/s
Get (empty)	0.00/s
Get (return)	0.00/s
Unroutable (drop)	0.00/s
Disk read	0.00/s
Disk write	0.00/s

Global counts ?

Connections: 2 Channels: 108 Exchanges: 7 Queues: 1 Consumers: 100

Nodes

Name	File descriptors	Socket descriptors	Erlang processes	Memory	Disk space	Uptime	Info	Reset stats	+/-
rabbit@ip-172-31-19-245	41	2	810	154 MiB	5.3 GiB	2h 9m	basic disc 1 rss	This node All nodes	
	32768 available	29401 available	1048576 available	386 MiB high watermark	512 MiB low watermark				

Churn statistics

Ports and contexts

Export definitions

Import definitions

resources

```

> test
  target
    assignment1-client.iml
  pom.xml
External Libraries
Scratches and Consoles

```

```

27   queue = new LinkedBlockingQueue<>();
28   winCount = new AtomicInteger( initialValue: 0 );
29   loseCount = new AtomicInteger( initialValue: 0 );
30   CountDownLatch latch1 = new CountDownLatch(numThreads);
31   long start = System.currentTimeMillis();
32   BodyList bodyList = new BodyList(queue, totalCount);
33   bodyList.generateList();
34   ResultList resultList = new ResultList();
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66

```

This is the End

```

Number of successful requests:500000
Number of failed requests:0
Total lasting time: 179829
Throughput: 2793 requests/second
Mean time: 34.997726
Median response time: 31.0
99th percentile response time: 103.0
Max response time: 1751.0
Min response time: 13.0

```

Process finished with exit code 0

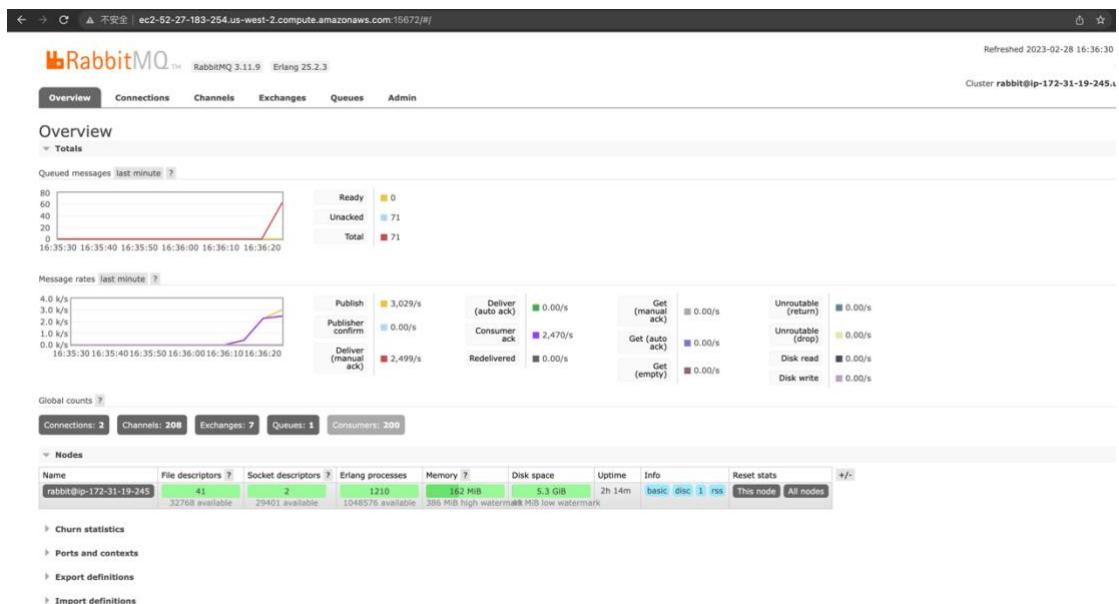
Run TODO Problems Profiler Terminal Build Dependencies

```

finished
Received: {"swiper": "4701", "swipee": "557855", "comment": "qe@hkeYoQmcuxXAIvEuTeTkqTxOWWDGogAKklspzYpdWStgYbtyFcukNhTdxgjdjBwDZNGoLoqfx#AnlICmyERNNwutvN0UqsLGGvDbONQgyrUltfqwiCLCuynkyqvzReLxLyCqmRQiEzMBUGr"
finished
Received: {"swiper": "3260", "swipee": "364907", "comment": "bvfKpYtYSotluBnrGqUcxEqHWEVsVlrJUqUzThQnUVGIYrnidpJpSnZGzaGR0PKJhonyDWqQiwWBfaVsNyBpjjiWWeGwxIoZmUBZKZhP0pRdSCoVonafzGFFJmVjWaxhgzzkkalzQjjnAkY"
finished
Received: {"swiper": "1623", "swipee": "501245", "comment": "wuxTSLt0VnhxJhsAaFMDYCLJTrLhsqdFsSDYqmPMaJdipKaagUrpBwvxEpEISkdGqcJfDeULwB0qlLw0JncDBMQuGKFQlVybFIgmxfeohnkloFdPxkAcVqhyVX1IBWgQCeJHYHNuvzR"
finished
Received: {"swiper": "2194", "swipee": "111308", "comment": "ZtwaxZqSxworyFvqFsfHZevLICiyeBsdUGBxdClhBJfrPYRFvvCR1EkZ00lettgdjpWRIAACfkwdVvcUQnJvtYTr0Cvq0hfyNBFcUhBchLxvSHswaFyzKdffHLqrvtZbtobACUnGikXlne"
finished
Received: {"swiper": "4158", "swipee": "23201", "comment": "VLphrFY2NyIAJfZqXgKfayKpdzqgeSsxCMwutuXAnFhXjupYcrw0uHwgJwpbTP1bnZvCldvRHGKtTxvrmSEwsPtnHpsSFNkXXZD0JhsazPhqXjbjzoC1bcPHrMpBwKeNdgXfsQZCBfr"
finished
Received: {"swiper": "1727", "swipee": "151159", "comment": "5QkzxTzcrZdgodbmpl0qcsBBHtrpQyvZKirunjzvPvInqabekBdsAymmRLIILuVXNDQGHNNQf1gkyoJMeEkvSljlHFOIY0oIPShtrzMeufjXixXM#rcGLR0ojM2qanAZUgbOX0dRTEbgwAPrjAna"

```

Sending 500k messages data in 200threads:
send/receive rates: 2.8k messages/s



不安全 | ec2-52-27-183-254.us-west-2.compute.amazonaws.com:15672/#/

RabbitMQ™ RabbitMQ 3.11.9 Erlang 25.2.3

Overview Connections Channels Exchanges Queues Admin

Overview

Totals

Queued messages last minute: ?

Ready: 0
Unacked: 55
Total: 55

Message rates last minute: ?

Publish	3,123/s	Deliver (auto ack)	0.00/s	Get (manual ack)	0.00/s	Unroutable (return)	0.00/s
Publisher confirm	0.00/s	Consumer ack	3,144/s	Get (auto ack)	0.00/s	Unroutable (drop)	0.00/s
Deliver (manual ack)	3,142/s	Redelivered	0.00/s	Get (empty)	0.00/s	Disk read	0.00/s

Global counts: ?

Connections: 2 Channels: 208 Exchanges: 7 Queues: 1 Consumers: 200

Nodes

Name	File descriptors ?	Socket descriptors ?	Erlang processes	Memory ?	Disk space	Uptime	Info	Reset stats	+/-
rabbit@ip-172-31-19-245	41 32768 available	2 29401 available	1210 1048576 available	173 MiB 386 MiB high watermark MiB low watermark	5.3 GiB	2h 15m	basic disc 1 rss	This node All nodes	+/-

Churn statistics
Ports and contexts
Export definitions

不安全 | ec2-52-27-183-254.us-west-2.compute.amazonaws.com:15672/#/

RabbitMQ™ RabbitMQ 3.11.9 Erlang 25.2.3

Overview Connections Channels Exchanges Queues Admin

Overview

Totals

Queued messages last minute: ?

Ready: 0
Unacked: 0
Total: 0

Message rates last minute: ?

Publish	0.00/s	Deliver (auto ack)	0.00/s	Get (manual ack)	0.00/s	Unroutable (return)	0.00/s
Publisher confirm	0.00/s	Consumer ack	0.00/s	Get (auto ack)	0.00/s	Unroutable (drop)	0.00/s
Deliver (manual ack)	0.00/s	Redelivered	0.00/s	Get (empty)	0.00/s	Disk read	0.00/s

Global counts: ?

Connections: 2 Channels: 208 Exchanges: 7 Queues: 1 Consumers: 200

Nodes

Name	File descriptors ?	Socket descriptors ?	Erlang processes	Memory ?	Disk space	Uptime	Info	Reset stats	+/-
rabbit@ip-172-31-19-245	41 32768 available	2 29401 available	1210 1048576 available	166 MiB 386 MiB high watermark MiB low watermark	5.3 GiB	2h 17m	basic disc 1 rss	This node All nodes	+/-

Churn statistics
Ports and contexts
Export definitions

```

Run: MultiThread x
86
89
90
90
90
90
91
This is the End
Number of successful requests:500000
Number of failed requests:0
Total lasting time: 173668
Throughput: 2890 requests/second
Mean time: 67.573606
Median response time: 63.0
99th percentile response time: 143.0
Max response time: 2442.0
Min response time: 13.0
Process finished with exit code 0

```

```

Finished
Received: {"swiper": "4604", "swipee": "356585", "comment": "zMovC1hBQfPjxqoCARvqaZBpWeHq1NsQ1cF01qtVyyiWgsncx1WJwzXnBxQJEtvwtbdhZ8IgkTExoa#xnB2kRiTzHHF0GUxhFXYLQ0kxBYVn1InzeTszb06FTriEsGJky6gRycdPnafnqnkkZtpQqRtENh
Finished
Received: {"swiper": "1375", "swipee": "555329", "comment": "i16GrbUfkffpnnaSetJhXj1AcaTzGpxxJmIRospfdDg1lkIQexzEdExwtbNXV0JgKSVytNPabuxnTMEdrqxCBtHjRnuwzunDrlyMfdAMvxMFaYnEcwCkqeCEPendstMUAreEk2YCeRGezjbFsxlxFVJtG
Finished
Received: {"swiper": "1666", "swipee": "167036", "comment": "#mvptDvtaHu2yTdcfLuMAnyCbjsVqPKXcvAcMtPxhuWaTqTzTOrulMSixhckjDrxJCbTmzJf1wLufqcvUsZmKamOHRoTmUFFohfQzQbLnYcEaK1AcQyGmTDpvcMFJ2t0
Finished
Received: {"swiper": "3696", "swipee": "619492", "comment": "FZUfHuayJguSxm0ZWTjJsf#RlqcBLq0jJfqDlzbzSy0d1jFVbcLippZJuitAtSpPqvPacsgDFDpmPnCtGnVzYQzVvYPhuZRSWuQJxjECJOPQJXQQTbIdoMic3jlybogeTrWEIjBLCDwtxRBp31NkXJYpzvaY
Finished
Received: {"swiper": "4450", "swipee": "334515", "comment": "qZkohbvZuexvnJgaKvBahtobrgfImm#auTotknTaqtrnnbyR0nVtfBxekTklhgZ1PJaunM1tKnhSrQuHptdyi1HnXZjOWwzPEyceZMwGHKGReentyRKMOGFkIpFcmt2lJS0gtTVBFBgRmALGOpvVwCvQy00
Finished
Received: {"swiper": "4307", "swipee": "424397", "comment": "DEZabgBYGozXxIy0m0UQvclfuUSdrtnJ0eUMMdNNryAtypnUHutrKCCHFXSIRfsnSnSwYJLTlxlcplzXkgdkW1CJLLTHKdEFsdriYMCuqBxXHDylkchalGLdthqQy0QzYRaisnFFNsntff00
Finished

```

Get result:

The one with 100 threads has the best performance.

