SpreadsheetWEB

API 2.0

Benchmarking Study

October 2015

Pagos, Inc.
47 3rd St
Cambridge, MA 02141
www.pagos.com

Table of Contents

Introduction

The purpose of this document is to present performance figures for SpreadsheetWEB's new API 2.0. Various scenarios consisting of differing concurrent usages were tested. The results were presented for each scenario to help the reader decide upon an optimum configuration, based on their required usage statistics. These scenarios are meant to realistically mirror software usage situations and should be used as reference for the reader's actual system requirements.

Test Configuration

The test machine was a 2.4 Ghz Intel Xeon (16 processors) with 8GB of RAM, running Windows Server 2012 R2. SpreadsheetWEB was configured to use 12 calculation nodes concurrently. All API requests were sent via Apache JMeter.

Test Scenarios

A spreadsheet of moderate complexity has been used in these tests. The spreadsheet (RatingEngine.xlsx) is designed for insurance premium calculations (e.g. HMO health insurance products). The file size is 1.3MB.

RatingEngine.xls was published as a web service. Apache JMeter (http://jmeter.apache.org/) was used to conduct web service tests based on varying concurrent usages.

Http Request used for benchmark was created by posting 4 input fields and getting 24 output fields in return. Data traffic caused by each request was 1,526 bytes.

Each test scenario consist of 5,000 requests ran at different concurrent user levels of 25, 40, 50, 65, 75, 90 and 100. Concurrent users are defined as the number of users making requests in every 2 seconds.

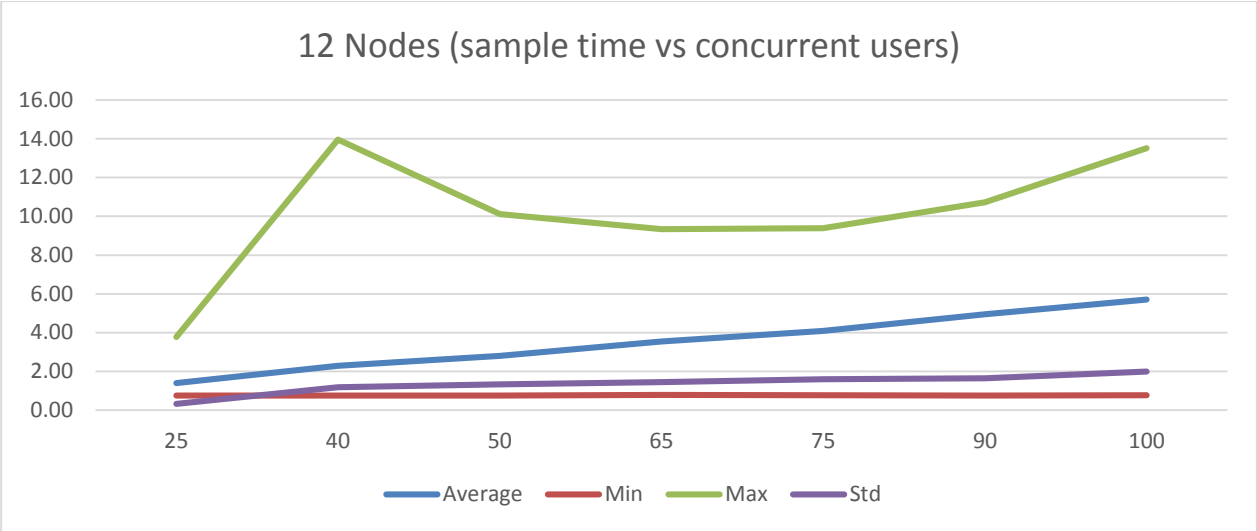
Based on capabilities of SpreadsheetWEB API 2.0, tests were run on 3 different configurations:

- Connection to a public calculation engine without session
- Connection to a private calculation engine without session
- Connection to a private calculation engine with session

Test Results

Connection to a public calculation engine without session

Average response times varied from 1.40 to 5.70 seconds, for 25 to 100 concurrent users respectively. Expected average response times increased, as the number of concurrent went up. Between 25 and 100 concurrent users, increase in average response times was fairly linear.



Note that minimum response times measured were almost identical around 0.7 seconds.

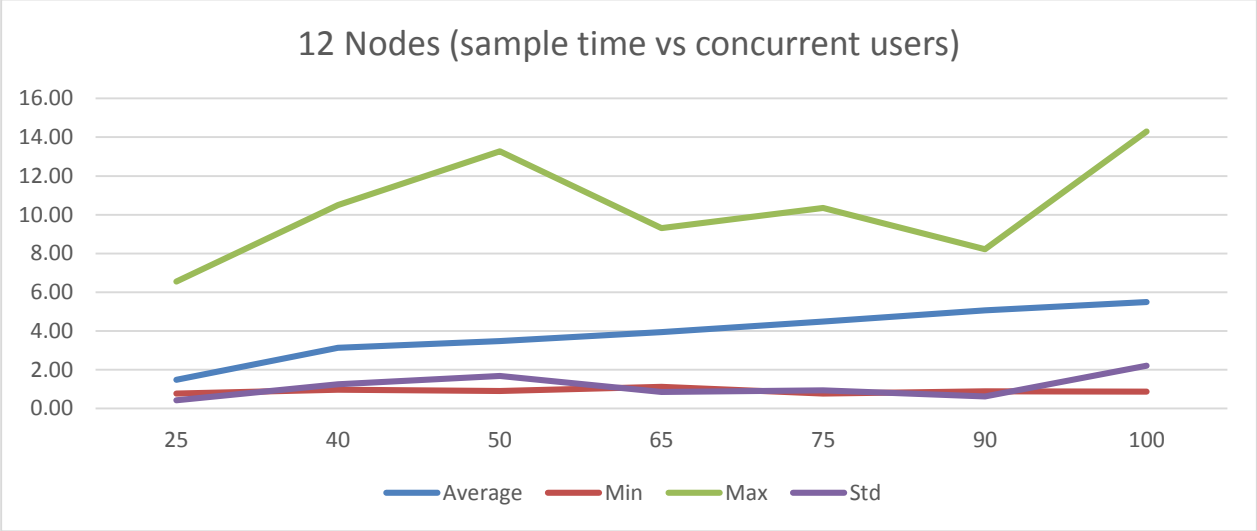
Concurrent User (per 2 sec)	Average	Min	Max	Std
25	1.40	0.75	3.78	0.32
40	2.29	0.76	13.96	1.18
50	2.79	0.75	10.12	1.34
65	3.55	0.78	9.33	1.44
75	4.08	0.77	9.39	1.59
90	4.95	0.76	10.73	1.64
100	5.70	0.77	13.52	1.99

