**Statistical Debugging Report**

The purpose of this task was to identify and analyze a deliberate delay bug introduced into a client-server file transfer application. An intentional delay was introduced in the client code during certain conditions, and the goal was to use Statistical Debugging (SD) technique to understand the impact of this delay on the occurrence of failures during file transfer. The bug we decided to implement in our client server application was to add an intentional delay in the client code which occurs randomly in 90% of cases during the transfer. The predicate for this task was the intentional delay introduced during the client code execution, Execution data was collected, and it included whether the intentional delay occurred and whether the file transfer resulted in any failure. The 2 key metrics for this task included the failure rate and delay rate, failure rate is the percent of executions resulting in failures and delay rate is the percentage of executions in which the intentional delay occurred. All the data was exported as a csv file, if we were to run the 2 files which have a bug (BuggyServer.py and BuggyClient.py) it would execute normally with a delay, but it would also create an csv export of the logging data to include the name of text file, if delay occurred (true/false), and if failure occurred during file transfer (True/False). We conducted a sample execution with the buggy server/client code and the sample data can be found in the “Bug Documentation” folder labeled as BugData.csv, within this file we can see that most transfers had a delay and it was recorded, we also looked at the trace results using the Jaeger UI and can see that files were taking longer times than usual, screenshots for all traces can be found in the Jaeger Output folder and can be viewed as desired. The statistical debugging technique we tried conduct was the observation of program behaviours where we check a certain condition and it evaluates to true/false, this was done by looking at the intentional delay and see if it was occurring, a logger was used to record the value true/false to indicate if the intentional delay was running for that certain file. Once we realized that the intentional delay was working and causing the file transfer process to become pretty slow, we need to find a way to show how to correct this bug as well as prove that it was still not occurring. The approach we took was to not get rid of the bug completely instead we changed the delay from 90% to 0%, meaning the intentional delay will occur at a rate of 0% or never, the code can be found as BugFixedClient.py and BugFixedServer.py, in this code you can find the delay change and also when it is executed it will also generate a csv export of the logger info during the file transfer, we conducted a sample execution and can be found in the “Bug Documentation” folder called bugFixedData.csv and within that file we can see that the bug has been adequately fixed because the intentional delay header stated false for all files meaning that it was not affecting the file transfer process in any way. Also, to verify that the delay was not in effect we also looked at the trace info using the Jaeger UI and it was seen that the files were being transferred at a normal pace (screenshots can be found in Jaeger Output).