Load balancer:

In the following part, I would show the create and the use of a load balancer.

First both two servers can pass the messages to RabbitMQ.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Put
server1	i-0178375f176da43	Running	t2.micro	2/2 checks passed	No alarms	+ us-west-2b	ec2-35-91-165-101.us-west-2.compute.amazonaws.com	55.1
RabbitMQ-Server	i-0677ae23ab5d85998	Running	t2.micro	2/2 checks passed	No alarms	+ us-west-2b	ec2-35-245-192-226.us-west-2.compute.amazonaws.com	54.1
abandoned.server	i-01641774ff0b77-2-f	Terminated	t2.micro	2/2 checks passed	No alarms	+ us-west-2b	-	-
server2	i-0250365a0dd9da89	Running	t2.micro	2/2 checks passed	No alarms	+ us-west-2b	ec2-35-93-49-23.us-west-2.compute.amazonaws.com	35.1

Create a load balancer:

Screenshot of the AWS EC2 Load Balancers page showing a single application load balancer named "a2-alb".

Name	DNS name	Status	VPC ID	Availability Zones	Type	Date created	Instance ID
a2-alb	a2-alb-1312944569.us-west-2.elb.amazonaws.com	Active	vpc-01a3369b1516ca171	4 Availability Zones	application	March 1, 2023, 18:25 (UTC-08:00)	-

Screenshot of the AWS EC2 Target Groups page showing a single target group named "a2-lb-group".

Name	ARN	Port	Protocol	Target type	Load balancer	VPC ID
a2-lb-group	arn:aws:elasticloadbalancing:us-west-2:277410911681:targetgroup/a2-lb-group/d871801d698aed6	8080	HTTP	Instance	a2-load-balancer	vpc-01a3369b1516ca171

```

shengguozhou@Shengguo-MacBook-Air Desktop % ls
5520mobile      a2-report.docx      cs5008
5610web         algorithm        cs5800
6650distributed_system assignment2-server test
LC              cs5001           test-6650-5610
Project         cs5002           -$-report.docx
a2-client-consumer cs5004           个人材料
shengguozhou@Shengguo-MacBook-Air Desktop % sudo scp -i 6650distributed_systems/hengguo_key.cer a2-client-consumer1/out/artifacts/consumer1_jar/consume* ec2-user@52.38.115.50:/home/ec2-user
Password: 
shengguozhou@Shengguo-MacBook-Air Desktop % ssh -i shengguo_key.cer ec2-user@52.38.115.50
Last login: Thu Mar  2 01:40:41 2023 from 24.35.95.190
[ec2-user@ip-172-31-25-235 ~]$ cd /home/ec2-user
[ec2-user@ip-172-31-25-235 ~]$ ls
[ec2-user@ip-172-31-25-235 ~]$ cd consumer1_jar
[ec2-user@ip-172-31-25-235 ~]$ rm consumer1.jar
[ec2-user@ip-172-31-25-235 ~]$ ls
[ec2-user@ip-172-31-25-235 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-31-25-235 ~]$ rm consumer1.jar
[ec2-user@ip-172-31-25-235 ~]$ ls
[ec2-user@ip-172-31-25-235 ~]$ exit
shengguozhou@Shengguo-MacBook-Air Desktop %

```

Consumer 1 is working on AWS EC2:

```
[ec2-user@ip-172-31-25-235 ~]$ ls  
consumer1.jar  
[ec2-user@ip-172-31-25-235 ~]$ java -jar /home/ec2-user/consumer1.jar  
no main manifest attribute, in /home/ec2-user/consumer1.jar  
[ec2-user@ip-172-31-25-235 ~]$ ls  
consumer1.jar  
[ec2-user@ip-172-31-25-235 ~]$ java -jar /home/ec2-user/consumer1.jar  
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".  
SLF4J: Defaulting to no-operation (NOP) logger implementation  
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details  
[  
Thread 98 waiting for messages.  
Thread 73 waiting for messages.  
Thread 32 waiting for messages.  
Thread 12 waiting for messages.  
Thread 31 waiting for messages.  
Thread 29 waiting for messages.  
Thread 28 waiting for messages.  
Thread 27 waiting for messages.  
Thread 26 waiting for messages.  
Thread 25 waiting for messages.  
Thread 24 waiting for messages.  
Thread 23 waiting for messages.  
Thread 22 waiting for messages.
```

Consumer2's .jar file is sent into the ec2 instance:

```
5520mobile      a2-report.docx      cs5008  
5610web        algorithm          cs5008  
6650distributed_system assignment2-server test  
LC             cs5001              test-6650-5610  
Project        cs5002              -$-report.docx  
a2-client-consumer cs5004              人材料  
shengguozhou@Shengguo-MacBook-Air Desktop % cd 6650distributed_system  
shengguozhou@Shengguo-MacBook-Air 6650distributed_system % chmod 500 shengguo_ke  
y.cer  
shengguozhou@Shengguo-MacBook-Air 6650distributed_system % sudo scp -i 6650dis  
tributed_system/shengguo_key.cer a2-client-consumer/consumer2/out/artifacts/consum  
er2_jar/consumer2.jar ec2-user@54.200.170.218:/home/ec2-user  
Password:  
Warning: Identity file 6650distributed_system/shengguo_key.cer not accessible: N  
o such file or directory.  
ec2-user@54.200.170.218: Permission denied (publickey,gssapi-keyex,gssapi-with-m  
ic).  
scp: Connection closed  
shengguozhou@Shengguo-MacBook-Air 6650distributed_system % cd ..  
shengguozhou@Shengguo-MacBook-Air Desktop % sudo scp -i 6650distributed_system/s  
hengguo_key.cer a2-client-consumer/consumer2/out/artifacts/consumer2_jar/consume  
r2.jar ec2-user@54.200.170.218:/home/ec2-user  
100% 5485KB  9.6MB/s  00:00  
shengguozhou@Shengguo-MacBook-Air ~ % ls  
Last login: Thu Mar  2 18:55:22 on ttys000  
shengguozhou@Shengguo-MacBook-Air ~ % cd Desktop  
shengguozhou@Shengguo-MacBook-Air Desktop % cd 6650distributed_system  
shengguozhou@Shengguo-MacBook-Air 6650distributed_system % ssh -i shengguo.key.c  
er ec2-user@54.200.170.218  
Last login: Thu Mar  2 18:51:57 2023 from 24.35.95.190  
-l ( -- )  Amazon Linux 2 AMI  
--|---|---|  
https://aws.amazon.com/amazon-linux-2/  
-bash: warning: setlocale: LC_CTYPE: cannot change locale (UTF-8): No such file  
or directory  
[ec2-user@ip-172-31-31-213 ~]$ ls  
consumer2.jar  
[ec2-user@ip-172-31-31-213 ~]$
```

Postman works well

The screenshot shows the Postman interface with the following details:

- Request URL:** POST http://34.222.146.112:8080/server-a2_war/skiers/swipe/left
- Body (JSON):**

```
1  "swipee": "123",  
2  "swiping": "sdvfa",  
3  "comment": "gffewefejbuvmk",  
4  "leftOrRight": "left"
```
- Response:** Status: 201 Created, Time: 993 ms, Size: 314 B
- Response Body (Pretty):**

```
1 Inside processRequest  
2 {  
3     "swipee": "123",  
4     "swiping": "sdvfa",  
5     "comment": "gffewefejbuvmk",  
6     "leftOrRight": "left"  
7 }  
8 Assignment 2 doPost is working.
```

Pass .war file into another server

```

/var/root/.ssh/known_hosts:8: ec2-52-13-110-185.us-west-2.compute.amazonaws.com
/var/root/.ssh/known_hosts:9: ec2-54-202-16-181.us-west-2.compute.amazonaws.com
/var/root/.ssh/known_hosts:10: ec2-52-13-16-49.us-west-2.compute.amazonaws.com
(2 additional names omitted)
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '34.222.146.112' (ED25519) to the list of known hosts.
server-a2.war.war      100% 4997KB  9.4MB/s  00:00
shengguo@Shengguo-MacBook-Air:~/Desktop$ sudo scp -i /home/shengguo/.ssh/id_rsa_ec2 user@54.213.154.109:/usr/share/tomcat/webapps/
shengguo.key.cer assignment2-server/out/artifacts/server_a2_war/server-a2.war.war
Password:
The authenticity of host '54.213.154.109 (54.213.154.109)' can't be established.
ED25519 key fingerprint is SHA256:3BACBqU1Vtu8804JmEJMUHjYMT4gDeSN+AjlLlJNo.
This host key is known by the following other names/addresses:
 /var/root/.ssh/known_hosts:18: 35.93.49.23
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '54.213.154.109' (ED25519) to the list of known hosts.
server-a2.war.war      100% 4997KB  8.8MB/s  00:00
shengguo@Shengguo-MacBook-Air:~/Desktop %

```

The screenshot shows the Postman interface with a POST request to `http://54.213.154.109:8080/server-a2.war/skiers/swipe/left`. The Body tab displays a JSON payload:

```

{
  "swiper": "123",
  "swipee": "sdvfd",
  "comment": "gtifwefwefjbjuvnk",
  "leftOrRight": "left"
}

```

The response status is 201 Created, with a response body showing the processed data:

```

{
  "Inside processRequest": [
    {
      "swiper": "123",
      "swipee": "sdvfd",
      "comment": "gtifwefwefjbjuvnk",
      "leftOrRight": "left"
    }
  ],
  "Assignment 2 doPost is working."
}

```

The screenshot shows the AWS EC2 Instances page. The left sidebar shows the navigation menu. The main table lists instances, with `server2` selected. The detailed view for `server2` shows the following information:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 address	Private IPv4 addresses
server1	i-01783875f317d4a43	Running	t2.micro	2/2 checks passed	No alarms	+ us-west-2b	ec2-54-222-146-112.us-west-2.compute.amazonaws.com	54.222	172.31.21.39
RabbitMQ-Server	i-0677ae23a5d85998	Running	t2.micro	2/2 checks passed	No alarms	+ us-west-2b	ec2-54-202-0-80.us-west-2.compute.amazonaws.com	54.202	
consumer1	i-0982162e1da4653e	Running	t2.micro	2/2 checks passed	No alarms	+ us-west-2b	ec2-54-202-81-188.us-west-2.compute.amazonaws.com	54.202	
consumer2	i-0531c1d0f0e146a4	Running	t2.micro	2/2 checks passed	No alarms	+ us-west-2b	ec2-54-200-170-218.us-west-2.compute.amazonaws.com	54.200	
server2	i-0250365a00dd9da89	Running	t2.micro	2/2 checks passed	No alarms	+ us-west-2b	ec2-54-213-154-109.us-west-2.compute.amazonaws.com	54.213	

