WRITEUP

Robert Hu

CruzID: ryhu

1 Testing

1. Created a shell executable file with multiple commands to run at the same time
2. COMMANDS USED
    a. curl localhost:[portnum]/filename
    b. curl -T local_file localhost:[portnum]/filename
    c. curl -I localhost:[portnum]/filename

2 Questions

1. Yes there is a difference in performance. The speedup is dependent on the number of threads available but, assuming the default 4 threads, the program is about 4 times faster as the four threads are able to work in parallel. In the original implementation, there is only one “thread” so all the work must be done in serial.

2. The bottleneck in my system is most likely to come from the dispatch part. The dispatch waits for threads to be available if there are no available worker threads to pass on the client socket descriptors. There is no concurrency in the dispatch part of the system. The threads are likely to be very concurrent and logging is not locked so all threads can write to the log at the same time. Increasing thread count and minimizing lock duration can increase concurrency.

3. Logging entire files is incredibly inefficient. If large systems like google logged the entire request, the entire file contents would take up too much memory. Most of the logging memory taken up in our programs are by file contents whereas the actual header logs are much smaller in comparison.