Gagan Gautam

009301019

Performance Analysis for Jax Web Services and Node.js(REST)

The performance analysis is performed for the server 1 which is simple math solver having two functions find_prime and check_prime. Find_prime takes a number as a input and output all the prime number between 1 and that number. Check_prime takes a number as input and returns whether the number is prime or not.

Case 1: 1000 calls by a single user

Average Time Taken (**JAX WEB Services**) :

1000 calls - 17400/1000= 17.4 ms

Average Time Taken (**node.js(REST)**):-

1000 calls - 8ms

Increase in latency is more than 10%

Case 2: 5000 calls by a single user

Average Time Taken (**JAX WEB Services**) :

1000 calls - 19399/1000= 19.4 ms

Average Time Taken (**node.js(REST)**):-

1000 calls - 7 ms

Increase in latency is more than 10%

Case 3: 100 users with 1000 calls

Average Time Taken (**JAX WEB Services**) :

Aprox. 21 ms

Analysis:

latency is the time delay between the response received for the request sent. So According to above analysis node.js(REST) is able to resolve the request received much faster than the JAX web services. Following are the reasons:

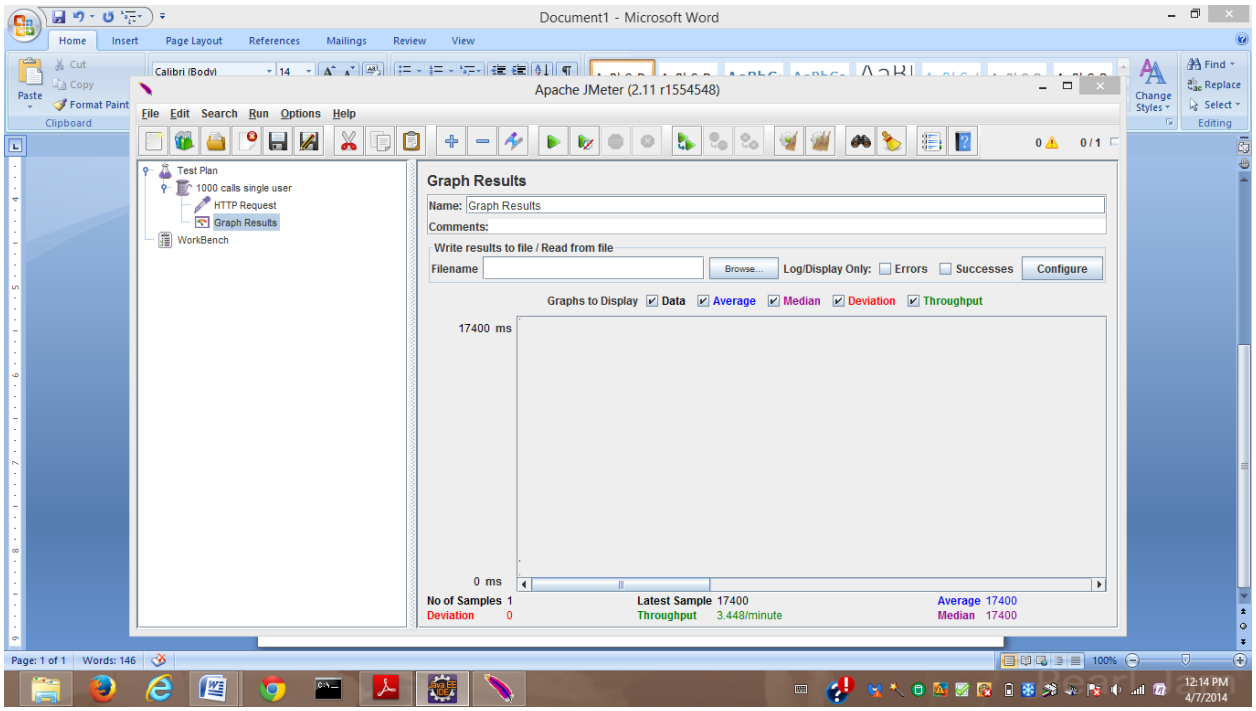
>node.js(REST) does not uses synchronous calls and thus does not wait for the response of a request if it is taking long. It goes on to serve another request. Since it executes asynchronously it is able to serve more requests which increases its throughput and the average time taken to resolve a request. It uses callback functions to support it's model. In case of the JAX-WS it resolves a request synchronously. So if a request takes more time to resolve , it waits for that request to finish and then only it goes on to serve other request. hence the average time taken increases and thus the latency increases.

