Explanation of collected Measurements

Scenario for Graphs 1 and 2: In each architecture, each publisher ran a loop that kept publishing information for 1 second on a topic that each subscriber present was subscribed to.

Graph 1:

The latency for a publication of one piece of information to the broker heavily depended on the number of publishers present. In the scenario with 10 publishers, the latency is the highest, while in the other two scenarios which only use on publisher, the latency is substantially smaller. This may be attributed to overhead associated with connecting to the broker and maintaining ten publishing connections while using zookeeper to track the broker.

Graph 2:

The latency for receiving information by the subscribers in all three configurations increased compared to the previous implementation which lacked zookeeper, indicating that the addition of zookeeper has introduced significant overhead in our implementation. Like last time, the configuration that featured 10 publishers had the greatest latency, since in our implementation the broker receives all information and may act as an information bottleneck if the system is overloaded with information. Given that the latencies for both publishing information to the broker and receiving information to a subscriber increased likely due to zookeeper, we decided to measure how much overhead zookeeper was introducing by simply timing how long it would take for one piece of information to flow through the system (from publisher to subscriber).

Measurements on the latency for one piece of information (1 Publisher, 1 Broker, 1 Subscriber):

The average time taken for one piece of information to be published and then received in this setup was 0.987958908 msec, whereas in our implementation without zookeeper, the average time taken was 0.86007118 msec. This means that zookeeper is responsible for an extra 0.127887726 ms worth of overhead in our system in this setup.