Before using the load balancer:

```

Received: ("swiper": "1270", "swipee": "57744", "comment": "ZjQggDLNkUdwfTMWijvOdwMMfcktldvpc
oPlsqRSGoWaJnxzRdrQyopeTaTpxwPLuPfZookO2KunFMxeTiiIiweiFnhVDbzGobbywFLxFErQJdysVsOfJHnw
tQkdfKfB18MapNdNmDgoUymBmhkuCjxyGNLfeuWfCTOBqkGsuhJVTIwyAenNvpFoKfZHZYbMgNMOTguFWfuyd
uEcOpJYfQwAtQixpJuaufhzNTNvKSjvKmnNoutDSHjIfwQORyKD7tVsnnbwQdZWG1", "leftOrRight": "left")
Finished
Received: ("swiper": "1298", "swipee": "swipee", "comment": "ZjQggDLNkUdwfTMWijvOdwMMfcktldvpc
oPlsqRSGoWaJnxzRdrQyopeTaTpxwPLuPfZookO2KunFMxeTiiIiweiFnhVDbzGobbywFLxFErQJdysVsOfJHnw
tQkdfKfB18MapNdNmDgoUymBmhkuCjxyGNLfeuWfCTOBqkGsuhJVTIwyAenNvpFoKfZHZYbMgNMOTguFWfuyd
uEcOpJYfQwAtQixpJuaufhzNTNvKSjvKmnNoutDSHjIfwQORyKD7tVsnnbwQdZWG1", "leftOrRight": "right")
Finished
Received: ("swiper": "1298", "swipee": "swipee", "comment": "ZjQggDLNkUdwfTMWijvOdwMMfcktldvpc
oPlsqRSGoWaJnxzRdrQyopeTaTpxwPLuPfZookO2KunFMxeTiiIiweiFnhVDbzGobbywFLxFErQJdysVsOfJHnw
tQkdfKfB18MapNdNmDgoUymBmhkuCjxyGNLfeuWfCTOBqkGsuhJVTIwyAenNvpFoKfZHZYbMgNMOTguFWfuyd
uEcOpJYfQwAtQixpJuaufhzNTNvKSjvKmnNoutDSHjIfwQORyKD7tVsnnbwQdZWG1", "leftOrRight": "right")
Finished
Received: ("swiper": "1298", "swipee": "311131", "comment": "ZjQggDLNkUdwfTMWijvOdwMMfcktldvpc
oPlsqRSGoWaJnxzRdrQyopeTaTpxwPLuPfZookO2KunFMxeTiiIiweiFnhVDbzGobbywFLxFErQJdysVsOfJHnw
tQkdfKfB18MapNdNmDgoUymBmhkuCjxyGNLfeuWfCTOBqkGsuhJVTIwyAenNvpFoKfZHZYbMgNMOTguFWfuyd
uEcOpJYfQwAtQixpJuaufhzNTNvKSjvKmnNoutDSHjIfwQORyKD7tVsnnbwQdZWG1", "leftOrRight": "right")
Finished
Received: ("swiper": "1298", "swipee": "311131", "comment": "ZjQggDLNkUdwfTMWijvOdwMMfcktldvpc
oPlsqRSGoWaJnxzRdrQyopeTaTpxwPLuPfZookO2KunFMxeTiiIiweiFnhVDbzGobbywFLxFErQJdysVsOfJHnw
tQkdfKfB18MapNdNmDgoUymBmhkuCjxyGNLfeuWfCTOBqkGsuhJVTIwyAenNvpFoKfZHZYbMgNMOTguFWfuyd
uEcOpJYfQwAtQixpJuaufhzNTNvKSjvKmnNoutDSHjIfwQORyKD7tVsnnbwQdZWG1", "leftOrRight": "right")
Finished
Received: ("swiper": "1459", "swipee": "265448", "comment": "KbEnCacAghDbHwzbyNfxrOwx
IsyxUHBYflizSFOKwhzGUNGzCQbxKyhuwXSwPGwyxocCpxHyCYJuLhWHPiD0HSWPsllxSuWfxok
dUMZhtArTHDdqxFALH13cAyK0SpFZFPqAxlUEHHCpBaQxKuWeMu1tVidwpcKFjyiqPJaITfmvp
vJHARJHxZChqMsMXvrdDNKLZPFxTpqySovNvowGnazy1hCnKirkkoYBmcokyQ1G8ej1d1CkPZQwo
cJHItahgmlVGONGNmwnqBws0LQWehihKgdMzUvickoDDQPeJgdVsxCUMNAoX3QtSkb", "leftOrRight": "right")
Finished
Received: ("swiper": "1459", "swipee": "265448", "comment": "KbEnCacAghDbHwzbyNfxrOwx
IsyxUHBYflizSFOKwhzGUNGzCQbxKyhuwXSwPGwyxocCpxHyCYJuLhWHPiD0HSWPsllxSuWfxok
dUMZhtArTHDdqxFALH13cAyK0SpFZFPqAxlUEHHCpBaQxKuWeMu1tVidwpcKFjyiqPJaITfmvp
vJHARJHxZChqMsMXvrdDNKLZPFxTpqySovNvowGnazy1hCnKirkkoYBmcokyQ1G8ej1d1CkPZQwo
cJHItahgmlVGONGNmwnqBws0LQWehihKgdMzUvickoDDQPeJgdVsxCUMNAoX3QtSkb", "leftOrRight": "left")
Finished
Received: ("swiper": "1459", "swipee": "981430", "comment": "YTbKtJvuTDFHvkjfssOqPHW
AaBhvRLZLxzumMnRgmZifzCtclybdUJyBynd0LNzefmGHzUWQnaFRNDRuXmrOrGayiaIRIV
twovWzWnkkeDtmoyQomBe6ivRGcxZvVcwitEljGpduxz1luXzveriksIfmtaTdAZJ0CEKwsmtXG
SbxgzVKPDEngbThfuQjGzH1boPYMGjaBucFXrXVqamChWgLgHQEZUJCQjCwRkRGMMFx", "leftOrRight": "left")
Finished
Received: ("swiper": "1459", "swipee": "981430", "comment": "YTbKtJvuTDFHvkjfssOqPHW
AaBhvRLZLxzumMnRgmZifzCtclybdUJyBynd0LNzefmGHzUWQnaFRNDRuXmrOrGayiaIRIV
twovWzWnkkeDtmoyQomBe6ivRGcxZvVcwitEljGpduxz1luXzveriksIfmtaTdAZJ0CEKwsmtXG
SbxgzVKPDEngbThfuQjGzH1boPYMGjaBucFXrXVqamChWgLgHQEZUJCQjCwRkRGMMFx", "leftOrRight": "right")
Finished
Received: ("swiper": "1459", "swipee": "1981430", "comment": "ZjQggDLNkUdwfTMWijvOdwMMfcktldvpc
oPlsqRSGoWaJnxzRdrQyopeTaTpxwPLuPfZookO2KunFMxeTiiIiweiFnhVDbzGobbywFLxFErQJdysVsOfJHnw
tQkdfKfB18MapNdNmDgoUymBmhkuCjxyGNLfeuWfCTOBqkGsuhJVTIwyAenNvpFoKfZHZYbMgNMOTguFWfuyd
uEcOpJYfQwAtQixpJuaufhzNTNvKSjvKmnNoutDSHjIfwQORyKD7tVsnnbwQdZWG1", "leftOrRight": "left")
Finished
Received: ("swiper": "1459", "swipee": "1981430", "comment": "ZjQggDLNkUdwfTMWijvOdwMMfcktldvpc
oPlsqRSGoWaJnxzRdrQyopeTaTpxwPLuPfZookO2KunFMxeTiiIiweiFnhVDbzGobbywFLxFErQJdysVsOfJHnw
tQkdfKfB18MapNdNmDgoUymBmhkuCjxyGNLfeuWfCTOBqkGsuhJVTIwyAenNvpFoKfZHZYbMgNMOTguFWfuyd
uEcOpJYfQwAtQixpJuaufhzNTNvKSjvKmnNoutDSHjIfwQORyKD7tVsnnbwQdZWG1", "leftOrRight": "right")
Finished
Received: ("swiper": "1459", "swipee": "283434", "comment": "jUTURtxvEuAVseohfxwppbprFTYARfrQ
VewuHncaJnVzWzTlNluyNPTNRckuSmRcyN", "leftOrRight": "left")
Finished

```

← → ⌂ ▲ 不安全 | 54.202.0.80:15672/#

RabbitMQ™ RabbitMQ 3.11.9 Erlang 25.2.3

Refreshed 2s Cluster rabbit@i

- [Overview](#)
- [Connections](#)
- [Channels](#)
- [Exchanges](#)
- [Queues](#)
- [Admin](#)

Overview

Totals

Queued messages last minute: ?

Ready: 0
Unacked: 31
Total: 31

Message rates last minute: ?

Category	Value
Publisher	2,823/s
Publisher confirm	0.00/s
Deliver (manual ack)	5,629/s
Consumer ack	5,629/s
Redelivered	0.00/s
Get (manual ack)	0.00/s
Get (auto ack)	0.00/s
Get (empty)	0.00/s
Unroutable (return)	0.00/s
Unroutable (drop)	0.00/s
Disk read	0.00/s
Disk write	0.00/s

Global counts: ?

Connections: 5 Channels: 316 Exchanges: 8 Queues: 2 Consumers: 300

Nodes

Name	File descriptors	Socket descriptors	Erlang processes	Memory	Disk space	Uptime	Info	Reset stats	+/-
rabbit@ip-172-31-19-245	44	5	1663	212 MiB	5.3 GiB	2h 29m	basic disc 1 rss	This node All nodes	+/-
	32768 available	29401 available	1048576 available	386 MiB high watermark	386 MiB low watermark				

- Churn statistics
- Ports and contexts
- Event definitions

← → ⌂ ▲ 不安全 | 54.202.0.80:15672/#

RabbitMQ™ RabbitMQ 3.11.9 Erlang 25.2.3

- [Overview](#)
- [Connections](#)
- [Channels](#)
- [Exchanges](#)
- [Queues](#)
- [Admin](#)

Overview

Totals

Queued messages last minute: ?

Ready: 0
Unacked: 56
Total: 56

Message rates last minute: ?

Category	Value
Publisher	2,905/s
Publisher confirm	0.00/s
Deliver (manual ack)	5,841/s
Consumer ack	5,842/s
Redelivered	0.00/s
Get (manual ack)	0.00/s
Get (auto ack)	0.00/s
Get (empty)	0.00/s
Unroutable (return)	0.00/s
Unroutable (drop)	0.00/s
Disk read	0.00/s
Disk write	0.00/s

Global counts: ?