> Other reason is the architecture of node.js(REST) and JAX-WS. Node.js(REST) does not require Tomcat server for its execution and everything is sent along the request and response which helps it to serve the requests faster. In case of JAX-WS the request and response has to pass through different layers which increases the latency time.

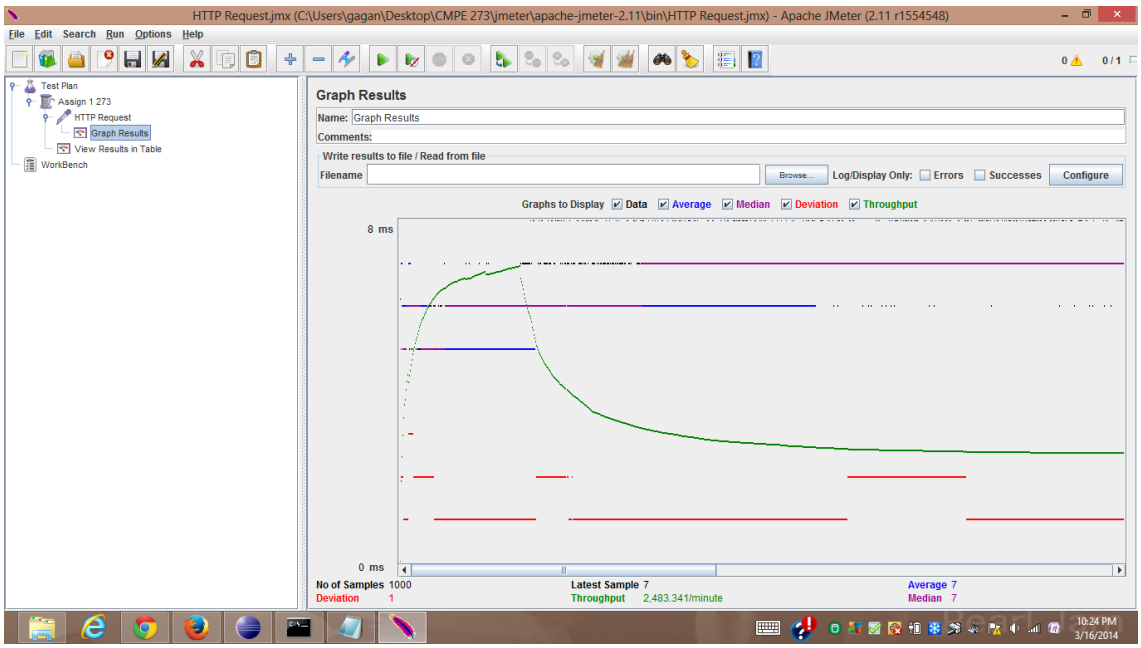
>One more observation is that node.js(REST) scales better than the JAX-WS. AS the average time for 5000 requests decreased in case of node.js that is due to it's asynchronous nature using event driven model as it does not queue requests and response like JAX-Ws.

