Performance Test Report

**High Level**

Client wanted to assess performance and stability of the site <https://momenton.com.au/> with a short quick ramp up test , to understand how platform and web user experience can be impacted with this quick burst of the load

**High Level Result Over view**

The platform and web application suite which hosted the site <https://momenton.com.au/>

had adverse impacts due to the burst load subjected to. During the peak steady state load average Response times were high as up to 27 seconds with 95 percentiles leading up to 42 seconds.

At peak load duration which is at 1000, threads, the application response time was steady. However, it is worth pointing to note below key observations

1. Application under test has an adverse user impact when subject to peak loads of this burst and will not be able to handle this workload for longer duration.
2. The aforesaid issue could be due to limitation in current http #httpthreadpool configuration (within the web server).
3. There is a potential connection queueing is happening and hence web server is rejecting the new incoming connections. This could be due to a server-side limitation within large JVM (AKA CPU)

However as the AUT (application under test) is not a large transactional system which do not anticipate the tested work load ,hence this would be of less significance .

**Details**

The performance test was done using JMETER load testing tool with 5.3x version and have generated this report using out of the box features of the JMeter.

Start Time: ""14/7/20, 3:18 pm""

End Time: ""14/7/20, 3:20 pm""

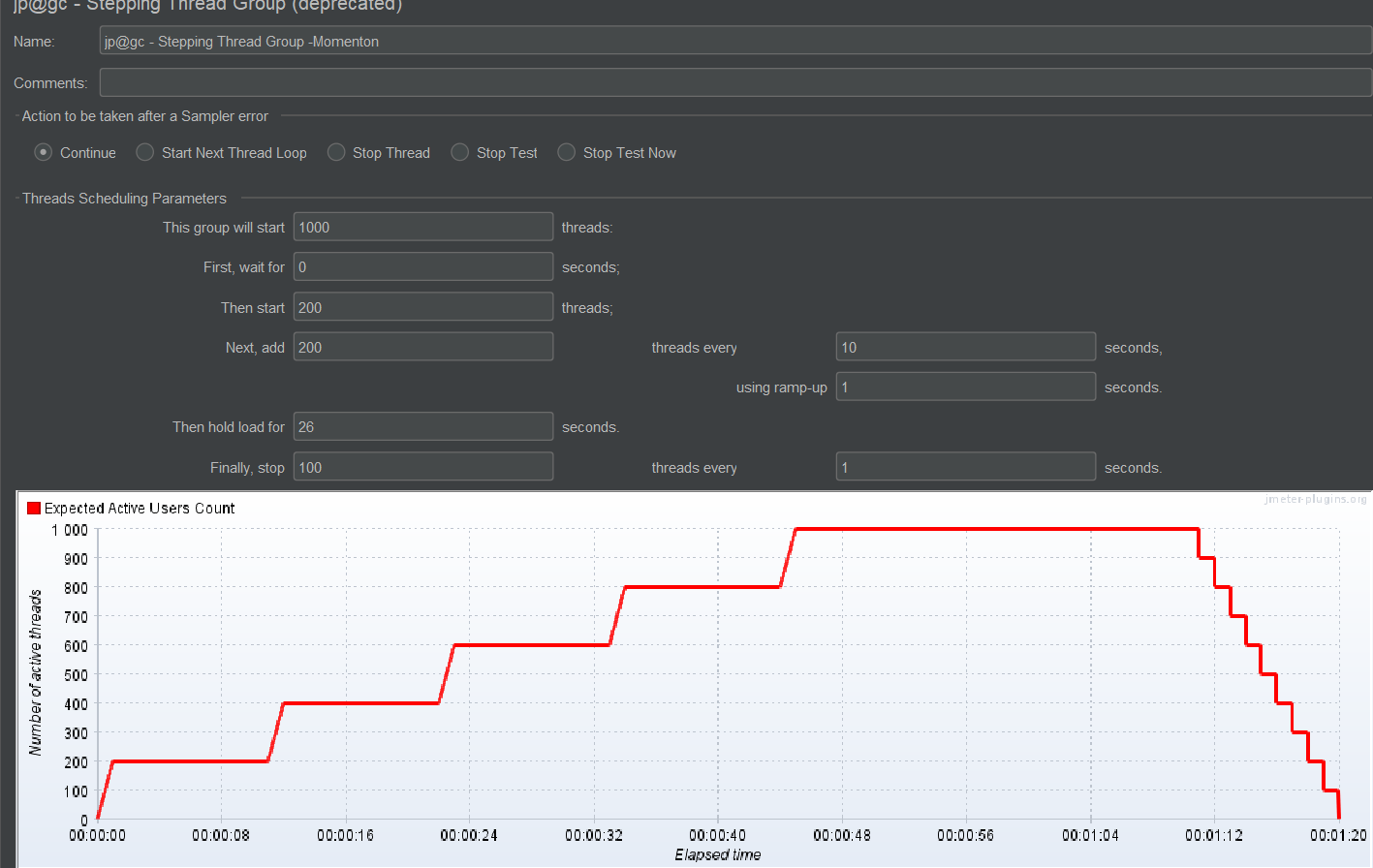
Application Under Test: <https://momenton.com.au/>

