**Synchronous and Asynchronous Design**

**Enterprise Architecture Design CA1**

**David O Callaghan**

**X00159351**

**May 2020**

**Assignment Part 1**

Implemented on the basis of Labs. Source code is located here : <https://github.com/x00159351/CA1/tree/master/EADes-CA1>

Screenshots of Website front ends in Figure1 and Figure 2 below.

URL Async : <http://35.228.4.237:31080/>

URL Sync : <http://35.228.232.138:31916/allthenews?style=plain>

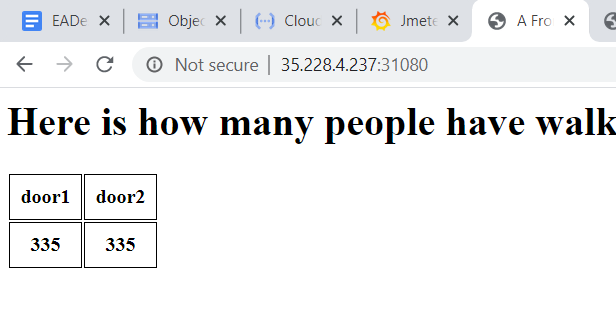


Figure 1 – Asynchronous Website (Author - 2020)

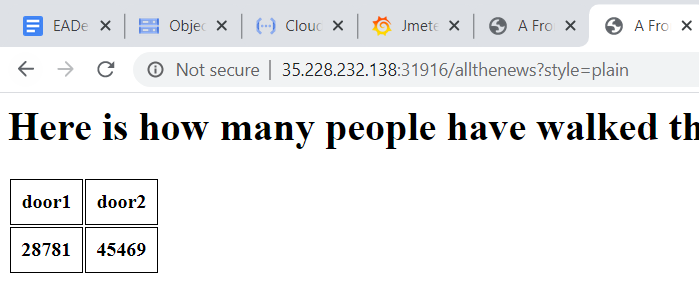


Figure 2 – ASynchronous Website (Author - 2020)

**Assignment Part 2**

**Part 1 :** Python script to run curl and measure response times created as a function in GCP(see Source Code). Calls script for creating graphs, output below.

URL Function : <https://us-central1-future-aurora-269520.cloudfunctions.net/ComparisonAsyncSync>

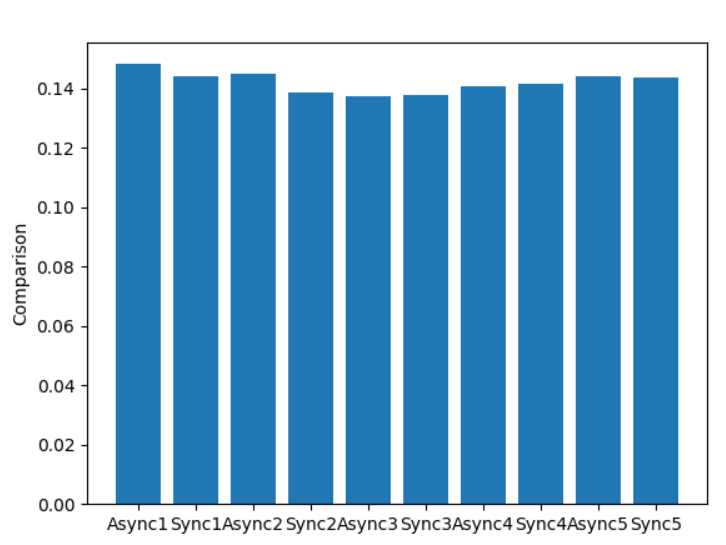


Figure 3 – Response comparison Sync/Async graphed in function (Author - 2020)

Jmeter script executed for comparison via local machine and pushing values to Influx DB(Source code contains jmx file). Values displayed in Jmeter graph and in Grafana. Timings taken locally from curl script(curl -w "@curl-format.txt" -o NUL -s http://35.228.232.138:31916/allthenews?style=plain?[1-100]) and graphed in excel for comparison. Differences between the results suggest curl locally is fastest, followed by pycurl run as a function and finally jmeter.