Following are the result screenshots:

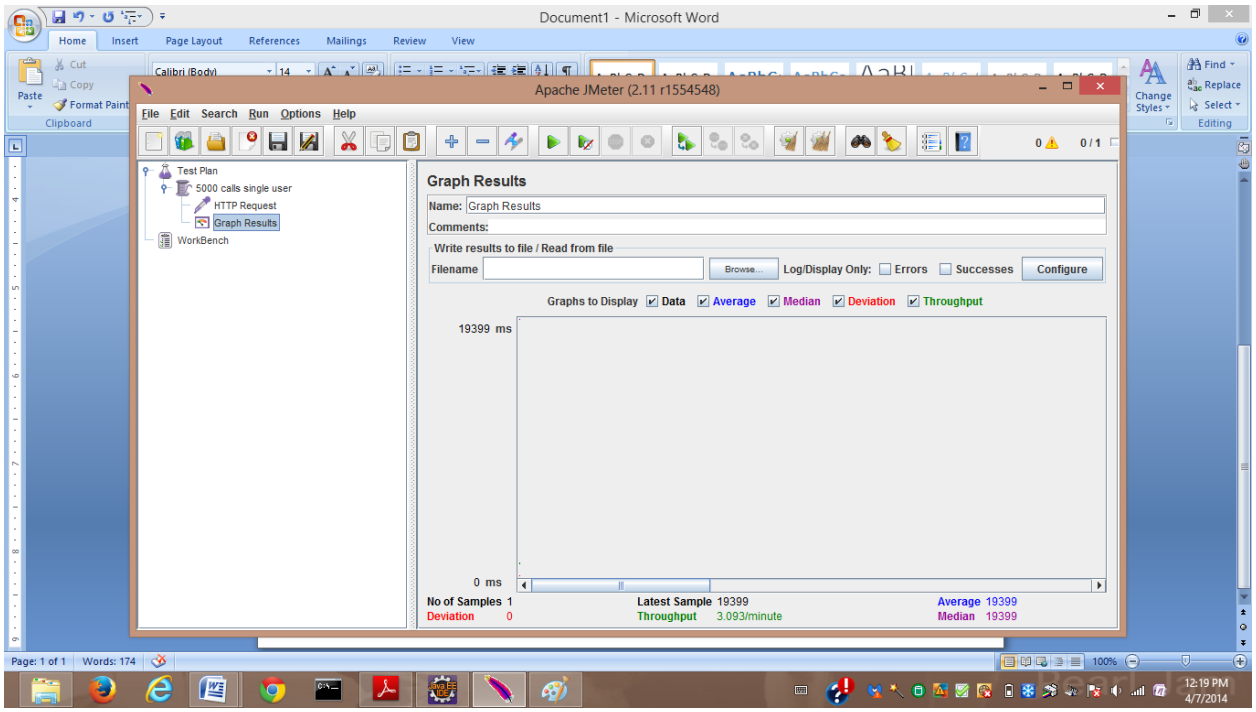
1000 calls by a single user in JAX web service:



