SYNCHROWEB
TECHNOLOGY

KIWIRE 1.6 ENTERPRISE PERFORMANCE BENCHMARK

Reviewed By:

Company	Name	Department	Date
Synchroweb Technology (M) Sdn Bhd	Pang Wee Tak	Technical	30/06/2016
Synchroweb Technology (M) Sdn Bhd	Mohamad Azriq Arshad	Technical	30/06/2016
Synchroweb Technology (M) Sdn Bhd	Mohd Eddy	Technical	30/06/2016

Related Documents

Document Name	Application and Release Number	Revision Number	Dated

Contents

1. Purpose of This Document

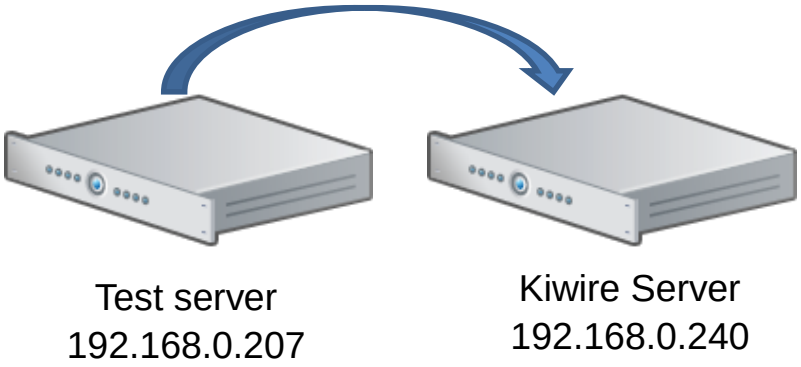
This document intent is to showcase a validated benchmark report of load stress test of kiwire 1.6 software

2. Scope

The tests will be conducted base on the standard version on a standard off the shelf hardware

3. Hardware

Hardware chosen are standard hardware from Dell, both test software and Kiwire are using the same type of hardware and mikrotik switches.



Logical diagram of test lab setup.

4. Server Details

Server Model	QuantaGrid S31A-1U
CPU	1 socket dual core
CPU model	
Ram	64 GB
Harddisk	960 GB (RAID 1)
Operating System	CentOS release 6.8 (Final)
Database	MySQL 5

5. Performance objective

The objective of this benchmark performance test in our lab is to perform and evaluate kiwire handle to stress load of 40,000 users as benchmark.

Kiwire consist of few components mainly

- Apache web server
- Radius services
- MySQL database engine

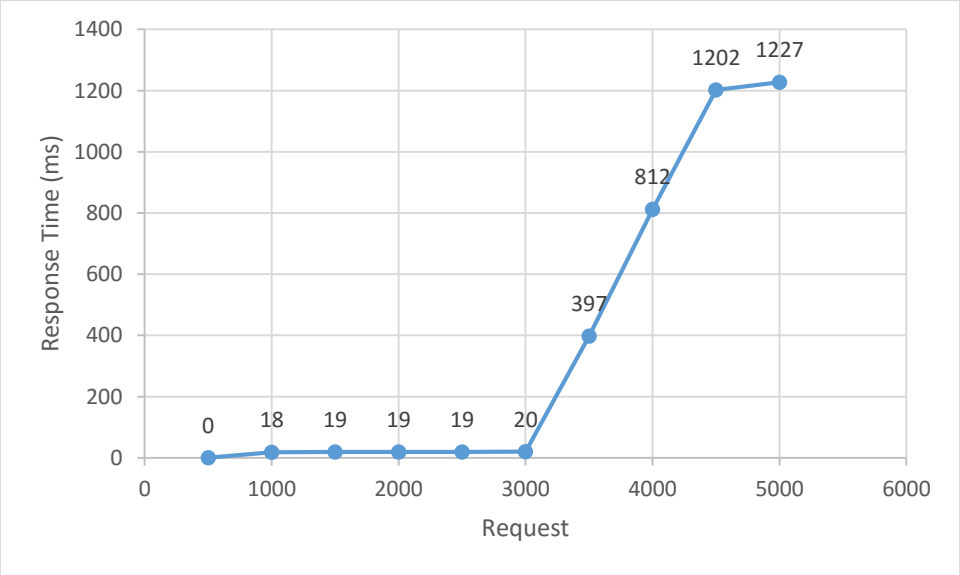
As the apache and radius are loading data from mysql every benchmark will directly load test the MySQL database this will ensure the mysql is being benchmark as well.