Connections: 5 Channels: 316 Exchanges: 8 Queues: 2 Consumers: 300

Nodes

Name	File descriptors	Socket descriptors	Erlang processes	Memory	Disk space	Uptime	Info	Reset stats	+/-
rabbit@ip-172-31-19-245	44	5	1663	213 MiB	5.3 GiB	2h 30m	basic disc 1 rss	This node All nodes	+/-
	32768 available	29401 available	1048576 available	386 MiB high watermark	386 MiB low watermark				

← → C ▲ 不安全 | 54.202.0.80:15672/#/

RabbitMQ™ RabbitMQ 3.11.9 Erlang 25.2.3

- [Overview](#)
- [Connections](#)
- [Channels](#)
- [Exchanges](#)
- [Queues](#)
- [Admin](#)

Overview

▼ Totals

Queued messages last minute ?

Ready	0
Unacked	30
Total	30

12:46:10 12:46:20 12:46:30 12:46:40 12:46:50 12:47:00

Message rates last minute ?

Publish	2,935/s
Publisher confirm	0.00/s
Deliver (auto ack)	0.00/s
Consumer ack	5,871/s
Redelivered	0.00/s
Get (manual ack)	0.00/s
Get (auto ack)	0.00/s
Get (empty)	0.00/s
Unroutable (return)	0.00/s
Unroutable (drop)	0.00/s
Disk read	0.00/s
Disk write	0.00/s

12:46:10 12:46:20 12:46:30 12:46:40 12:46:50 12:47:00

Global counts ?

Connections: 5 Channels: 316 Exchanges: 8 Queues: 2 Consumers: 300

▼ Nodes

Name	File descriptors ?	Socket descriptors ?	Erlang processes	Memory ?	Disk space	Uptime	Info	Reset stats
rabbit@ip-172-31-19-245	44 32768 available	5 29401 available	1663 1048576 available	213 MB 386 MB high watermark MiB low watermark	5.3 GB	2h 31m	basic disc 1 rss	This node All nodes

▶ Churn statistics

← → C ▲ 不安全 | 54.202.0.80:15672/#/

RabbitMQ™ RabbitMQ 3.11.9 Erlang 25.2.3

Refreshed 2023-03-14 Cluster [rabbit@ip-172-31-19-245](#)

- [Overview](#)
- [Connections](#)
- [Channels](#)
- [Exchanges](#)
- [Queues](#)
- [Admin](#)

Overview

▼ Totals

Queued messages last minute ?

Ready	0
Unacked	0
Total	0

12:47:00 12:47:10 12:47:20 12:47:30 12:47:40 12:47:50

Message rates last minute ?

Publish	0.00/s
Publisher confirm	0.00/s
Deliver (auto ack)	0.00/s
Consumer ack	0.00/s
Redelivered	0.00/s
Get (manual ack)	0.00/s
Get (auto ack)	0.00/s
Get (empty)	0.00/s
Unroutable (return)	0.00/s
Unroutable (drop)	0.00/s
Disk read	0.00/s
Disk write	0.00/s

12:47:00 12:47:10 12:47:20 12:47:30 12:47:40 12:47:50

Global counts ?

Connections: 5 Channels: 316 Exchanges: 8 Queues: 2 Consumers: 300

▼ Nodes

Name	File descriptors ?	Socket descriptors ?	Erlang processes	Memory ?	Disk space	Uptime	Info	Reset stats
rabbit@ip-172-31-19-245	44 32768 available	5 29401 available	1663 1048576 available	199 MB 386 MB high watermark MiB low watermark	5.3 GB	2h 32m	basic disc 1 rss	This node All nodes

▶ Churn statistics

▶ Ports and contexts

▶ Export definitions

```
KOQBZyxHAEAMESFHwMwwrnJjrgz0BeFwLMJyQgSQaEEdvaiYIwtetYJdyQ1sAPGrPlCbcuygSvPLwC
Received: {"swiper": "1874", "swipee": "953765", "comment": "TwkrifiOHMzrsgpLAUpBpqlaATnOglundz7e
apWQjtsPMHTAFYYjjEvVEGUzirwXhQsaFlp10KCNgYtIuvyBhCSQRWg1z0hncInf
femzjSwNfdGccpVflwkghyV0uailrcBcllUMBMGMG1jPoxuYttifStOEBrhPvPsifcZOR", "leftOrRight": "leftOrRight": "left"}
Finished
Received: {"swiper": "1874", "swipee": "953765", "comment": "TwkrifiOHMzrsgpLAUpBpqlaATnOglundz7e
apWQjtsPMHTAFYYjjEvVEGUzirwXhQsaFlp10KCNgYtIuvyBhCSQRWg1z0hncInf
femzjSwNfdGccpVflwkghyV0uailrcBcllUMBMGMG1jPoxuYttifStOEBrhPvPsifcZOR", "leftOrRight": "leftOrRight": "left"}
Finished
Received: {"swiper": "4280", "swipee": "98794", "comment": "LarjzWmR1ZBkhnmdPAEvUBw
K7QDPLVGZmgntaYSZUFPyTewDTCjivBrqGontTFFGEZYTakhDmwhaiVxxZmZxChselVfpPhz0lluw
FgAlflosVYtsZRTSLaIutbpojGTBuSlucSOQiyTzcWbzQPUycadSp1efdqtib1i1JvhJicxtdTfRFA
lpFBwgilmPQhuFHzFvnfQbeuQejVoxFvRAWizWCXzbwHkyQFbxvzTQSqyJgvNqTdsby", "leftOrRight": "leftOrRight": "right"}
Finished
Received: {"swiper": "796", "swipee": "434872", "comment": "XAKU0teAXxEEMfgjB0pUSMDvu
K7QDPLVGZmgntaYSZUFPyTewDTCjivBrqGontTFFGEZYTakhDmwhaiVxxZmZxChselVfpPhz0lluw
FgAlflosVYtsZRTSLaIutbpojGTBuSlucSOQiyTzcWbzQPUycadSp1efdqtib1i1JvhJicxtdTfRFA
lpFBwgilmPQhuFHzFvnfQbeuQejVoxFvRAWizWCXzbwHkyQFbxvzTQSqyJgvNqTdsby", "leftOrRight": "leftOrRight": "right"}
Finished
Received: {"swiper": "147140", "swipee": "857", "comment": "zwebeIdMwMTDqySrhmrvUMDE
cnylnYLcpPznewEGrDck1kbRnprnhElbBhMBApNjuYAUGHjhrsbdPxPnPFCMPTwQd0cKHYjiUxHs
kkwbOPMjgPUYQZyaPfKIUkmuolgrYSSNFgzsjs0zdBzDgEqgqCaLtaOxtgzIwoxwacCZCDJruS
USawBLoBnzymYrBAKYQmoGjwvDVyeEqzgTSfwmGccYaYWFHMeD0Rin0wyMtdvg0", "leftOrRight": "leftOrRight": "left"}
Finished
Received: {"swiper": "147140", "swipee": "857", "comment": "zwebeIdMwMTDqySrhmrvUMDE
cnylnYLcpPznewEGrDck1kbRnprnhElbBhMBApNjuYAUGHjhrsbdPxPnPFCMPTwQd0cKHYjiUxHs
kkwbOPMjgPUYQZyaPfKIUkmuolgrYSSNFgzsjs0zdBzDgEqgqCaLtaOxtgzIwoxwacCZCDJruS
USawBLoBnzymYrBAKYQmoGjwvDVyeEqzgTSfwmGccYaYWFHMeD0Rin0wyMtdvg0", "leftOrRight": "leftOrRight": "left"}
Finished
Received: {"swiper": "147140", "swipee": "857", "comment": "zwebeIdMwMTDqySrhmrvUMDE
cnylnYLcpPznewEGrDck1kbRnprnhElbBhMBApNjuYAUGHjhrsbdPxPnPFCMPTwQd0cKHYjiUxHs
kkwbOPMjgPUYQZyaPfKIUkmuolgrYSSNFgzsjs0zdBzDgEqgqCaLtaOxtgzIwoxwacCZCDJruS
USawBLoBnzymYrBAKYQmoGjwvDVyeEqzgTSfwmGccYaYWFHMeD0Rin0wyMtdvg0", "leftOrRight": "leftOrRight": "left"}

```

```

import java.io.IOException;
import java.util.Collections;
import java.util.List;
import java.util.concurrent.CountDownLatch;
import java.util.concurrent.BlockingQueue;
import java.util.concurrent.LinkedBlockingQueue;
import java.util.concurrent.atomic.AtomicInteger;

public class MultiThread {
    private static String IP;
    private static BlockingQueue<SwipeEvent> queue;
    private static AtomicInteger winCount;
    private static AtomicInteger loseCount;
    private static Integer numThreads = 100;
    private static Integer totalCount = 500_000;

    public static void main(String[] args) throws InterruptedException {
        System.out.println("Start from here");

        // IP = "http://localhost:8080/server-a2-war/skiers/";
        IP = "http://34.222.112.8088/server-a2-war/skiers/";
        // IP = "http://a2-load-balancer-1376876592.us-west-2.elb.amazonaws.com:8080/server-a2-war
        queue = new LinkedBlockingQueue<>();
        winCount = new AtomicInteger( initialValue: 0 );
        loseCount = new AtomicInteger( initialValue: 0 );
        CountDownLatch latch1 = new CountDownLatch(numThreads);
        long start = System.currentTimeMillis();

        for (int i = 0; i < numThreads; i++) {
            Thread thread = new Thread(new Runnable() {
                public void run() {
                    try {
                        while (true) {
                            SwipeEvent event = queue.take();
                            if (event.isWin) {
                                winCount.incrementAndGet();
                            } else {
                                loseCount.incrementAndGet();
                            }
                        }
                    } catch (InterruptedException e) {
                        e.printStackTrace();
                    }
                }
            });
            thread.start();
        }

        latch1.await();
        long end = System.currentTimeMillis();
        System.out.println("Total lasting time: " + (end - start));
        System.out.println("Throughput: " + ((totalCount / (end - start)) * 1000) + " requests/second");
        System.out.println("Mean time: " + (double) (end - start) / numThreads);
        System.out.println("Median response time: " + (double) (end - start) / numThreads);
        System.out.println("99th percentile response time: " + (double) (end - start) / numThreads);
        System.out.println("Max response time: " + (double) (end - start) / numThreads);
        System.out.println("Min response time: " + (double) (end - start) / numThreads);
    }
}

```

The speed of sending data is around 6k messages/s

2nd time test:

```

import java.io.IOException;
import java.util.Collections;
import java.util.List;
import java.util.concurrent.CountDownLatch;
import java.util.concurrent.BlockingQueue;
import java.util.concurrent.LinkedBlockingQueue;
import java.util.concurrent.atomic.AtomicInteger;

public class MultiThread {
    private static String IP;
    private static BlockingQueue<SwipeEvent> queue;
    private static AtomicInteger winCount;
    private static AtomicInteger loseCount;
    private static Integer numThreads = 100;
    private static Integer totalCount = 500_000;

    public static void main(String[] args) throws InterruptedException {
        System.out.println("Start from here");

        // IP = "http://localhost:8080/server-a2-war/skiers/";
        IP = "http://34.222.112.8088/server-a2-war/skiers/";
        // IP = "http://a2-load-balancer-1376876592.us-west-2.elb.amazonaws.com:8080/server-a2-war
        queue = new LinkedBlockingQueue<>();
        winCount = new AtomicInteger( initialValue: 0 );
        loseCount = new AtomicInteger( initialValue: 0 );
        CountDownLatch latch1 = new CountDownLatch(numThreads);
        long start = System.currentTimeMillis();

        for (int i = 0; i < numThreads; i++) {
            Thread thread = new Thread(new Runnable() {
                public void run() {
                    try {
                        while (true) {
                            SwipeEvent event = queue.take();
                            if (event.isWin) {
                                winCount.incrementAndGet();
                            } else {
                                loseCount.incrementAndGet();
                            }
                        }
                    } catch (InterruptedException e) {
                        e.printStackTrace();
                    }
                }
            });
            thread.start();
        }

        latch1.await();
        long end = System.currentTimeMillis();
        System.out.println("Total lasting time: " + (end - start));
        System.out.println("Throughput: " + ((totalCount / (end - start)) * 1000) + " requests/second");
        System.out.println("Mean time: " + (double) (end - start) / numThreads);
        System.out.println("Median response time: " + (double) (end - start) / numThreads);
        System.out.println("99th percentile response time: " + (double) (end - start) / numThreads);
        System.out.println("Max response time: " + (double) (end - start) / numThreads);
        System.out.println("Min response time: " + (double) (end - start) / numThreads);
    }
}

```

← → ⌂ 不安全 | 54.202.0.80:15672/#/ Refr

RabbitMQ™ RabbitMQ 3.11.9 Erlang 25.2.3 Cluster r

- [Overview](#)
- [Connections](#)
- [Channels](#)
- [Exchanges](#)
- [Queues](#)
- [Admin](#)

Overview

▼ Totals

Queued messages last minute ?

