**MP5 – CSCE 313**

Ryan Walters

Garrett Haynes

**Introduction:**

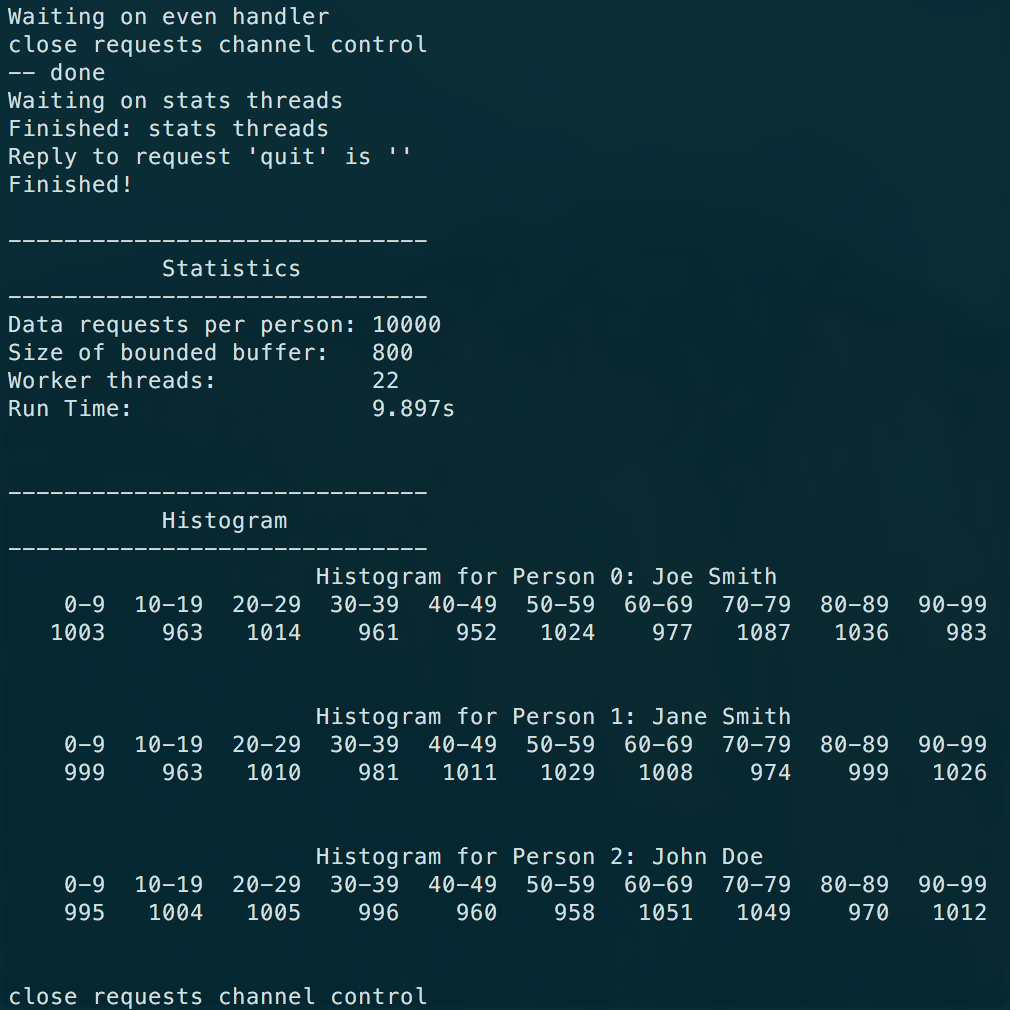
In this machine problem we replaced the worker thread functionality with a single thread event handler. This event handler managed multiple request channels to take care of the requests. We experience similar performance results when changing the number of request channels, which correlates to worker threads from MP4.

**Procedure:**

First we brainstormed how to implement an event handler in place of worker threads. After creating a plan of attack, we converted our pseudo-code to C++ code and tested it. After fixing a few issues, we managed to successfully implement our program.

**Result:**

The result of running the program can be seen in the following screenshot:

****

**Conclusion:**

**Measure performance:**

**Does increasing the number of request channels still improve the performance and if so by how much?**

Using 10000 requests and a buffer size of 800 we got the following results for runtime at different amounts of request channels:

|  |  |
| --- | --- |
| Number of Request Channels | Runtime (s) |
| 20 | 10.2 |
| 21 | 9.9 |
| 22 | 9.8 |
| 25 | 9.8 |
| 28 | 9.8 |
| 30 | 9.8 |

As you can see, to a certain extent, increasing the number of request channels does improve runtime, however this effect seems to bottle off at around 22 request channels for our amount of requests and buffer size.

**Is there a point at which increasing the number of request channels does not further improve performance?**

As explained above, after roughly 22 request channels, increasing the amount of worker threads does not give as much of a desired runtime improvement.