5.1. Apache Benchmark

The tools we use will be apache “ab”. ab is a tool for benchmarking Apache Hypertext Transfer Protocol (HTTP) server. It is designed to give an impression of how current Apache installation performs. This especially shows how many requests per second Apache installation is capable of serving.

5.1.1. Benchmark 1

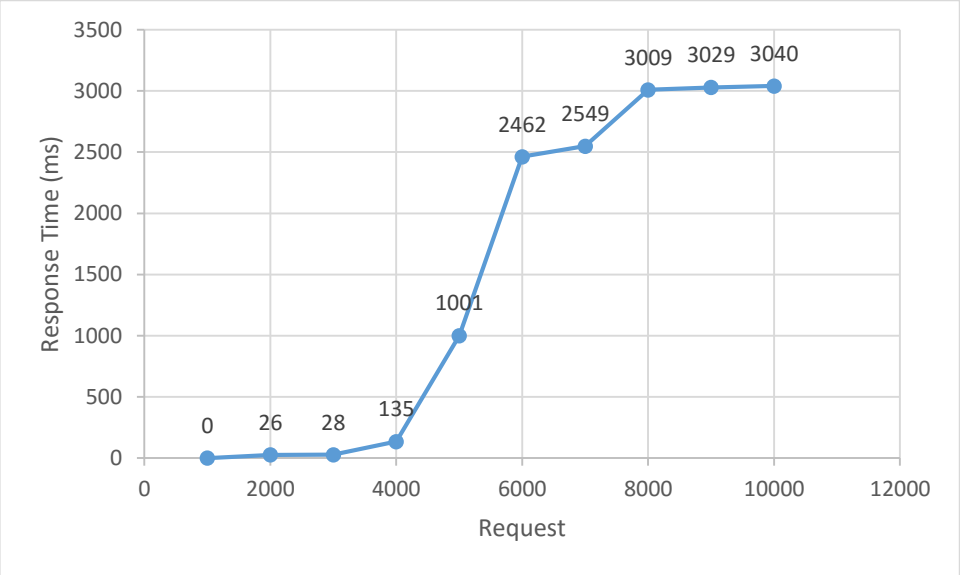
On benchmark 1 , we be doing a test of 5000 request with 5000 concurrency at 1 burst.



```
[root@localhost]# ab -k -n 5000 -c 5000 http://192.168.0.240/
This is ApacheBench, Version 2.3 <$Revision: 655654 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 192.168.0.240 (be patient)
Completed 500 requests
Completed 1000 requests
Completed 1500 requests
Completed 2000 requests
Completed 2500 requests
Completed 3000 requests
Completed 3500 requests
Completed 4000 requests
Completed 4500 requests
Completed 5000 requests
Finished 5000 requests


Server Software:        Apache/2.2.15
Server Hostname:        192.168.0.240
Server Port:            80

Document Path:          /
Document Length:        4961 bytes

Concurrency Level:      5000
Time taken for tests:   1.267 seconds
Complete requests:      5000
Failed requests:        0
Write errors:           0
Non-2xx responses:      5019
Keep-Alive requests:    0
Total transferred:      25986818 bytes
HTML transferred:       24877619 bytes
Requests per second:    3947.44 [#/sec] (mean)
Time per request:       1266.644 [ms] (mean)
Time per request:       0.253 [ms] (mean, across all concurrent requests)
Transfer rate:          20035.43 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0   31 145.0      1    1112
Processing:     6   41 114.1     17     650
Waiting:        1   41 114.1     16     650
Total:         12   73 207.4     18    1227

Percentage of the requests served within a certain time (ms)
  50%     18
  66%     19
  75%     19
  80%     19
  90%     20
  95%    397
  98%    812
  99%   1202
 100%   1227 (longest request)
```

