Summary of the work done from week 9 (19.08.2019) to week 12 (15.09.2019)

During the nineth week of the internship I ran few tests to check whether the regex filter was working correctly for different benchmark tests and whether it was giving all the metrics I needed. But, the filters were still having some errors and I fixed them. And, also the /metrics endpoint doesn’t give the throughput, but it gives the total requests sent to the server so far. Therefore, I collected that and I used two global variables to store the previous request count and previous time, then by using those values throughput can be calculated. After that, Ballerina-1.0.0 alpha3 was released. So, I converted my previous code into the new version. There were some syntax changes and the documentation was not finished at that time. I managed to fix some problems and for undocumented ones, I used the ballerina forum to get the solution and successfully converted all the code to Ballerina-1.0.0alpha3.

In the next week I created a Ballerina echo service and used it instead of the netty echo service which we already used for simple passthrough scenario. This is to check the impact of tuning two services at the same time. There was not much differences in the latency and throughput because both are simple use cases without much CPU utilization. Therefore I created a prime server that checks whether a number is prime and tested the performance of just that prime server with different prime numbers. After that changed the threadpool sizes of the prime servers and obtained the impact of threadpool size with the prime number used. These tests can take hours to complete. I added the prime calculation to passthrough service and did the tests, after that for echo service and finally for both and ran the tests for different threadpool sizes.

In the eleventh week, I ran these tests for 0.991 version and 1.0.0alpha3 version of ballerina in order to compare the performance of the two versions. It seemed that the alpha3 under the default threadpool size performed worse than 0.991 with a significant drop of throughput like 7000 requests/sec difference for 5 minutes test. Therefore, I analyzed the code of the new version. And, they have made a change to the threadpool implementation. They have removed the Threadpool executor and added only the non-blocking threads without a thread pool. And the default is set to two times that of the number of cores of the CPU. But, the default thread pool size of Ballerina 0.991 is 100. And ,these may be the cause for the performance drop.

During the twelfth week, I re ran the tests to make sure that the results are correct because, the results obtained by the Ballerina team showed that Ballerina 1.0.0 performed better than 0.991 version. By that time Ballerina 1.0.0 was released. But, still I got the results that showed 1.0.0 doesn’t perform better. Therefore, I made further analysis by using Java Flight recorder. But, there were problems in using it for Ballerina as it is meant for Java. The first problem is Ballerina can run only in Java 8, if its run with later versions it will start the server, but the server will not respond. Therefore, I have to use Java 8. But, at that time Flight Recording was a commercial feature, and it was not found in Open JDK 8.

Therefore, I used Oracle JDK 8, to do the Flight Recording. Flight Recording is open sourced from Open JDK 11 onwards. For, ballerina 1.0.0 it is possible to use the Flight Recorder as they have made it possible to convert the bal file into jar file, so that I can directly use the JVM arguments for Flight recording. But, for 0.991 version there is no way to create a jar file. Therefore, I have to set the Flags using JAVA_OPTS variable used for Ballerina configurations. But, still it didn’t work because the internal JDK used by 0.991 version was Open JDK 8, so I have to change it to Oracle JDK 8 and do the Flight Recording. I also, did tests to make sure that the performance of Open JDK8 and Oracle JDK 8 are similar. After, that I got the Flight recordings, Flame graphs and GC logs for further analysis.

……

Signature