Collected Latency Measurements

Two different setups were tested:

* 1 publisher, 1 broker, 1 subscriber
* 1 publisher, 1 broker, 5 subscribers

First, the end-to-end latency for the publication (and receipt) of one message was measured in both systems. The results are shown in Graph 1 below.

Graph 1. End-to-End Latency Avg for One Message (ms)

The end-to-end latency for a publication of one piece of information was greater in the architecture with more subscribers. We postulated that this was because the step between the broker and subscriber was larger since there needed to be more communication between processes, which would induce overhead. In order to test this, we measured the time from broker to subscriber

Graph 2. A breakdown of latency by stage. The blue indicates the architecture containing only 1 subscriber, while the orange indicates the architecture containing 5 subscribers.

The results of this experiment proved our hypothesis to be correct, as the only major difference in latencies was in the stage of communication between the broker and subscriber. It is also easily observable that the most time is consumed in the broker for the architecture with only 1 subscriber, which makes sense given that our implementation has the broker doing most of the message processing.

Finally, we compared our results of Assignment 3’s architecture (which contained capabilities to remember a topic’s history and maintain publisher priority) against the architecture of Assignment 2. The average time taken for one piece of information to be published and then received (1pub,1broker,1sub) in Assignment 2 was 0.987958908 msec, while in Assignment 3 it is 1.35000547 msec. This indicates that about .35 msec of overhead are a result of the new features implemented in this assignment.