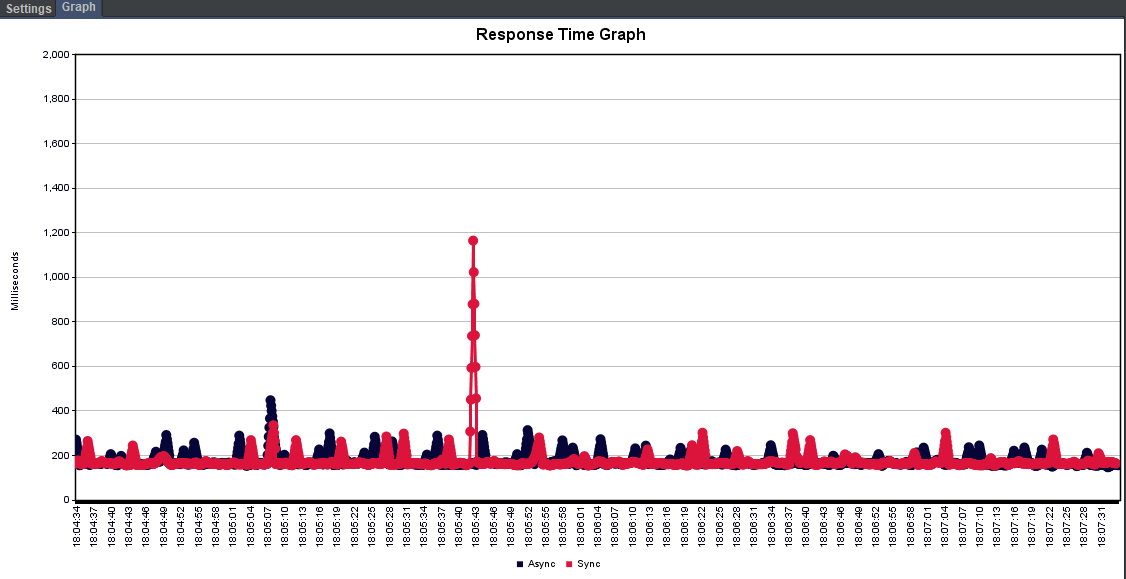


Figure 4 – Response comparison Sync/Async graphed in jMeter (Author - 2020)

