ECE 428/CS 425 Distributed Systems **Navid Mokhlesi, navidm2**
Spring 2020 **Amber Sahdev, asahdev2**
MP 0
Cluster: g22
URL: https://gitlab.engr.illinois.edu/asahdev2/ece428
Revision Number: 31a62561593afdf7e79f99c119203dc0fad33066

We learned Go Lang and wrote our MP in that language because of the concurrency features that Go has built in. To build the code run the following within the MP0 directory of the repository

```
> make
```

After running the makefile, we can run the logger as follows:

```
> ./logger [port number]
```

Similarly, we can run the node as follows:

```
> ./node [node name] [address] [port number]
```

To pipe the event generator output into node:

```
> python3 -u generator.py [hertz] | ./node [node name] [address] [port number]
```

We also need to install certain python dependencies to generate graphs as follows:

```
> pip3 install matplotlib
> pip3 install numpy
```

The python script to generate graphs can be run as follows

```
> python3 graph.py
```

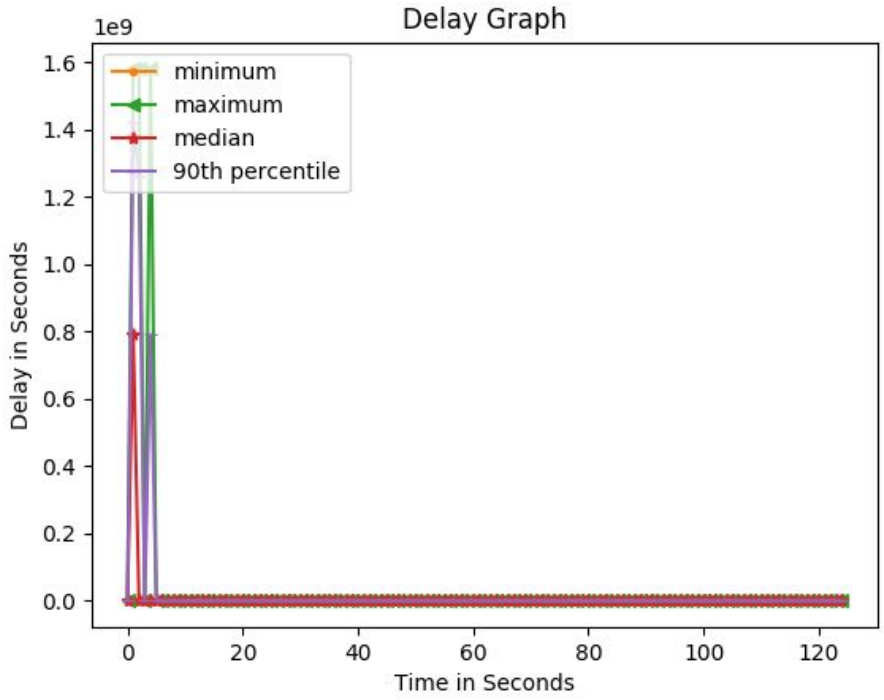
For analyzing our distributed logging system, we plotted out two graphs that measure the

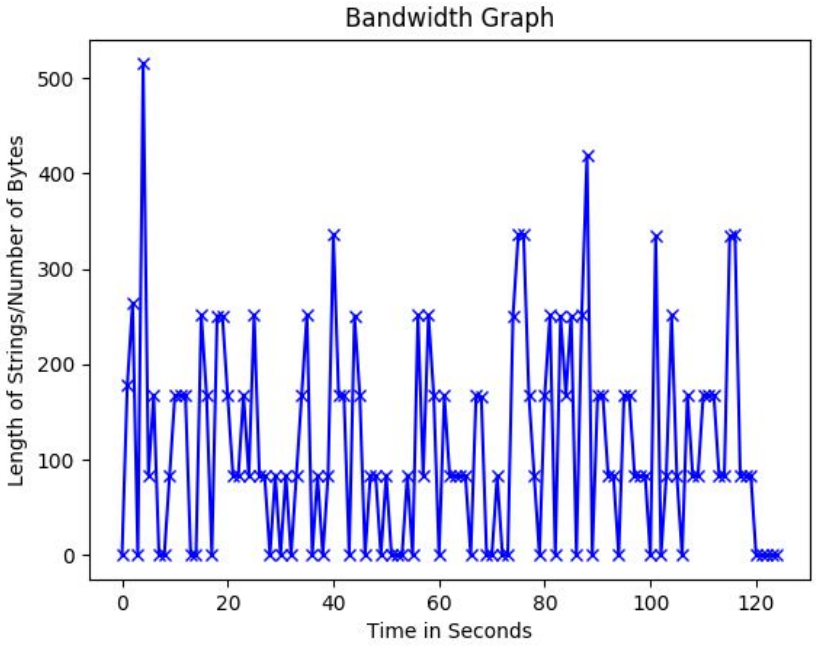
- Time delay between when the node sent the message and when the logger logged the message
- Bandwidth as the length of strings (number of bytes) logged per second

We were able to calculate the time delay by just subtracting the time while printing the logs by the time that our node sent us in the message. We then proceeded to plot the minimum, maximum, median, and the 90th percentile delay of our logging system. Similarly, bandwidth was just calculated as the sum of length of strings (number of bytes) logged per second.

Graphs of the evaluation as described above:

1. 3 nodes, 0.5 Hz each, running for 100 seconds





2. 8 nodes, 5 Hz each, running for 100 seconds