5.1.2. Benchmark 2

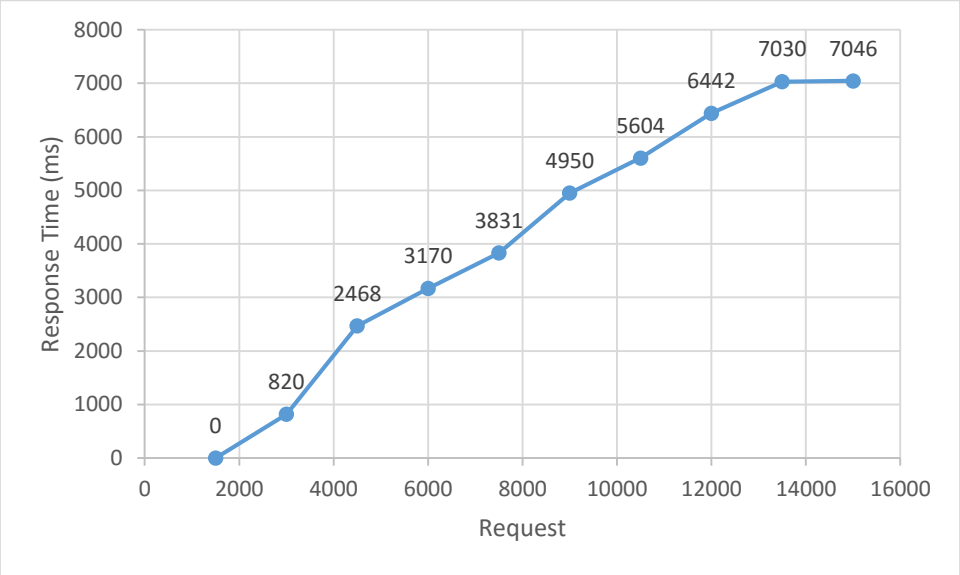
On bechmark 2 we perform a test with 10,000 request with 10,000 concurrency at 1 go this simulate a flood user of user.




```
Server Software:        Apache/2.2.15
Server Hostname:        192.168.0.240
Server Port:            80

Document Path:          /
Document Length:        4961 bytes

Concurrency Level:      10000
Time taken for tests:   3.090 seconds
Complete requests:      10000
Failed requests:        0
Write errors:           0
Non-2xx responses:      10015
Keep-Alive requests:    0
Total transferred:      51862220 bytes
HTML transferred:       49648905 bytes
Requests per second:    3235.90 [#/sec] (mean)
Time per request:       3090.326 [ms] (mean)
Time per request:       0.309 [ms] (mean, across all concurrent requests)
Transfer rate:          16388.79 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0  230 544.4      2    3008
Processing:     0  202 484.3     23    2338
Waiting:        0  200 482.6     23    2338
Total:         17  433 837.4     26    3040

Percentage of the requests served within a certain time (ms)
  50%     26
  66%     28
  75%    135
  80%   1001
  90%   2462
  95%   2549
  98%   3009
  99%   3029
 100%   3040 (longest request)
```

5.1.3. Benchmark 3

On bechmark 2 we perform a test with 15,000 request with 15,000 concurrency at 1 go this simulate a flood user of user.




```
Server Software:        Apache/2.2.15
Server Hostname:        192.168.0.240
Server Port:            80

Document Path:          /
Document Length:        4961 bytes

Concurrency Level:      15000
Time taken for tests:   7.133 seconds
Complete requests:      15000
Failed requests:        0
Write errors:           0
Non-2xx responses:      15042
Keep-Alive requests:    0
Total transferred:      77869538 bytes
HTML transferred:       74545256 bytes
Requests per second:    2102.88 [#/sec] (mean)
Time per request:       7133.085 [ms] (mean)
Time per request:       0.476 [ms] (mean, across all concurrent requests)
Transfer rate:          10660.81 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median   max
Connect:        0  982 1458.1      6    7033
Processing:     0  752 1189.7     52    5116
Waiting:        0  746 1186.9     52    5115
Total:         11 1734 2027.0    820    7046

Percentage of the requests served within a certain time (ms)
  50%    820
  66%   2468
  75%   3170
  80%   3831
  90%   4950
  95%   5604
  98%   6442
  99%   7030
 100%   7046 (longest request)
```

5.1.4. Summary

The load test shown that the system is able to witstand 15,000 connection incoming without any error and time taken to furfill the load of web portal page is within 7.1 second. The time to load will increase as the load increases which is acceptable between normal parameter.