Table below shows throughput measured as number of requests processed per seconds and it gave relatively stable results. Server was able to process 1.49 million records in a day. Top CPU usage was measured to be around 98% during these tests.

Concurrent User (per 2 sec)	Throughput (req per sec)
25	17.4
40	16.7
50	17.2
65	17.6
75	17.6
90	17.4
100	16.9

Connection to a private calculation engine without session

In this test, each request had to be authenticated first. Average response times varied from 1.48 to 5.50 seconds for 25 to 100 concurrent user range. Response times were slower in the low end, as can be expected due to authentication. Response times were similar for larger concurrent user scenarios.



Note that minimum response times varied between 0.78 and 1.12, attributable to authentication.

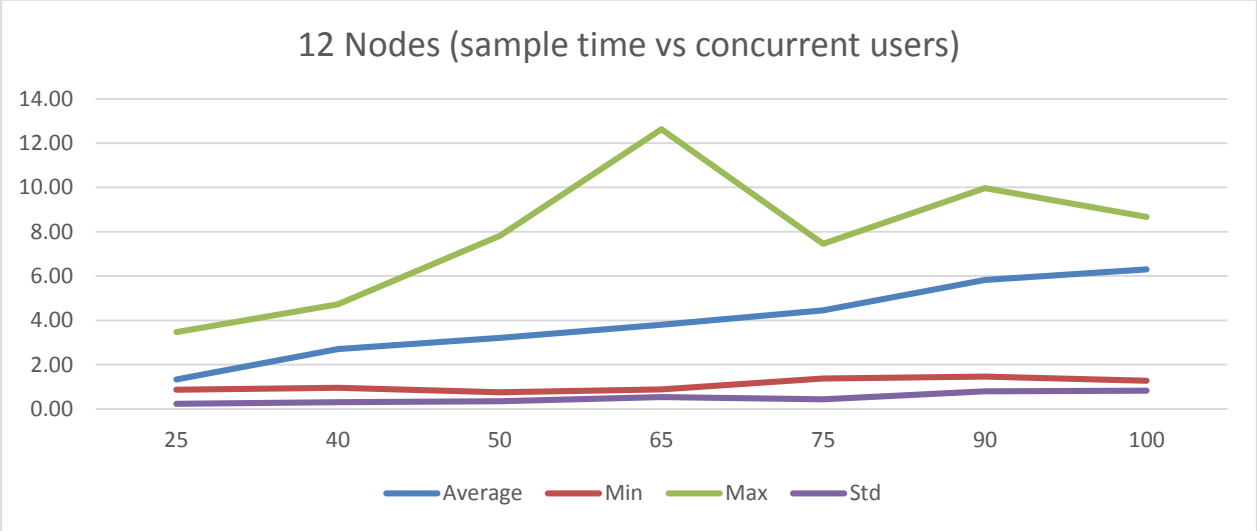
Concurrent User (per 2 sec)	Average	Min	Max	Std
25	1.48	0.77	6.55	0.43
40	3.14	0.98	10.49	1.25
50	3.49	0.90	13.27	1.68
65	3.95	1.12	9.30	0.85
75	4.48	0.78	10.35	0.94
90	5.07	0.89	8.22	0.62
100	5.50	0.88	14.30	2.20

Throughput numbers show a larger variation compared to the previous test. Total number of requests that server could process was about 1.35 million a day. Top CPU usage was measured to be around 98% during these tests.

Concurrent User (per 2 sec)	Throughput (req per sec)
25	16.5
40	12.4
50	13.8
65	16.0
75	16.4
90	17.4
100	17.2

Connection to a private calculation engine with session

Sessions were enabled in this test. So each request included authentication and a session ID which would need to be closed after the request is complete. As expected, this test yielded the slowest response times, especially in higher concurrent user scenarios.



Minimum response times were measured as high as 1.46 seconds, which is longest of all three tests.

Concurrent User (per 2 sec)	Average	Min	Max	Std
25	1.33	0.86	3.47	0.23
40	2.71	0.96	4.73	0.30
50	3.21	0.75	7.82	0.34
65	3.81	0.89	12.63	0.54
75	4.46	1.38	7.45	0.44
90	5.82	1.46	9.97	0.80
100	6.30	1.27	8.66	0.82

Throughput numbers were relatively stable and consistently lower than the other two tests. A larger variation compared to previous test. Total number of request that server could process was about 1.29 million a day. Top CPU usage was measured to be around 90% during these tests, which indicates a larger portion of a response went to non-CPU operations.

Concurrent User (per 2 sec)	Throughput (req per sec)
25	14.8
40	14.6
50	15.2
65	15.4
75	15.4
90	14.9
100	14.6