**Work load model**

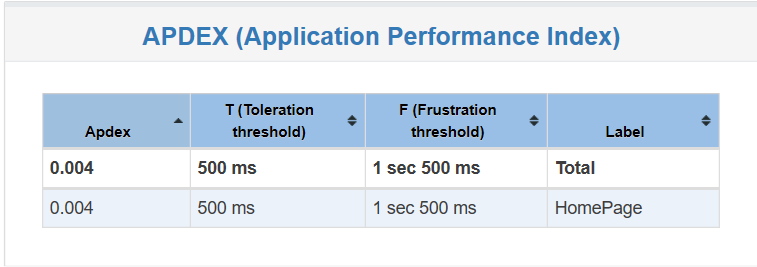
The load injection model was straightforward as given in challenge. I have performed this in step up manner to avoid a major burst send into their server farms.

The load was done with 1000 threads each as shown below. No specific through put was set

The load model is called a step-up model with each step comprises of 200 threads each step holding load for 10 second duration and on final step with 1000 thread, the load was injected for 25 second duration as instructed in challenge



[APDEX](https://en.wikipedia.org/wiki/Apdex) (Application Performance Index) table that computes for every transaction the APDEX based on configurable values for tolerated and satisfied thresholds

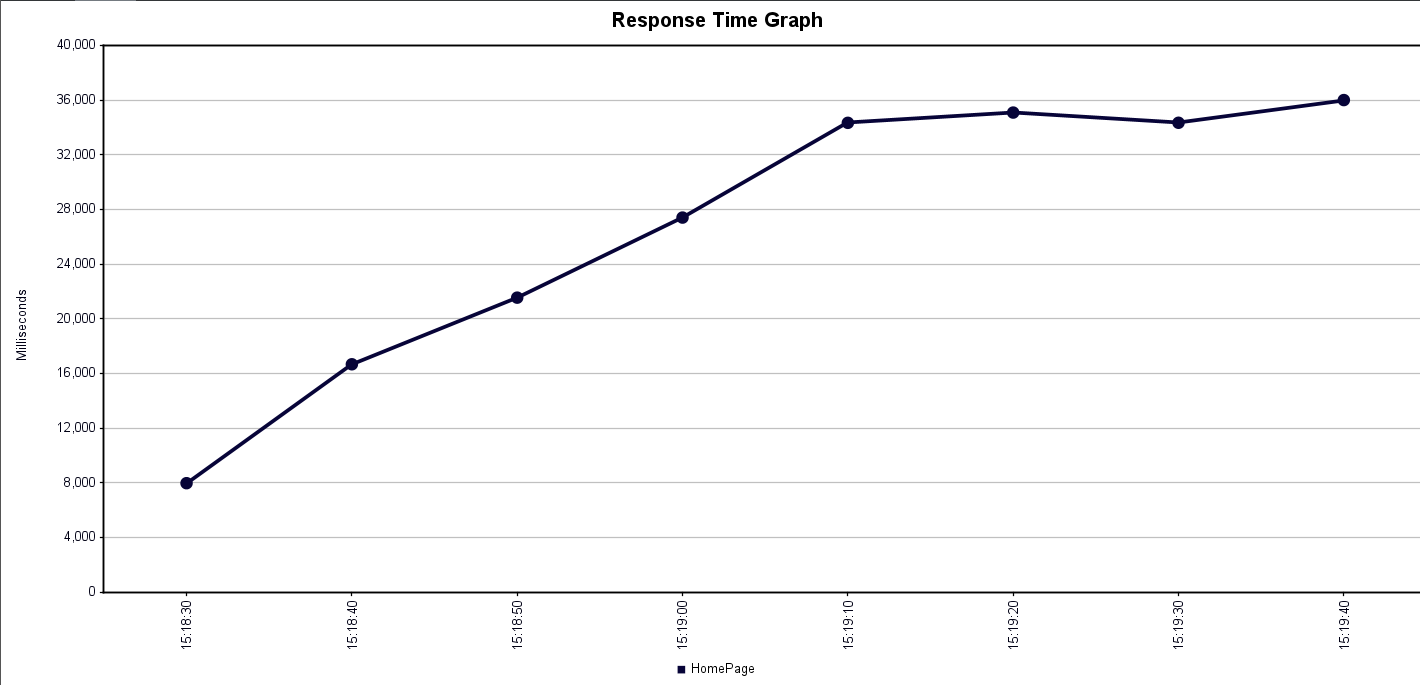


**Response time Overview and end User Impacts**

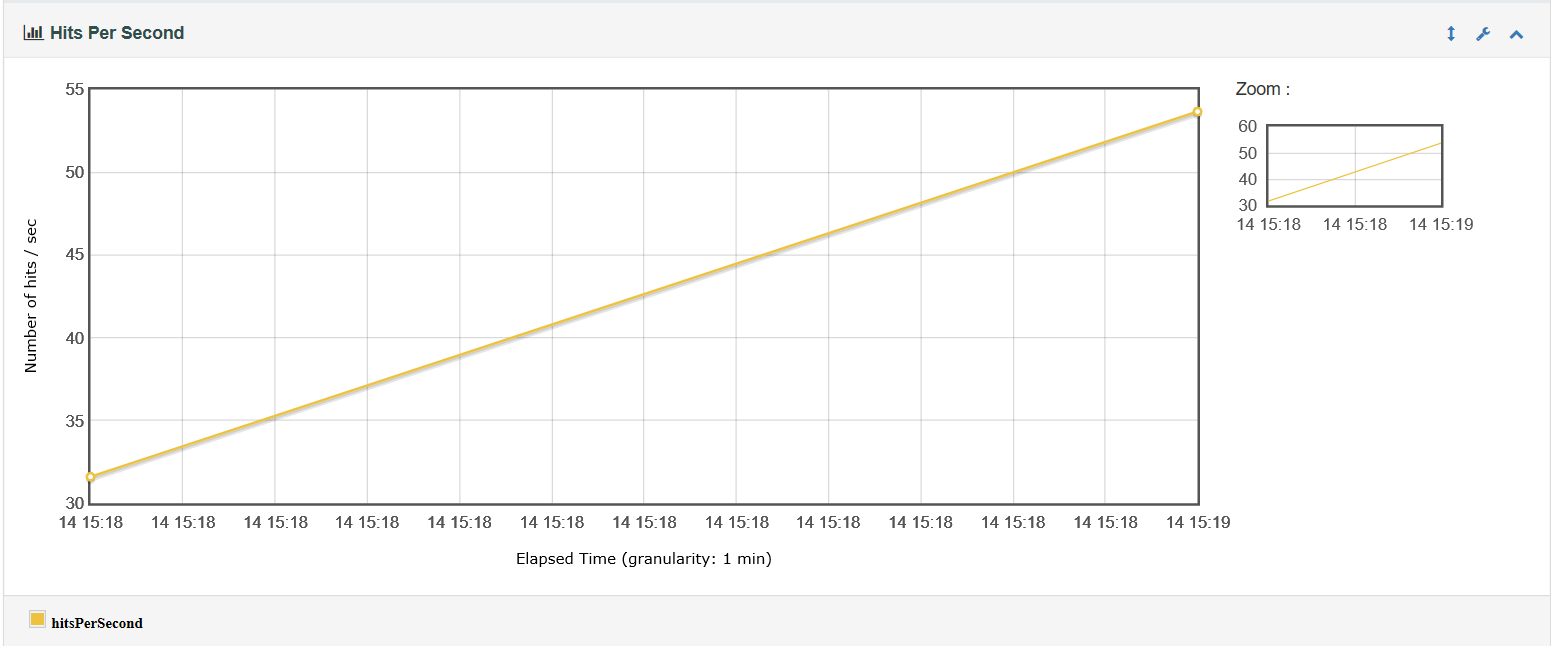
Average response time during the test observed to be approximately 27 seconds with an approximate average through put load of 23.7 transactions per second across whole duration including ramp up and ramp down, which is quite high for a web site this. However, we need to consider the fact that this is not a customer facing transaction and not expecting large concurrent user load at any given point of time.

The below report shows that we have injected approximately 2558 requests over the duration of the test 95 percentile of the response times were around 42 second which is rest of the samples were higher than this value

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Request Name** | **Throughput(ms)** | **Average(ms)** | **90% PC (ms)** | **95% PC (ms)** | **Error %** | **Throughput (TPS)** |
| Homepage | 2558 | 26968 | 39771 | 42180 | 6.41% | 23.66262 |
| TOTAL | 2558 | 26968 | 39771 | 42180 | 6.41% | 23.66262 |



Above Response time and latency graph with liner to time clearly indicates a server side resource constraint , were in which server is unable to accept new connections .This could be attributed due to web incorrect or inadequate HTTP connection pool within the web server .The below provided through put graph concur to the above observation as through put achieved gradually getting decreased as the load injection rises .



**Identified end user Impact**

During the load test the web site has become quite slow and many of the requests were taking more than 1.5 seconds to return a successful page. A lot of requests (6.4%) failed to get response back and resulted in error which indicates that if same burst load occurs in production there is a heavy user impact with lot of customer dissatisfaction

**Error Analysis**

The error code suggest default JMETER time out value of 20 second was suffient to meet time out from server . The server was unable to accept the requested HTTP connections to server during this time

Recommendation: Increase the JMETER time out to see whether more wait by client could get connections from server .[Could be again due to #httpconnection pool issue]