5.2. Radius Server Load test

The Radius engine is an RFC compliant enterprise radius engine. For the test we be using Linux radclient which can send no of requests in parallel to the radius server. Radius is already a robust service, we will perform multi-thread load test which will initiate the program 45 instance with 1000 connection for start to simulate 45,000 connection to the system.

Command used is

"*time echo "User-Name=test, Password=test" | radclient -c 1000 -p 1000 -q -s 192.168.2.9:1812 auth testing123*"

5.2.1. Benchmark 1

The first benchmark is 10,000 connection request via 10 instance with 1000 request each. Which checking if there is any lost-auth which stand for server unable to response or timeout.

Instance with 1000 request in parallel	Time to complete (in second)
1	5.257
2	5.426
3	5.646
4	5.625
5	5.700
6	5.632
7	5.609
8	5.686
9	5.653
10	5.669
Average	**5.5903 Sec**

Lost Auth : 0

5.2.2. Benchmark 2

20,000 connection via 20 instances.

Instance with 1000 request in parallel	Time to complete (in second)
1	13.863
2	14.112
3	14.189
4	12.250
5	12.337
6	12.552
7	14.263
8	12.278
9	14.201
10	13.825
11	14.041
12	14.247
13	13.033
14	12.465
15	14.160
16	14.011
17	12.753
18	12.792
19	12.637
20	13.841
Average	**13.393 Sec**

Lost Auths : 0

5.2.3. Benchmark 3

30,000 connection via 30 instances.

Instance with 1000 request in parallel	Time to complete (in second)
1	5.500
2	41.372
3	41.295
4	41.264
5	41.422
6	41.356
7	41.235
8	36.274
9	36.220
10	41.093
11	41.453
12	41.113
13	41.330
14	41.157
15	36.347
16	41.164
17	41.148
18	41.265
19	41.112
20	34.479
21	26.251
22	18.467
23	12.193
24	5.138
25	11.101
26	19.671
27	12.057
28	12.559
29	11.622
30	19.310
Average	**30.532 Sec**

Lost Auths : 0

5.2.4. Benchmark 4

40,000 connection via 40 instances.

Instance with 1000 request in parallel	Time to complete (in second)
1	43.120
2	0.206
3	52.027
4	59.541
5	59.876
6	53.337
7	51.851
8	0.842
9	52.413
10	54.017
11	59.133
12	53.857
13	0.798
14	1.717
15	1.063
16	1.684
17	54.485
18	0.869
19	6.365
20	54.846
21	41.072
22	48.808
23	26.967
24	32.845
25	33.676
26	38.955
27	34.102
28	26.368
29	26.288
30	24.377
31	32.841
32	33.604
33	17.577
34	25.815
35	26.098
36	45.576
37	52.954
38	45.362
39	53.769

40	31.144
Average	**34.006 Sec**

Lost Auths : 0

5.2.5. Benchmark 5

45,000 connection via 45 instances.

Instance with 1000 request in parallel	Time to complete (in second)
1	50.870
2	60.969
3	61.730
4	66.905
5	61.048
6	59.570
7	61.275
8	53.447
9	67.104
10	61.404
11	61.364
12	46.511
13	61.395
14	54.632
15	59.796
16	58.671
17	53.114
18	60.624
19	60.036
20	43.705
21	54.047
22	46.167
23	26.762
24	25.498
25	33.620
26	34.209
27	40.162
28	27.638
29	20.935
30	28.039

31	40.336
32	25.531
33	26.289
34	25.637
35	25.622
36	38.569
37	41.275
38	33.560
39	32.363
40	33.400
41	33.386
42	32.655
43	67.197
44	67.309
45	27.115
Average	**45.598 Sec**

Lost Auths : 0

6. Stress test , benchmark summary.

From the load test performed its safe to conclude the kiwire can safely handle 45,000 request flood and able to compute it within 45secs with zero fault and error .

This is a result of our development culture of utilize coding techniques and good programming practices to create high quality code plays an important role in software quality and performance. In addition, by consistently applying a well-defined coding standard and proper coding techniques, and holding routine code reviews.

Note:
This test is performed in an full kiwire enviroment without any 3rd party api or interface involved , as 3rd party api are not part of kiwire standard application services offering.

The test is perform on a local area network which network lattency will not affect our performance. In real world the performance result may defer due to end-user network enviroment.