6432021021
Jiratchaya Prajonkitchai

Activity 2 Web App Deployment (IaaS)

1. Understanding overload

1.1 What option did you use to run siege in order to trigger overload on your web servers and get HTTP status errors?

```
siege -c153 -d1 -r1 http://ec2-54-202-195-197.us-west-2.compute.amazonaws.com/index.php
```

```
[ec2-user@ip-172-31-17-222 ~]$ siege -c153 -d1 -r1 http://ec2-54-202-195-197.us-west-2.compute.amazonaws.com/index.php
[alert] Zip encoding disabled; siege requires zlib support to enable it
** SIEGE 4.1.6
** Preparing 153 concurrent users for battle.
The server is now under siege...
HTTP/1.1 500     36.64 secs:    1485 bytes ==> GET  /index.php
HTTP/1.1 200    167.67 secs:    3036 bytes ==> GET  /index.php
HTTP/1.1 200      0.01 secs:    3490 bytes ==> GET  /styles.css
HTTP/1.1 200      0.02 secs:     193 bytes ==> GET  /css?family=Lobster+Two
HTTP/1.1 200    168.00 secs:    3036 bytes ==> GET  /index.php
HTTP/1.1 200    168.01 secs:    3036 bytes ==> GET  /index.php
HTTP/1.1 200    168.00 secs:    3036 bytes ==> GET  /index.php
HTTP/1.1 200      0.01 secs:    3490 bytes ==> GET  /styles.css
HTTP/1.1 200      0.01 secs:    3490 bytes ==> GET  /styles.css
HTTP/1.1 200      0.01 secs:    3490 bytes ==> GET  /styles.css
HTTP/1.1 200      0.02 secs:     193 bytes ==> GET  /css?family=Lobster+Two
HTTP/1.1 200      0.02 secs:     193 bytes ==> GET  /css?family=Lobster+Two
```

```
Transactions:                    456 hits
Availability:                  99.78 %
Elapsed time:                170.67 secs
Data transferred:              0.98 MB
Response time:                56.36 secs
Transaction rate:              2.67 trans/sec
Throughput:                    0.01 MB/sec
Concurrency:                 150.59
Successful transactions:       456
Failed transactions:             1
Longest transaction:         169.55
Shortest transaction:          0.00
```

noted: using -c152 gets all HTTP 200 OK

```
[ec2-user@ip-172-31-17-222 ~]$ siege -c152 -d1 -r1 http://ec2-54-202-195-197.us-west-2.compute.amazonaws.com/index.php
[alert] Zip encoding disabled; siege requires zlib support to enable it
** SIEGE 4.1.6
** Preparing 152 concurrent users for battle.
The server is now under siege...
HTTP/1.1 200    161.16 secs:    3036 bytes ==> GET  /index.php
HTTP/1.1 200      0.00 secs:    3490 bytes ==> GET  /styles.css
HTTP/1.1 200      0.02 secs:     193 bytes ==> GET  /css?family=Lobster+Two
HTTP/1.1 200    161.44 secs:    3036 bytes ==> GET  /index.php
HTTP/1.1 200      0.00 secs:    3490 bytes ==> GET  /styles.css
HTTP/1.1 200      0.02 secs:     193 bytes ==> GET  /css?family=Lobster+Two
HTTP/1.1 200    161.63 secs:    3036 bytes ==> GET  /index.php
HTTP/1.1 200    161.65 secs:    3036 bytes ==> GET  /index.php
HTTP/1.1 200      0.01 secs:    3490 bytes ==> GET  /styles.css
HTTP/1.1 200      0.00 secs:    3490 bytes ==> GET  /styles.css
HTTP/1.1 200      0.02 secs:     193 bytes ==> GET  /css?family=Lobster+Two
HTTP/1.1 200      0.02 secs:     193 bytes ==> GET  /css?family=Lobster+Two
HTTP/1.1 200    161.67 secs:    3036 bytes ==> GET  /index.php
HTTP/1.1 200      0.01 secs:    3490 bytes ==> GET  /styles.css
```

```
Transactions:                    456 hits
Availability:                  100.00 %
Elapsed time:                166.99 secs
Data transferred:              0.97 MB
Response time:                54.53 secs
Transaction rate:              2.73 trans/sec
Throughput:                    0.01 MB/sec
Concurrency:                 148.91
Successful transactions:       456
Failed transactions:             0
Longest transaction:         165.87
Shortest transaction:          0.00
```

1.2 For the experiment using the siege options in 1.1, what is the average response time, transaction rate, and concurrency? How many of your transactions were successful vs. failed?