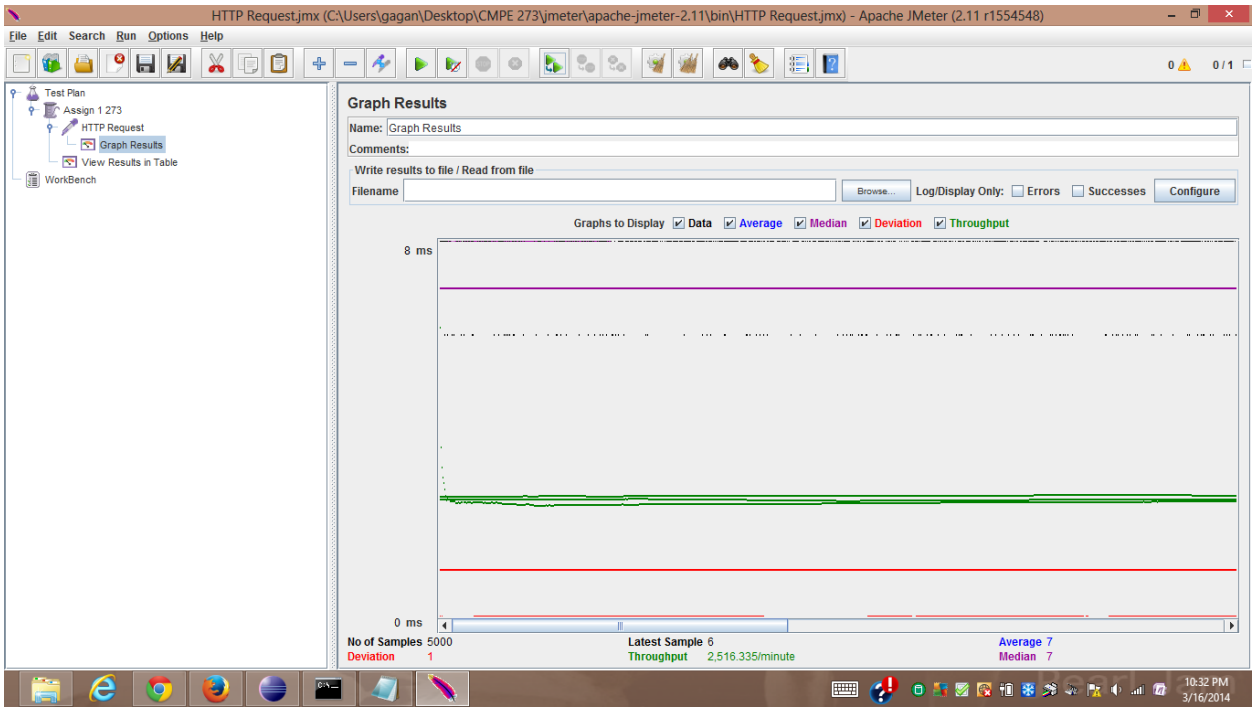
graph with 1000 calls to the server(single user)(node.js)



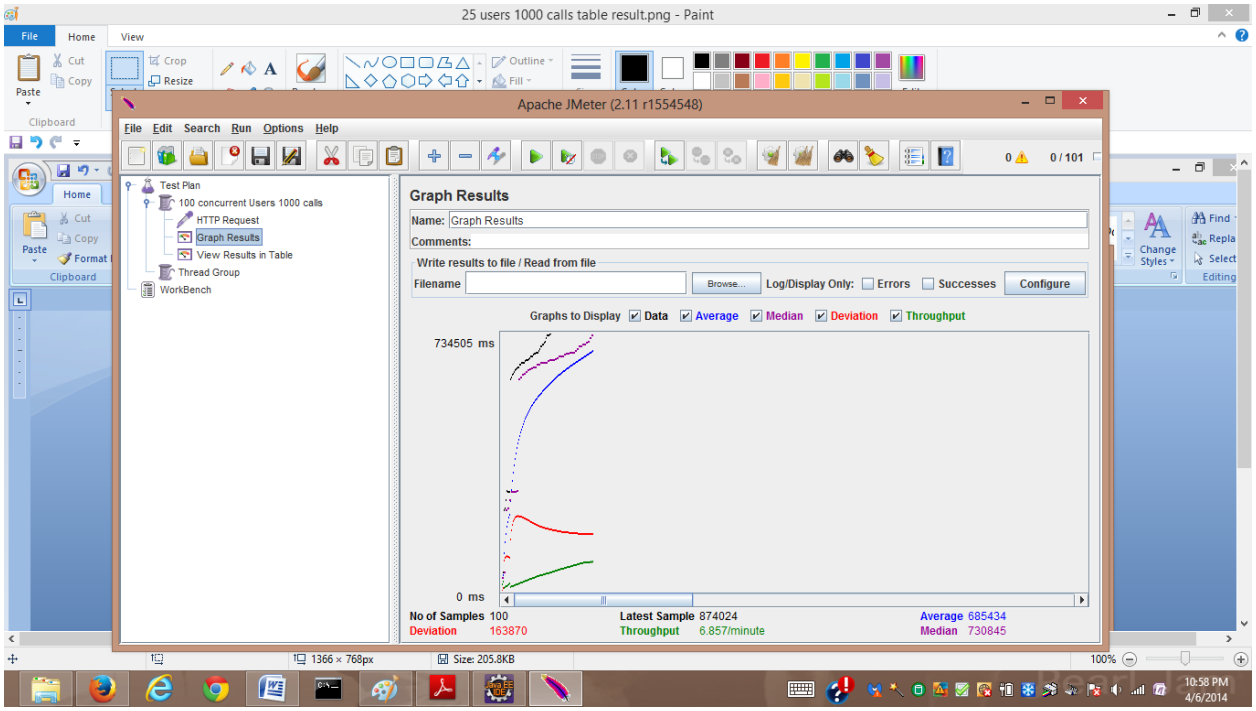
Graph with 5000 calls for single user using Jax-Ws