Ready	0
Unacked	66
Total	66

Message rates last minute ?

Publish	2,947/s
Publisher confirm	0.00/s
Deliver (auto ack)	0.00/s
Consumer ack	5,950/s
Redelivered	0.00/s
Get (manual ack)	0.00/s
Get (auto ack)	0.00/s
Get (empty)	0.00/s
Unroutable (return)	0.00/s
Unroutable (drop)	0.00/s
Disk read	0.00/s
Disk write	0.00/s

Global counts ?

Connections: 5 Channels: 316 Exchanges: 8 Queues: 2 Consumers: 300

▼ Nodes

Name	File descriptors ?	Socket descriptors ?	Erlang processes	Memory ?	Disk space	Uptime	Info	Reset stats	+/-
rabbit@ip-172-31-19-245	44 32768 available	5 29401 available	1663 1048576 available	205 MiB 386 MiB high watermark	5.3 GiB MiB low watermark	1h 57m	basic disc 1 rss	This node All nodes	+/-

» Churn statistics

» Ports and contexts

← → ⌂ 不安全 | 54.202.0.80:15672/#/ Refreshed 2023- Cluster rabbit@ip-1:

RabbitMQ™ RabbitMQ 3.11.9 Erlang 25.2.3

- [Overview](#)
- [Connections](#)
- [Channels](#)
- [Exchanges](#)
- [Queues](#)
- [Admin](#)

Overview

▼ Totals

Queued messages last minute ?

Ready	0
Unacked	104
Total	104

Message rates last minute ?

Publish	2,807/s
Publisher confirm	0.00/s
Deliver (auto ack)	0.00/s
Consumer ack	5,886/s
Redelivered	0.00/s
Get (manual ack)	5,881/s
Get (auto ack)	0.00/s
Get (empty)	0.00/s
Unroutable (return)	0.00/s
Unroutable (drop)	0.00/s
Disk read	0.00/s
Disk write	0.00/s

Global counts ?

Connections: 5 Channels: 316 Exchanges: 8 Queues: 2 Consumers: 300

▼ Nodes

Name	File descriptors ?	Socket descriptors ?	Erlang processes	Memory ?	Disk space	Uptime	Info	Reset stats	+/-
rabbit@ip-172-31-19-245	44 32768 available	5 29401 available	1663 1048576 available	206 MiB 386 MiB high watermark	5.3 GiB MiB low watermark	1h 57m	basic disc 1 rss	This node All nodes	+/-

» Churn statistics

» Ports and contexts

← → ⌂ 不安全 | 54.202.0.80:15672/#/

RabbitMQ™ RabbitMQ 3.11.9 Erlang 25.2.3

Overview Connections Channels Exchanges Queues Admin

Overview

▼ Totals

Queued messages last minute ?

Ready	0
Unacked	0
Total	0

Message rates last minute ?

Publish	0.00/s
Publisher confirm	0.00/s
Deliver (auto ack)	0.00/s
Consumer ack	0.00/s
Redelivered	0.00/s
Get (manual ack)	0.00/s
Get (auto ack)	0.00/s
Get (empty)	0.00/s
Unroutable (return)	0.00/s
Unroutable (drop)	0.00/s
Disk read	0.00/s
Disk write	0.00/s

Global counts ?

Connections: 5 Channels: 316 Exchanges: 8 Queues: 2 Consumers: 300

▼ Nodes

Name	File descriptors ?	Socket descriptors ?	Erlang processes	Memory ?	Disk space	Uptime	Info	Reset stats	+/-
rabbit@ip-172-31-19-245	44 32768 available	5 29401 available	1663 1048576 available	201 MiB 386 MiB high watermark	5.3 GiB 8 MiB low watermark	1h 59m	basic disc 1 rss	This node All nodes	+/-

▶ Churn statistics

▶ Ports and contexts

← → ⌂ 不安全 | 54.202.0.80:15672/#/

RabbitMQ™ RabbitMQ 3.11.9 Erlang 25.2.3

Overview Connections Channels Exchanges Queues Admin

Overview

▼ Totals

Queued messages last minute ?

Ready	0
Unacked	0
Total	0

Message rates last minute ?

Publish	0.00/s
Publisher confirm	0.00/s
Deliver (auto ack)	0.00/s
Consumer ack	0.00/s
Redelivered	0.00/s
Get (manual ack)	0.00/s
Get (auto ack)	0.00/s
Get (empty)	0.00/s
Unroutable (return)	0.00/s
Unroutable (drop)	0.00/s
Disk read	0.00/s
Disk write	0.00/s

Global counts ?

Connections: 5 Channels: 316 Exchanges: 8 Queues: 2 Consumers: 300

▼ Nodes

Name	File descriptors ?	Socket descriptors ?	Erlang processes	Memory ?	Disk space	Uptime	Info	Reset stats	+/-
rabbit@ip-172-31-19-245	44 32768 available	5 29401 available	1663 1048576 available	193 MiB 386 MiB high watermark	5.3 GiB 8 MiB low watermark	1h 59m	basic disc 1 rss	This node All nodes	+/-

▶ Churn statistics

▶ Ports and contexts

```
import java.io.IOException;
import java.util.Collections;
import java.util.List;
import java.util.concurrent.CountDownLatch;
import java.util.concurrent.BlockingQueue;
import java.util.concurrent.LinkedBlockingQueue;
import java.util.concurrent.atomic.AtomicInteger;

public class MultiThread {
    private static String IP;
    private static BlockingQueue<SwipeEvent> queue;
    private static AtomicInteger winCount;
    private static AtomicInteger loseCount;
    private static Integer numThreads = 100;
    private static Integer totalCount = 500_000;

    public static void main(String[] args) throws InterruptedException {
        System.out.println("Start from here");
        // IP = "http://localhost:8888/server-a2-war/skiers/";
        IP = "http://34.222.166.112:8888/server-a2-war/skiers/";
        // IP = "http://a2-load-balancer-176870592.us-west-2.elb.amazonaws.com:8888/server-a2-war/skiers/";
        queue = new LinkedBlockingQueue<>();
        winCount = new AtomicInteger( initialValue: 0 );
        loseCount = new AtomicInteger( initialValue: 0 );
        CountDownLatch latch1 = new CountDownLatch(numThreads);
        long start = System.currentTimeMillis();
    }
}
```

This is the End
Number of successful requests:500000
Number of failed requests:0
Total lasting time: 179750
Throughput: 2793 requests/second
Mean time: 35.041014
Median response time: 33.0
99th percentile response time: 86.0
Max response time: 1856.0
Min response time: 14.0

Still the same, the speed of sending messages is around 6k messgaes /s ,
and the number of messages inside the RabbitMQ is rather in a stable performance.

After using the load balancer:

Set client's sending messages' IP address to load balancer's DNS IP address

```
import java.io.IOException;
import java.util.Collections;
import java.util.List;
import java.util.concurrent.CountDownLatch;
import java.util.concurrent.BlockingQueue;
import java.util.concurrent.LinkedBlockingQueue;
import java.util.concurrent.atomic.AtomicInteger;

public class MultiThread {
    private static String IP;
    private static BlockingQueue<SwipeEvent> queue;
    private static AtomicInteger winCount;
    private static AtomicInteger loseCount;
    private static Integer numThreads = 100;
    private static Integer totalCount = 500_000;

    public static void main(String[] args) throws InterruptedException {
        System.out.println("Start from here");
        // IP = "http://localhost:8888/server-a2-war/skiers/";
        IP = "http://a2-44-244-204-294.us-west-2.compute.amazonaws.com:8888/server-a2-war/skiers/";
        // IP = "http://a2-load-balancer-176870592.us-west-2.elb.amazonaws.com:8888/server-a2-war/skiers/";
        queue = new LinkedBlockingQueue<>();
        winCount = new AtomicInteger( initialValue: 0 );
        loseCount = new AtomicInteger( initialValue: 0 );
        CountDownLatch latch1 = new CountDownLatch(numThreads);
        long start = System.currentTimeMillis();
        BodyList bodyList = new BodyList(queue, totalCount);
        bodyList.generateList();
        ResultList resultList = new ResultList();
        for (int i = 0; i < numThreads; i++) {
            RunInOneThread r1 = new RunInOneThread(IP, queue, winCount, loseCount, latch1, resultList);
            Thread thread1 = new Thread(r1);
            thread1.start();
            System.out.println(i);
        }
    }
}
```

