Performance Analysis

<REDACTED>– <REDACTED>– James Ryan – <REDACTED>

University of Victoria

SENG 468

# Performance Analysis Using Provided Workloads

Analysis of our project’s performance was gained using the provided user workload files, along with randomly generated workloads which we created using a workload generation script. These workload files were run using our system by a workload tester, and the resulting log file was submitted to the course website which provided verification of correct functionality and a Transactions Per Second (TPS) metric. This metric is what we used to assess our systems overall performance. The following chart shows the TPS of our system over time.

As seen above, the performance of our system improved over time with drastic variations. The variations over time typically resulted from some sort of configuration change, good or bad. As seen in the next chart, several the dips in TPS were of a result of some change to our configuration or transaction server which caused an increase in quotes to the quote server. From our understanding of computer networking, this is because of the overhead and delay caused by sending information of a network. Because of this, a recommendation we might pose if this were a product is to have our transaction servers co-located with the quote server(s), or as physically close to the quote server(s) as we could get them. This would be to reduce as much of the network delay as we can, by eliminating the need to travel over a network, or reduce the number of nodes the quote information needs travel through.

# Further Performance Analysis

Due to time constraints of this project, we were unable to perform and further performance analysis past the technique described above. If given further time for this project, we would like to perform profiling and other testing on our system. Particularly, we would focus on finding parts of our code which runs a lot and takes a lot of time which we could improve to increase our TPS. We do know that quotes to the quote server take significant amount of time, which we do not have control over. However, we believe it may be possible to better optimize our system around this.