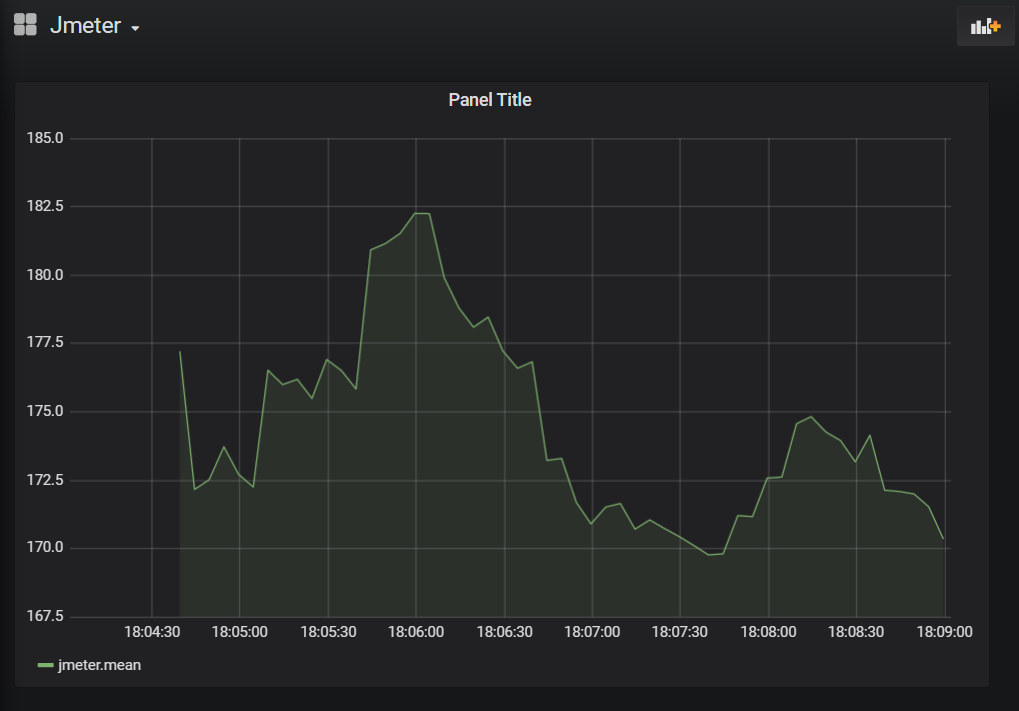


Figure 5 – Response comparison Sync/Async graphed in Grafana (Author - 2020)

Figure 6 – Response comparison Sync/Async graphed in excel (Author - 2020)

**Part 2 :** To plot the Async response as a function of the frequency of events and polling, pods were updated directly on GCP and a Pycurl script run 100 times to get an average response time for each scenario. Values were manually updated in the json and pushed to the graphing function via postman. Results are shown in figure 7 below.

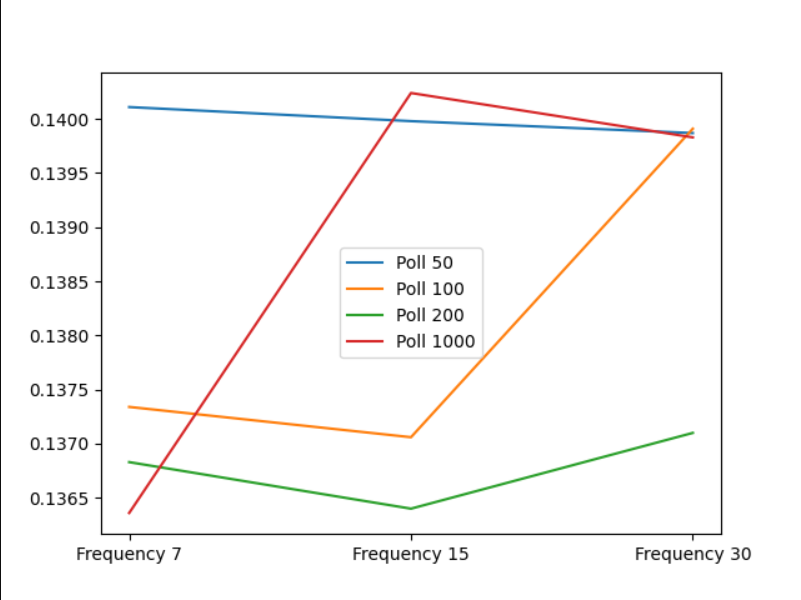


Figure 7 – Response comparison of Async with various polling and frequencies graphed in function (Author - 2020)

**Part 3 :** Pods were deleted using kubectl command : kubectl delete pod ‘name of pod’

Timings for restart were measured in seconds from the log file in GCP

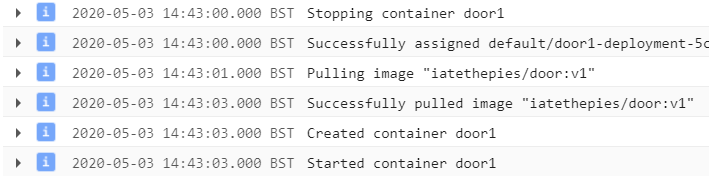


Figure 8 – Logging in Google Cloud to confirm restart time of Pods (Author - 2020)