Tested on AWS EC2:

```
③ ⓘ 6660distributed_system -- ec2-user@ip-172-31-25-228:~ - ssh -i shengguo_key.pem ec2-user@i-04.202.81.68 - 80v2s
i1stKEzOcPcgIktkWNeypAUPlVs1oWStK1lqEHzZiaAuOpHFTf1tBBZSxxecUzrtEBumDBXTyPcMcQ
BlzQNinCHoAkMfjyFZKMuRPyfzMoOKnaXlOxvdiFL0zibtrvcUdBNuYv0oVOSsRdHvrn0if
sZ3bdowarWjRdkRHczblZyOjVhG1Baauoo1aGoeQejtcuYYV1Qo1wmpdbdmVflxBY", "leftOrRight": "left"
Finished
Received: {"swiper": "881", "swipee": "446423", "comment": "UiA DmgyWlUxvZkSiicGVttzz
TTyTwSMfdLZNQkuUgwyjkiQxVpoxHMPexNFDhJ00PBjLkPwDUzbzqAMCudfLkHwGi1AxMxxPXR
PBjgppJRCQDgg0PybwjsiqMBAbXVpJGIttyJMLhwSllogHupXvChndeul1jBGMCsXsLnZRzmvoCB
NFoWj1KTCkzwIVJQEpucIFRxInBdwZ1EhEtJKXUWsyLSZUzhLsSRyWlbgjRxTMIQSds", "leftOrRight": "left"
Finished
Received: {"swiper": "881", "swipee": "446423", "comment": "UiA DmgyWlUxvZkSiicGVttzz
TTyTwSMfdLZNQkuUgwyjkiQxVpoxHMPexNFDhJ00PBjLkPwDUzbzqAMCudfLkHwGi1AxMxxPXR
PBjgppJRCQDgg0PybwjsiqMBAbXVpJGIttyJMLhwSllogHupXvChndeul1jBGMCsXsLnZRzmvoCB
NFoWj1KTCkzwIVJQEpucIFRxInBdwZ1EhEtJKXUWsyLSZUzhLsSRyWlbgjRxTMIQSds", "leftOrRight": "left"
Finished
Received: {"swiper": "2321", "swipee": "781075", "comment": "ZhaNmneIk5sVdx8mRntFsxEW
ucxJkWxOsyscGwv5Ec1ygKhCpAUykuTAAxXHkIABoahprssBwoYBMyXhVpmnKPxw0Qxrz
jSBYKBEtvsCkpaBfgFvu0WejPLGDnbkjcWeSwAnJiedwHECqUsaqNIEUowXsudkTQoAWhrBdqHWo
dJTFQd0KJZog1tUPH02MKmHmkLucWYkltLlpxBcbt1pjiiefZauTx1FoYzjlkzTzahnalyokyB
klMzojTfjqhcBpcGJD1YtLdYhkwBH2ozlbyLTTHYBVnNMFOloQeKtdprhNSCDQVwWuD", "leftOrRight": "left"
Finished
Received: {"swiper": "1267", "swipee": "829126", "comment": "GwDKDjAhazMowwLjKTjSiPh
sdD9kBeTvsCkpaBfgFvu0WejPLGDnbkjcWeSwAnJiedwHECqUsaqNIEUowXsudkTQoAWhrBdqHWo
dJTFQd0KJZog1tUPH02MKmHmkLucWYkltLlpxBcbt1pjiiefZauTx1FoYzjlkzTzahnalyokyB
CksLAxgeBpHypngElfmszCkgThzqzlyihNO", "leftOrRight": "right"
Finished
Received: {"swiper": "3539", "swipee": "45330", "comment": "tBKerVdJgAwXewReVUHnsQZh
sKtA7zDiexzPdTJpyXapoxghbokSPaABOhGalYVmbQu1vBfipiYgFOVctijlrCxnHyeZUJANjNjh1YspC
oktMQGYkwcCcyFgvIVtUkxzaldbQpkvxb", "leftOrRight": "right"
Finished
Received: {"swiper": "1481", "swipee": "681656", "comment": "JzNj0iXUQ1GyMbCMzWtou1LeSgsaDzC
oiewMkjuiWfCpNCfxTdZDbnLEYwPodQjnn1tQVyzZwkapphGlgyaqZLwhwnNnjbPKdbkpYhKay1v1
ahQdvknCatGnNpxwHtzE02jCffjshJ0Dyx1bpPraHeu1MAXbCpnJrsGyrD0usYKV0tTsYQxdwNndr
sAY10AH0ndrxEpi1VFiptFsd1Pkw", "leftOrRight": "right"
Finished
Received: {"swiper": "1877", "swipee": "91828", "comment": "EHbhCkdWYrBpEqeJhpjZXSPrbLkWdu
wETflgjDRAbCz1gixTaUcTCHQsUveHkakjzyeoBto8FpkcdUSparetsAdSmQhqfj3Pktjja1iHfY
skATzDiexzPdTJpyXapoxghbokSPaABOhGalYVmbQu1vBfipiYgFOVctijlrCxnHyeZUJANjNjh1YspC
oktMQGYkwcCcyFgvIVtUkxzaldbQpkvxb", "leftOrRight": "right"
Finished
Received: {"swiper": "324", "swipee": "370008", "comment": "xAHbOoDjDjhjvhrrcfnrwbrfzHzHRAqn
Kdd1RecljAC21KxARCSUsUkunzYyFFKvGvcgpfadCccbqvYd1QnwHpeoQYXPsDZ1GAtzeyD3KnqLnduBrpY
Ni9TysqYL0SofnphJUySyfTqgbJHejtjsHrXMHgQwaTCLvkADUbMzutXNODGeAvVzdECsofQOOAxtZs
FDstDf0YKjkrabGftHubcCmeSaQtVte1iGwZisenEgulfQ0QhPfbstHMzczQhwvxdMbkZmWrdSsCpcat
MMClwWhRwMpklluOyf1uELthremzicubLneTDMiBmluMIXRv1t1vXSMewvraYClnRjPc1EDv1uEfizmsNukh
```

◀ ▶ ⚡ 不安全 | 54.202.8.0:15672/#/

RabbitMQ™ RabbitMQ 3.11.9 Erlang 25.2.3

- [Overview](#)
- [Connections](#)
- [Channels](#)
- [Exchanges](#)
- [Queues](#)
- [Admin](#)

Overview

▼ Totals

Queued messages last minute ?

Ready	0
Unacked	92
Total	92

Message rates last minute ?

Publish	3,518/s
Publisher confirm	0.00/s
Deliver (manual ack)	7,139/s
Consumer ack	0.00/s
Redelivered	0.00/s
Get (manual ack)	0.00/s
Get (auto ack)	0.00/s
Get (empty)	0.00/s
Unrouteable (return)	0.00/s
Unrouteable (drop)	0.00/s
Disk read	0.00/s
Disk write	0.00/s

Global counts ?

Connections: 5 Channels: 316 Exchanges: 8 Queues: 2 Consumers: 300

▼ Nodes

Name	File descriptors ?	Socket descriptors ?	Erlang processes	Memory ?	Disk space	Uptime	Info	Reset stats	+/-
rabbit@ip-172-31-19-245	44	5	1663	205 MiB	5.3 GiB	2h 33m	basic disc 1 rss	This node All nodes	+/-
	32768 available	29401 available	1048576 available	386 MiB high watermark	5 MiB low watermark				

▶ Churn statistics

◀ ▶ ⚡ 不安全 | 54.202.8.0:15672/#/

RabbitMQ™ RabbitMQ 3.11.9 Erlang 25.2.3

Refreshed Cluster rabbit

- [Overview](#)
- [Connections](#)
- [Channels](#)
- [Exchanges](#)
- [Queues](#)
- [Admin](#)

Overview

▼ Totals

Queued messages last minute ?

Ready	0
Unacked	55
Total	55

Message rates last minute ?

Publish	3,604/s
Publisher confirm	0.00/s
Deliver (auto ack)	7,240/s
Consumer ack	0.00/s
Redelivered	0.00/s
Get (auto ack)	0.00/s
Get (empty)	0.00/s
Unrouteable (return)	0.00/s
Unrouteable (drop)	0.00/s
Disk read	0.00/s
Disk write	0.00/s

Global counts ?

Connections: 5 Channels: 316 Exchanges: 8 Queues: 2 Consumers: 300

▼ Nodes

Name	File descriptors ?	Socket descriptors ?	Erlang processes	Memory ?	Disk space	Uptime	Info	Reset stats	+/-
rabbit@ip-172-31-19-245	44	5	1663	213 MiB	5.3 GiB	2h 34m	basic disc 1 rss	This node All nodes	+/-
	32768 available	29401 available	1048576 available	386 MiB high watermark	5 MiB low watermark				

▶ Churn statistics

▶ Ports and contexts

▶ Export definitions

▶ Import definitions

[HTTP API](#) [Server Docs](#) [Tutorials](#) [Community Support](#) [Community Slack](#) [Commercial Support](#) [Plugins](#) [GitHub](#) [Changelog](#)

<http://54.202.0.80:15672/#/>

RabbitMQ™ RabbitMQ 3.11.9 Erlang 25.2.3

Overview

Totals

Queued messages last minute ?

Ready: 0
Unacked: 0
Total: 0

Message rates last minute ?

Operation	Rate
Publish	0.00/s
Publisher confirm	0.00/s
Deliver (auto ack)	0.00/s
Consumer ack	0.00/s
Deliver (manual ack)	0.00/s
Redelivered	0.00/s
Get (manual ack)	0.00/s
Get (auto ack)	0.00/s
Get (empty)	0.00/s
Unroutable (return)	0.00/s
Unroutable (drop)	0.00/s
Disk read	0.00/s
Disk write	0.00/s

Global counts ?

Connections: 5 Channels: 316 Exchanges: 8 Queues: 2 Consumers: 300

Nodes

Name	File descriptors	Socket descriptors	Erlang processes	Memory	Disk space	Uptime	Info	Reset stats
rabbit@ip-172-31-19-245	44	5	1663	194 MB	5.3 GB	2h 36m	basic disc 1 rss	This node All nodes

Churn statistics

Ports and contexts

Project ~/Desktop/a2-client-consumer/client

src

- client
- .idea
- src
 - main
 - java
 - event
 - GraphPlot
 - OutputCSV
 - Result
 - ResultList
 - ResultProcess
 - SwipeEvent
 - BodyList
 - GenerateSingleBody
 - MultiThread
 - relations.png
 - relations.uml
 - RunInOneThread
 - resources
- test
- target
- assignment1-client.iml
- pom.xml

External Libraries

Scratches and Consoles

MultiThread.java

```

import java.io.IOException;
import java.util.Collections;
import java.util.List;
import java.util.concurrent.CountDownLatch;
import java.util.concurrent.BlockingQueue;
import java.util.concurrent.LinkedBlockingQueue;
import java.util.concurrent.atomic.AtomicInteger;

public class MultiThread {
    private static String IP;
    private static BlockingQueue<SwipeEvent> queue;
    private static AtomicInteger winCount;
    private static AtomicInteger loseCount;
    private static Integer numThreads = 100;
    private static Integer totalCount = 500_000;

    public static void main(String[] args) throws InterruptedException {
        System.out.println("Start from here");
        IP = "http://localhost:8080/server-a2-war/skiers/";
        // IP = "http://34.222.146.112:8080/server-a2-war/skiers/";
        IP = "http://a2-load-balancer-1376870592.us-west-2.elb.amazonaws.com:8080/server-a2-war/skiers";
        queue = new LinkedBlockingQueue<>();
        winCount = new AtomicInteger( initialValue: 0 );
        loseCount = new AtomicInteger( initialValue: 0 );
        CountDownLatch latch1 = new CountDownLatch(numThreads);
        long start = System.currentTimeMillis();
    }
}