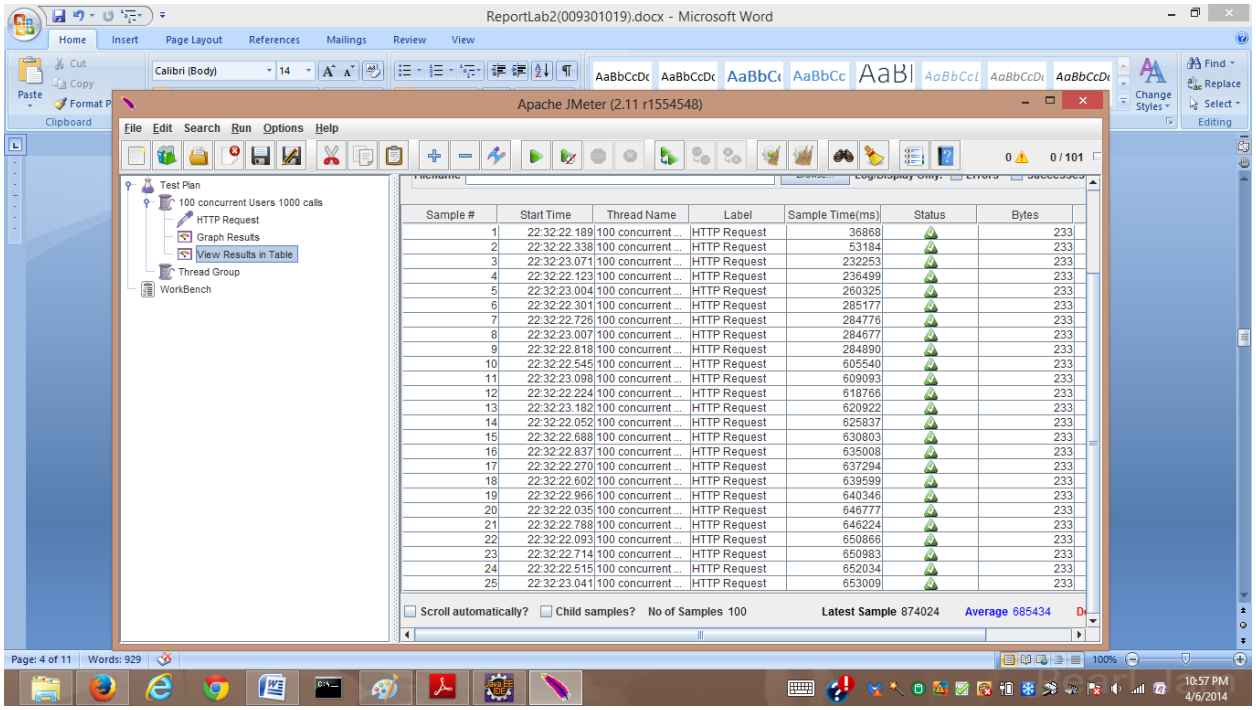


Graph with 5 thousand calls(single user)(node.js)



Graph for JAX-WS for 25 users with 1000 calls(table and graph result):





Graph with 100 users with concurrent 1000 calls(Node.js)