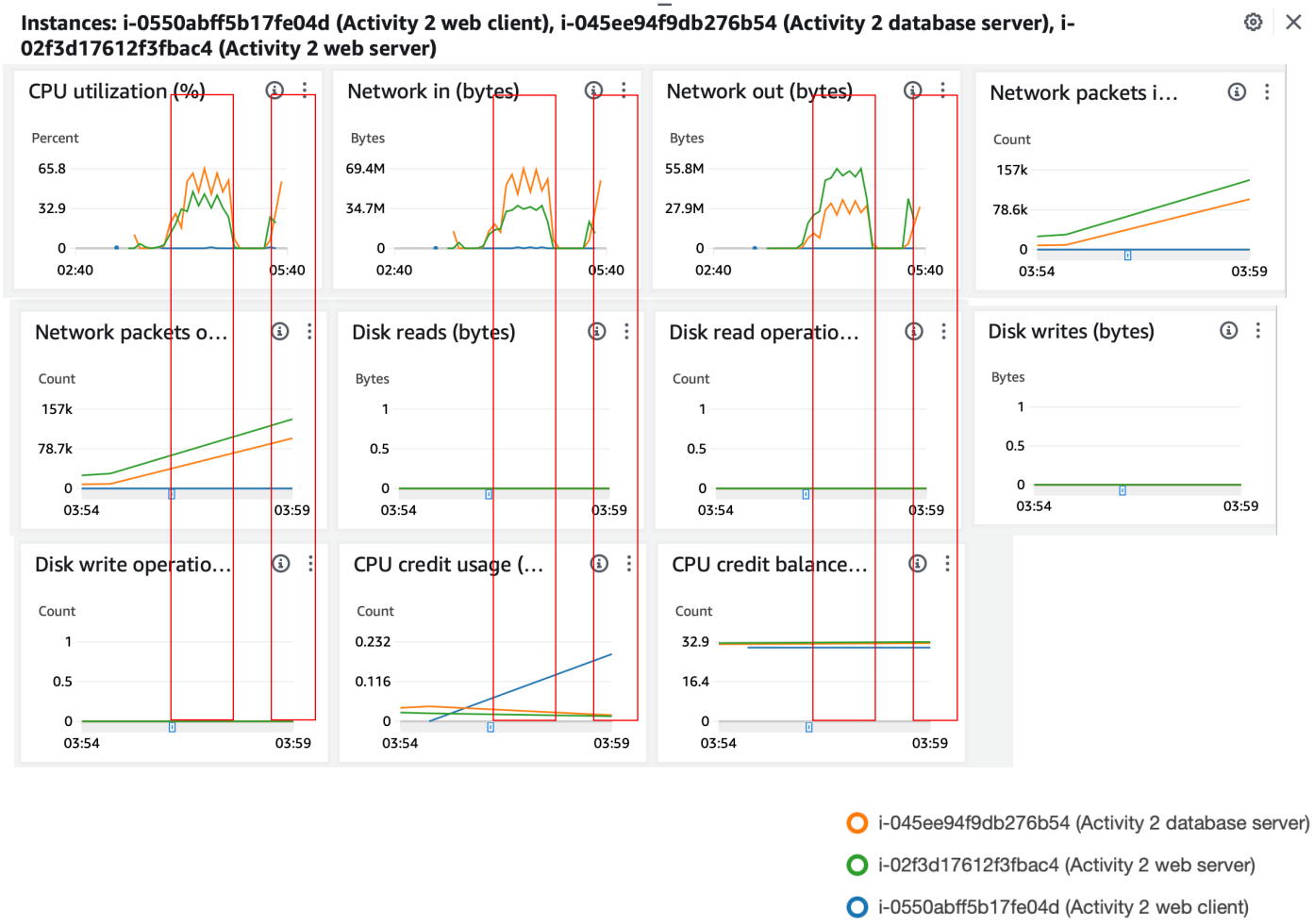
Average Response Time = 56.36 secs
Transaction Rate = 2.67 trans/sec
Concurrency = 150.59

Successful Transactions = 456
Failed Transaction = 1

```
Transactions:                    456 hits
Availability:                  99.78 %
Elapsed time:                170.67 secs
Data transferred:              0.98 MB
Response time:                56.36 secs
Transaction rate:              2.67 trans/sec
Throughput:                    0.01 MB/sec
Concurrency:                 150.59
Successful transactions:       456
Failed transactions:             1
Longest transaction:         169.55
Shortest transaction:          0.00
```

2. Take screenshots from the monitoring panel for your 3 instances to show load changes and point out when you start to HTTP status errors (not HTTP 200 OK).

The red rectangle indicates the commencement of HTTP status errors.



3. Do you think Amazon is providing sufficient resources for this instance size, price point, and number of clients you would generally have for a web application?

AWS offers a variety of instance types, so users have to choose their own instances based on specific needs. The pricing is flexible and is estimated depending on the usage and instance resources that users set up.

For this instance, I think Amazon is providing ample resources, particularly given that it's a free instance. It seems to be a decent fit for handling 152 users simultaneously, although the access time is longer compared to typical websites.

4. Where is the bottleneck in this system: your client, your webserver, or your database server? Why do you think so? Provide evidence.

I think the bottleneck is occurring in the database server. The evidence for this is the high CPU utilization observed in the performance graph, coupled with slow response times.