```

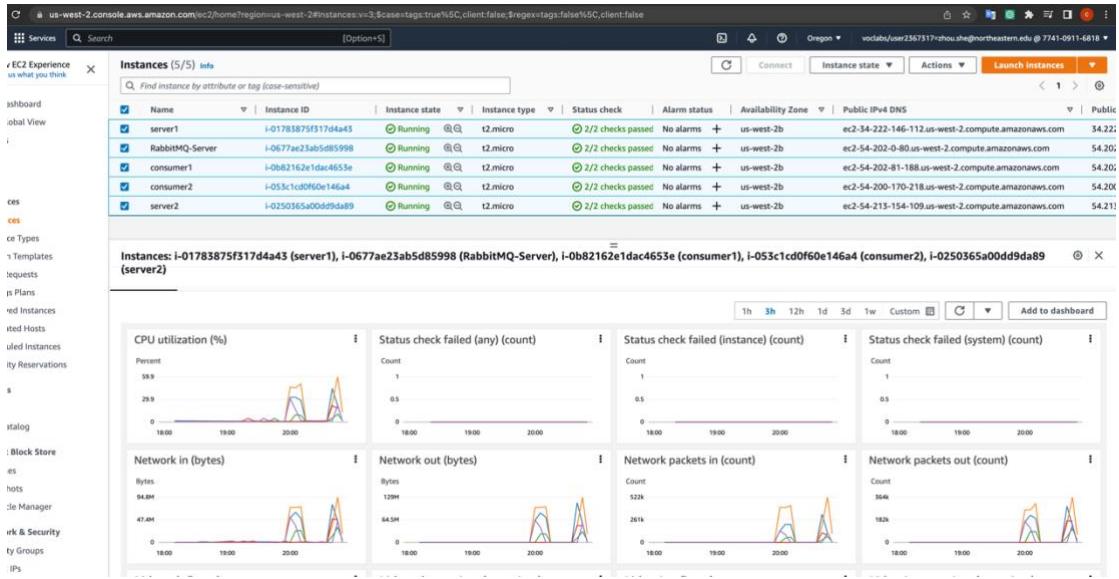
run: MultiThread

```

57
59
59
This is the End
Number of successful requests:500000
Number of failed requests:0
Total lasting time: 152364
Throughput: 3289 requests/second
Mean time: 29.512364
Median response time: 27.0
99th percentile response time: 72.0
Max response time: 1548.0
Min response time: 14.0

```

Process finished with exit code 0



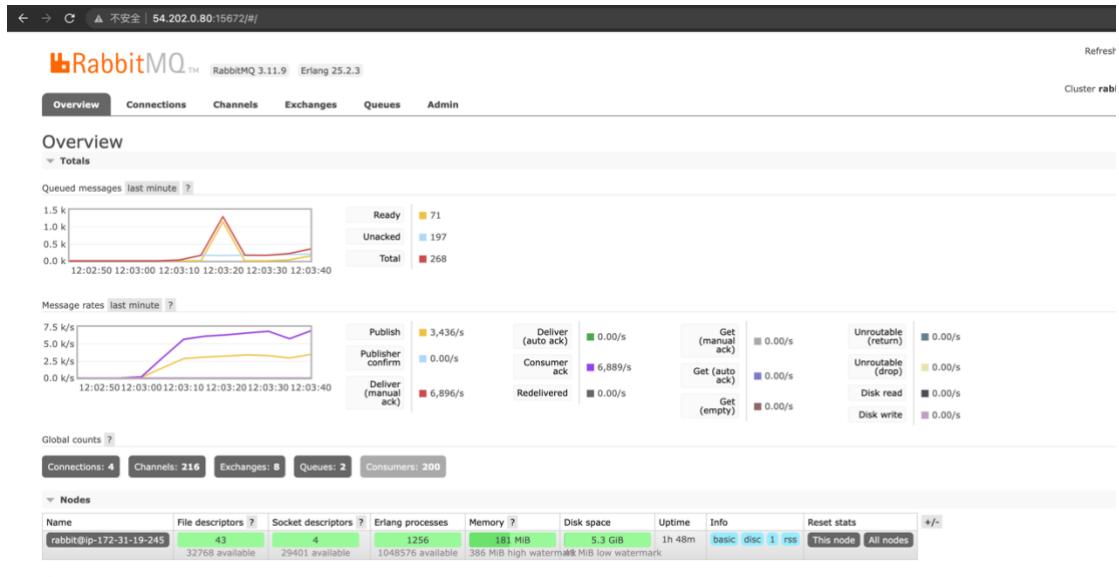
And the result shows that it is faster, the average speed of sending messages is around: 7.5k messages /s, improved by 25%.

And sometimes there would be more messages inside the RabbitMQ, because the server is faster, but the two consumers are slow compared with the relative speed of server, so in some few seconds, the messages would be blocked inside the queue.

But because of the load balancer, the speed of server does grow.

2nd time test:

On AWS EC2:



← → ⌂ 不安全 | 54.202.0.80:15672/#/

RabbitMQ™ RabbitMQ 3.11.9 Erlang 25.2.3

Overview Connections Channels Exchanges Queues Admin

Overview

▼ Totals

Queued messages last minute ?

Ready	15
Unacked	200
Total	215

Message rates last minute ?

Publish	3,287/s
Publisher confirm	0.00/s
Deliver (auto ack)	0.00/s
Consumer ack	6,600/s
Redelivered	0.00/s
Get (manual ack)	0.00/s
Get (auto ack)	0.00/s
Get (empty)	0.00/s
Unroutable (return)	0.00/s
Unroutable (drop)	0.00/s
Disk read	0.00/s
Disk write	0.00/s

Global counts ?

Connections: 4 Channels: 216 Exchanges: 8 Queues: 2 Consumers: 200

▼ Nodes

Name	File descriptors ?	Socket descriptors ?	Erlang processes	Memory ?	Disk space	Uptime	Info	Reset stats	+/-
rabbit@ip-172-31-19-245	43 32768 available	4 29401 available	1256 1048576 available	177 MiB 386 MiB high watermark MiB low watermark	5.3 GiB	1h 48m	basic disc 1 rss	This node All nodes	+/-

▶ Churn statistics

▶ Ports and contexts

▶ Export definitions

▶ Import definitions

← → ⌂ 不安全 | 54.202.0.80:15672/#/

RabbitMQ™ RabbitMQ 3.11.9 Erlang 25.2.3

Overview Connections Channels Exchanges Queues Admin

Overview

▼ Totals

Queued messages last minute ?

Ready	0
Unacked	172
Total	172

Message rates last minute ?

Publish	3,227/s
Publisher confirm	0.00/s
Deliver (auto ack)	0.00/s
Consumer ack	6,870/s
Redelivered	0.00/s
Get (manual ack)	0.00/s
Get (auto ack)	0.00/s
Get (empty)	0.00/s
Unroutable (return)	0.00/s
Unroutable (drop)	0.00/s
Disk read	0.00/s
Disk write	0.00/s

Global counts ?

Connections: 4 Channels: 216 Exchanges: 8 Queues: 2 Consumers: 200

▼ Nodes

Name	File descriptors ?	Socket descriptors ?	Erlang processes	Memory ?	Disk space	Uptime	Info	Reset stats	+/-
rabbit@ip-172-31-19-245	43 32768 available	4 29401 available	1256 1048576 available	183 MiB 386 MiB high watermark MiB low watermark	5.3 GiB	1h 48m	basic disc 1 rss	This node All nodes	+/-

▶ Churn statistics

▶ Ports and contexts

◀ ▶ ⌂ 不安全 | 54.202.0.80:15672/#/ Refreshed 2023-03-02 1 Cluster rabbit@ip-172-31-19-245

RabbitMQ™ RabbitMQ 3.11.9 Erlang 25.2.3 Overview

Totals

Queued messages last minute ?

Ready	586
Unacked	200
Total	786

Message rates last minute ?

Publish	3,646/s
Publisher confirm	0.00/s
Deliver (auto ack)	0.00/s
Consumer ack	7,303/s
Redelivered	0.00/s
Get (manual ack)	0.00/s
Get (auto ack)	0.00/s
Get (empty)	0.00/s
Unroutable (return)	0.00/s
Unroutable (drop)	0.00/s
Disk read	0.00/s
Disk write	0.00/s

Global counts ?

Connections: 4 Channels: 216 Exchanges: 8 Queues: 2 Consumers: 200

Nodes

Name	File descriptors	Socket descriptors	Erlang processes	Memory	Disk space	Uptime	Info	Reset stats
rabbit@ip-172-31-19-245	43	4	1256	178 MB	5.3 GB	1h 49m	basic disc 1 rss	+/- This node All nodes
	32768 available	29401 available	1048576 available	386 MiB high watermark	386 MiB low watermark			

Churn statistics

Ports and contexts

Export definitions

◀ ▶ ⌂ 不安全 | 54.202.0.80:15672/#/ Refreshed 2023-03-02 1 Cluster ra

RabbitMQ™ RabbitMQ 3.11.9 Erlang 25.2.3 Overview

Totals

Queued messages last minute ?

Ready	10,773
Unacked	181
Total	10,954

Message rates last minute ?

Publish	3,482/s
Publisher confirm	0.00/s
Deliver (auto ack)	0.00/s
Consumer ack	5,626/s
Redelivered	0.00/s
Get (manual ack)	0.00/s
Get (auto ack)	0.00/s
Get (empty)	0.00/s
Unroutable (return)	0.00/s
Unroutable (drop)	0.00/s
Disk read	0.00/s
Disk write	0.00/s

Global counts ?

Connections: 4 Channels: 216 Exchanges: 8 Queues: 2 Consumers: 200

Nodes

Name	File descriptors	Socket descriptors	Erlang processes	Memory	Disk space	Uptime	Info	Reset stats
rabbit@ip-172-31-19-245	43	4	1256	247 MiB	5.3 GiB	1h 50m	basic disc 1 rss	+/- This node All nodes
	32768 available	29401 available	1048576 available	386 MiB high watermark	386 MiB low watermark			

Churn statistics

Ports and contexts

Export definitions

Tunnel definitions

不安全 | 54.202.0.80:15672/#/

RabbitMQ

RabbitMQ 3.11.9 Erlang 25.2.3

- Overview
- Connections
- Channels
- Exchanges
- Queues
- Admin

Overview

Totals

Queued messages last minute ?

Ready	0
Unacked	0
Total	0

Message rates last minute ?

Publish	0.00/s
Publisher confirm	0.00/s
Deliver (manual ack)	0.00/s
Deliver (auto ack)	0.00/s
Consumer ack	0.00/s
Redelivered	0.00/s
Get (empty)	0.00/s
Get (manual ack)	0.00/s
Get (auto ack)	0.00/s
Unroutable (drop)	0.00/s
Unroutable (return)	0.00/s
Disk read	0.00/s
Disk write	0.00/s

Global counts ?

