I made a Apache web server performance evaluation on different AWS instance types first.

Here is the set up of my first instance:

platform: Linux

instance type: t2.micro

web server: Apache 2.4.46

1 vCPUs and 1 GiB Meomory

Here is the outcome for the first test on my server:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Total data transferred | Test completed Time | Requests per seconds | Time per request | Transfer rate |
| -n 500 -c 100 | 4220500 bytes | 0.112 seconds | 4451.77 | 22.463 ms | 36696.63 [Kbytes/sec] |
| -n 500 -c 200 | 4220500 bytes | 0.225 seconds | 2221.03 | 90.048 ms | 18308.30 [Kbytes/sec] |
| -n 500 -c 300 | 4220500 bytes | 0.240 seconds | 2085.75 | 143.833 ms | 17193.17 [Kbytes/sec] |

Here are my observation about the outcome:

1. Total data transferred is 4220500 bytes for 500 requests. Close to 8K per page (un-compressed). Which is inline with my homepage size.
2. Test completed in 0.112 seconds. No failed requests. Perfect number I think.
3. Requests per seconds: 4451.77. Pretty good number.
4. Time per request: 22.463 milli-seconds (for 100 concurrent requests). So across all requests it is 22.463ms/100 = 0.22463ms.
5. Transfer rate: 36696.63 [Kbytes/sec] received. Since we ran the test on same machine, this number was expected to be high and can be ignored.
6. In connection time stats, you can see many requests had to wait for few seconds. This may be due to apache putting requests in wait queue.

Overall test results are pretty good. This test does not really tested the network latency. But it primarily tested if 100 concurrent requests reached to the web server, how does it perform.

Second test:

Here are my observation about the outcome:

1. Total data transferred is 4220500 bytes for 500 requests. Close to 8K per page (un-compressed). Which is inline with my homepage size.
2. Test completed in 0.225 seconds. No failed requests. Perfect number I think.
3. Requests per seconds: 2221.03. Pretty good number.
4. Time per request: 90.048 milli-seconds (for 200 concurrent requests). So across all requests it is 90.048ms/200 = 0.450ms.
5. Transfer rate: 18308.30 [Kbytes/sec] received. Since we ran the test on same machine, this number was expected to be high and can be ignored.
6. In connection time stats, you can see many requests had to wait for few seconds. This may be due to apache putting requests in wait queue.

Overall test results are pretty good. This test does not really tested the network latency. But it primarily tested if 200 concurrent requests reached to the web server, how does it perform.

Third test:

Here are my observation about the outcome:

1. Total data transferred is 4220500 bytes for 500 requests. Close to 8K per page (un-compressed). Which is inline with my homepage size.
2. Test completed in 0.240 seconds. No failed requests. Perfect number I think.
3. Requests per seconds: 2085.75. Pretty good number.
4. Time per request: 143.833 milli-seconds (for 300 concurrent requests). So across all requests it is 143.833ms/300 = 0.479ms.
5. Transfer rate: 17193.17 [Kbytes/sec] received. Since we ran the test on same machine, this number was expected to be high and can be ignored.
6. In connection time stats, you can see many requests had to wait for few seconds. This may be due to apache putting requests in wait queue.

Overall test results are pretty good. This test does not really tested the network latency. But it primarily tested if 300 concurrent requests reached to the web server, how does it perform.

I generated this graph by gnuplot and here is the out come:

Chart, table

Description automatically generated

Here is the set up of my second instance:

1. platform: Linux
2. instance type: t3a.xlarge
3. web server: Apache 2.4.46
4. 4 vCPUs and 16 GiB Meomory

Here is the outcome for the second test on my server:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Total data transferred | Test completed Time | Requests per seconds | Time per request | Transfer rate |
| -n 5000 -c 1000 | 15715000 bytes | 0.564 seconds | 8861.10 | 112.853 ms | 27197.69 [Kbytes/sec] |
| -n 5000 -c 2000 | 15715000 bytes | 0.796 seconds | 6282.91 | 318.324 ms | 19284.38 [Kbytes/sec] |
| -n 5000 -c 2500 | 15715000 bytes | 0.883 seconds | 5660.57 | 441.651 ms | 17374.20 [Kbytes/sec] |

Here are my observation about the outcome:

1. Total data transferred is 15715000 bytes for 5000 requests. Close to 3K per page (un-compressed). Which is inline with my homepage size.
2. Test completed in 0.564 seconds. No failed requests. Perfect number I think.
3. Requests per seconds: 8861.10. Pretty good number.
4. Time per request: 112.853 milli-seconds (for 1000 concurrent requests). So across all requests it is 112.853ms/1000 = 0.113ms.
5. Transfer rate: 27197.69 [Kbytes/sec] received. Since we ran the test on same machine, this number was expected to be high and can be ignored.
6. In connection time stats, you can see many requests had to wait for few seconds. This may be due to apache putting requests in wait queue.

Overall test results are pretty good. This test does not really tested the network latency. But it primarily tested if 1000 concurrent requests reached to the web server, how does it perform.

Second test:

Here are my observation about the outcome:

1. Total data transferred is 15715000 bytes bytes for 5000 requests. Close to 3K per page (un-compressed). Which is inline with my homepage size.
2. Test completed in 0.796 seconds seconds. No failed requests. Perfect number I think.
3. Requests per seconds: 6282.91. Pretty good number.
4. Time per request: 318.324 milli-seconds (for 2000 concurrent requests). So across all requests it is 318.324 ms ms/2000 = 0.159ms.
5. Transfer rate: 19284.38 [Kbytes/sec] received. Since we ran the test on same machine, this number was expected to be high and can be ignored.
6. In connection time stats, you can see many requests had to wait for few seconds. This may be due to apache putting requests in wait queue.

Overall test results are pretty good. This test does not really tested the network latency. But it primarily tested if 2000 concurrent requests reached to the web server, how does it perform.

Third test:

Here are my observation about the outcome:

1. Total data transferred is 15715000 bytes for 2500 requests. Close to 3K per page (un-compressed). Which is inline with my homepage size.
2. Test completed in 0.883 seconds. No failed requests. Perfect number I think.
3. Requests per seconds: 5660.57. Pretty good number.
4. Time per request: 441.651 ms milli-seconds (for 2500 concurrent requests). So across all requests it is 441.651 ms /2500 = 0.177ms.
5. Transfer rate: 17374.20 [Kbytes/sec] received. Since we ran the test on same machine, this number was expected to be high and can be ignored.
6. In connection time stats, you can see many requests had to wait for few seconds. This may be due to apache putting requests in wait queue.

Overall test results are pretty good. This test does not really tested the network latency. But it primarily tested if 2500 concurrent requests reached to the web server, how does it perform.

And we can see t3a.xlarge is much more powerful than t2.micro since t2.micro always fails to deal with 5000 requests. t3a.xlarge instance can easily handle with 5000 requests in different concurrent request situations.

I generated this graph by gnuplot and here is the out come:

Table

Description automatically generated