Values were documented and entered manually in postman and pushed to the graphing function to get screenshot below. Excel graph created for comparison.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Service | Test 1 stop | Test 1 start | Test 2 stop | Test 2 start | Test 3 stop | Test 3 start | Average |
| Sync | G - Sync | 14:33:48 | 14:33:49 | 14:26:54 | 14:26:56 | 14:28:24 | 14:28:25 | 1.333333 |
| A - Sync | 14:30:42 | 14:30:43 | 14:31:52 | 14:31:53 | 14:32:50 | 14:32:51 | 1 |
| B - Sync | 14:35:17 | 14:35:18 | 14:34:34 | 14:34:38 | 14:37:52 | 14:37:53 | 2 |
| Async | G - Async | 14:44:50 | 14:44:51 | 14:46:40 | 14:46:43 | 14:48:09 | 14:48:10 | 1.666667 |
| A - Async | 14:43:00 | 14:43:03 | 14:46:15 | 14:46:16 | 14:48:20 | 14:48:21 | 1.666667 |
| B - Async | 14:44:38 | 14:44:39 | 14:46:30 | 14:46:31 | 14:48:29 | 14:48:30 | 1 |

Figure 9 – Stopping and Starting times of each pod averaged over 3 runs (Author - 2020)

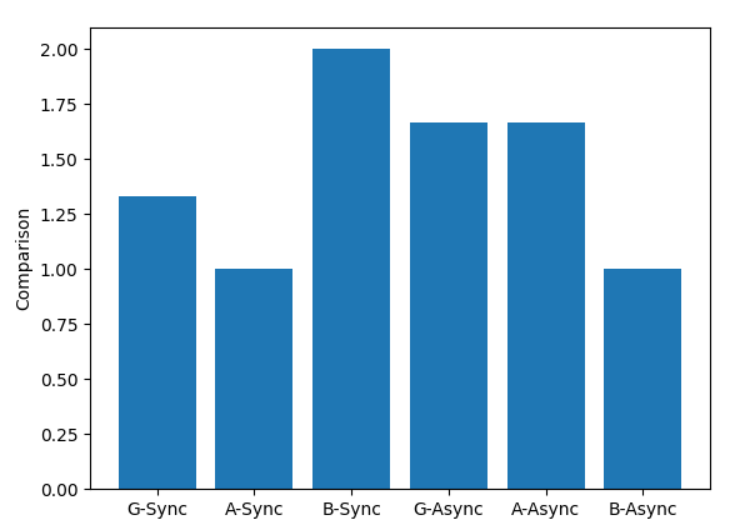


Figure 10 – Stopping and Starting times of each pod graphed by function after push from Postman (Author - 2020)

Figure 11 – Stopping and Starting times of each pod graphed in excel (Author - 2020)

# **Table of Figures**

[Figure 1 – Asynchronous Website (Author - 2020) 2](#_Toc39522012)

[Figure 2 – ASynchronous Website (Author - 2020) 2](#_Toc39522013)

[Figure 3 – Response comparison Sync/Async graphed in function (Author - 2020) 3](#_Toc39522014)

[Figure 4 – Response comparison Sync/Async graphed in jMeter (Author - 2020) 3](#_Toc39522015)

[Figure 5 – Response comparison Sync/Async graphed in Grafana (Author - 2020) 4](#_Toc39522016)

[Figure 6 – Response comparison Sync/Async graphed in excel (Author - 2020) 4](#_Toc39522017)

[Figure 7 – Response comparison of Async with various polling and frequencies graphed in function (Author - 2020) 5](#_Toc39522018)

[Figure 8 – Logging in Google Cloud to confirm restart time of Pods (Author - 2020) 5](#_Toc39522019)

[Figure 9 – Stopping and Starting times of each pod averaged over 3 runs (Author - 2020) 6](#_Toc39522020)

[Figure 10 – Stopping and Starting times of each pod graphed by function after push from Postman (Author - 2020) 6](#_Toc39522021)

[Figure 11 – Stopping and Starting times of each pod graphed in excel (Author - 2020) 7](#_Toc39522022)