Connections: 4 Channels: 216 Exchanges: 8 Queues: 2 Consumers: 200

Nodes

Name	File descriptors	Socket descriptors	Erlang processes	Memory	Disk space	Uptime	Info	Reset stats
rabbit@ip-172-31-19-245	43	4	1256	167 MB	5.3 GB	1h 50m	basic disc 1 rss	+/- This node All nodes
	32768 available	29401 available	1048576 available	386 MB high watermark	512 MB low watermark			

Churn statistics

src main java MultiThread main

Project

client ~/Desktop/a2-client-consumer/client

.idea

src

main

java

event

GraphPlot

OutputCSV

Result

ResultList

ResultProcess

SwipeEvent

BodyList

GenerateSingleBody

MultiThread

relations.png

relations.uml

RunInOneThread

resources

test

target

assignment1-client.iml

pom.xml

External Libraries

MultiThread

```

src/main/java/MultiThread.java
6 import java.io.IOException;
7 import java.util.Collections;
8 import java.util.List;
9 import java.util.concurrent.CountDownLatch;
10 import java.util.concurrent.BlockingQueue;
11 import java.util.concurrent.LinkedBlockingQueue;
12 import java.util.concurrent.atomic.AtomicInteger;
13
14 public class MultiThread {
15     private static String IP;
16     private static BlockingQueue<SwipeEvent> queue;
17     private static AtomicInteger winCount;
18     private static AtomicInteger loseCount;
19     private static Integer numThreads = 100;
20     private static Integer totalCount = 500_000;
21
22     public static void main(String[] args) throws InterruptedException {
23         System.out.println("Start from here");
24
25         IP = "http://localhost:8080/server-a2-war/skiers/";
26         // IP = "http://34.222.146.112:8080/server-a2-war/skiers/";
27         // IP = "http://a2-load-balancer-1376870592.us-west-2.elb.amazonaws.com:8080/server-a2-war/skiers";
    }
    
```

73
90
66
72
72
73
67
70

67

This is the End

Number of successful requests:500000

Number of failed requests:0

Total lasting time: 156801

Throughput: 3205 requests/second

Mean time: 30.436458

Median response time: 28.0

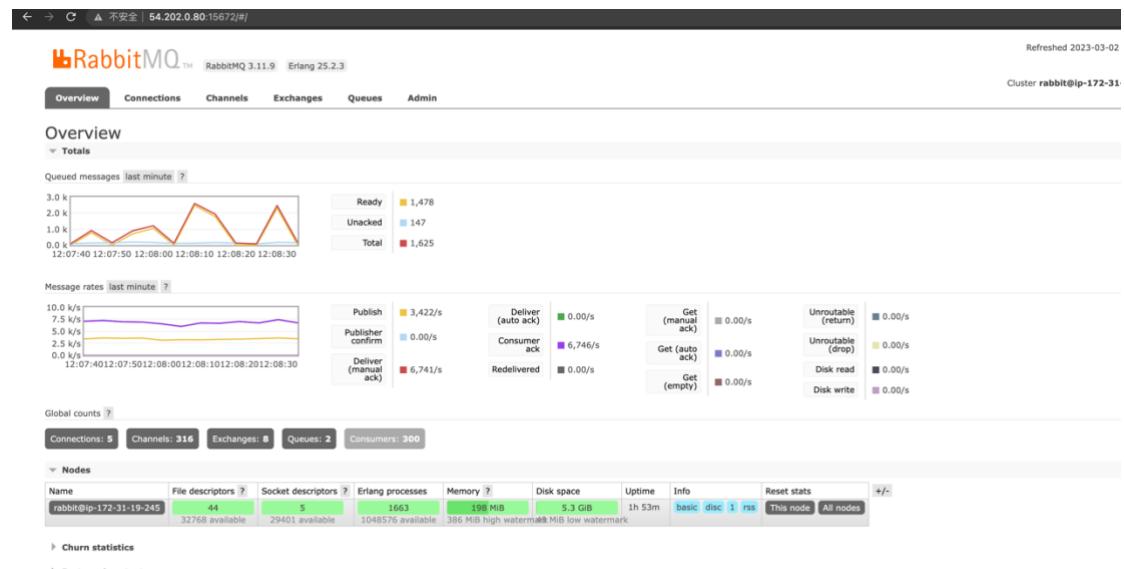
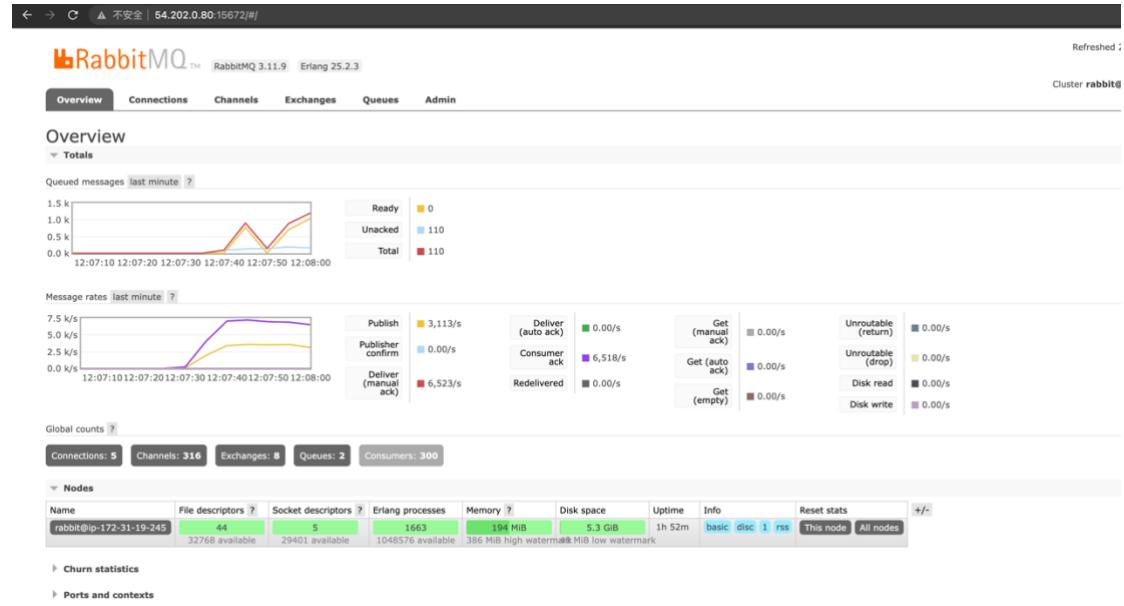
99th percentile response time: 88.0

Max response time: 2007.0

Min response time: 14.0

Process finished with exit code 0

2nd time test on AWS EC2:



<http://54.202.0.80:15672/#/>

RabbitMQ™ RabbitMQ 3.11.9 Erlang 25.2.3

Overview Connections Channels Exchanges Queues Admin

Overview

Totals

Queued messages last minute ?

Ready: 0
Unacked: 0
Total: 0

Message rates last minute ?

Publish: 0.00/s	Deliver (auto ack): 0.00/s	Get (manual ack): 0.00/s	Unroutable (return): 0.00/s
Publisher confirm: 0.00/s	Consumer ack: 0.00/s	Get (auto ack): 0.00/s	Unroutable (drop): 0.00/s
Deliver (manual ack): 0.00/s	Redelivered: 0.00/s	Get (empty): 0.00/s	Disk read: 0.00/s

Global counts ?

Connections: 5 Channels: 316 Exchanges: 8 Queues: 2 Consumers: 300

Nodes

Name	File descriptors ?	Socket descriptors ?	Erlang processes	Memory ?	Disk space	Uptime	Info	Reset stats
rabbit@(p-172-31-19-245)	44 32768 available	5 29401 available	1663 1048576 available	184 MB 386 MB high watermark 8 MB low watermark	5.3 GB	1h 54m	basic disc 1 rss	+/- This node All nodes

Churn statistics
Ports and contexts
Export definitions
Import definitions

Client: http://localhost:15672/api/clients/assignment1-client

```

> .idea
> src
  > main
    > java
      > event
        > GraphPlot
        > OutputCSV
        > Result
        > ResultList
        > ResultProcess
        > SwipeEvent
        > BodyList
        > GenerateSingleBody
        > MultiThread
        > relations.png
        > relations.uml
        > RunOnOneThread
      > resources
    > test
> target
  & assignment1-client.iml
  pom.xml
  & External libraries
run: MultiThread
  34
  34
  31
  32
  21
  43
  55
  57
This is the End
Number of successful requests:500000
Number of failed requests:0
Total lasting time: 154983
Throughput: 3246 requests/second
Mean time: 29.948938
Median response time: 27.0
99th percentile response time: 75.0
Max response time: 2044.0
Min response time: 14.0
  
```

Process finished with exit code 0

Git Run TODO Problems Profiler Terminal Build Dependencies

Result messages in consumer:

```
○ ● ■ 6650distributed_system - ec2-user@ip-172-31-25-235:~ -- ssh -i shengguo_key.cer ec2-user@54.202.81.188 - 80x24
yvjmtNfdvFhHLyvLKhfeBpvcrhMvgICPQQqXJBd1zKxAsLphjhawaRmmIAanntWZUUfcfSqdDuNJwM1h0
pWsDaBFkthDXiGmthRaYhGT0gcJdKJ0OFWdMLfnxDQxxHsJzoGvkMTQqfhPlgIDcrXtaZgvQbpDnHgjQ
NjMgGftEYzyAOgXIVLSgqfnLQRUpDUjxmIfhoDrjIeMMAHMzSyAehzLcwTCmSiZHoiHpxifP", "leftOrRight": "right"}
Finished
Received: {"swiper": "2027", "swipee": "201595", "comment": "IIIsPSkKOGLaorkOGKUR0bx
IrMdMNkhwUZnOTAgroVacZqPAdedcPzdJfvLRaQdgcgizoORnFOmULQTieEPAAvpIIQGiXaYtSupEip
HiiWkeXiVXcyDKqkiAOJNHY1DCVCLXUteJozG1YwTJBRenVzrXdzClxydAbBneGEiUIYsHZasHtlgXVY
jMNqwOoJAszSyiJ0txQDaGebNcdNWlckoLhhdYgLXEWnpIXbaasbLMFJvgrCJucdThSqkyY", "leftOrRight": "right"}
Finished
Received: {"swiper": "1645", "swipee": "763066", "comment": "yEDiatddOutJruIgAYLkjmm
lxPaAkHJTfkBJisfTbBRQBujNeIEeTKukiNUhupyFmILigQuQuxtRGZHZmYsKQHXzLQYHWtdXNqCmjed
RaBpmhhTpRbkjLBVtmPKoFBdlHQmucGAjlupMEDsqOfwBUiZUDwUuvynZfqrdWFQGWBXYykJEVSZKUr
XjMSmuxtUFRPzNmNiMWQAjctMcymRsXsYxrtJCTAEtCePB0XLJuxQneFNHAOnOnMNuYNk", "leftOrRight": "left"}
Finished
Received: {"swiper": "4253", "swipee": "26631", "comment": "FbaXTwAguLyRHDPPVFmBEeguI
XXEQHqCJWLajdGZdszHdsEHQRqhpYKsuIeAUAGBdofQQjBWipSyqsjGAEUazbRoLAU1RkrDPIvPRhsu
cRkSrHVBXiwWDycqIJQzaWJXDPMNYQZNQKOP0xoECGgluktnfHGhfJWMuHEqMkIyzfmkuLTAoqhffqsWf
oLrQmBoZPjYDpSzolUhFCGHJLovPMBSJsyDrodHLYOeKBoMcfCcgNnSuKxGCoEGgbuLX1K", "leftOrRight": "left"}
Finished
```

And the result is the same, we could still get a 25% increase of speed when sending out data from server. It shows that the load balancer can improve it.

In my test result, I can ensure the queue length doesn't continually grow, then shrink, giving a 'pointy' queue length profile, ie \wedge . An increase to a plateau is fine, ie $/ \backslash$. The plateau is less than around a 1000, which means my curve is in great shape.