Explanation of collected Measurements

Scenario for Graphs 1,2,3,4: 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:

Graph 2:

The amount of information published (meaning sent to the broker), or latency for a publication, was similar throughout the three architectures even as more subscribers added. Since this test was conducted locally, adding more publishers (on different threads) did not increase the amount of information published significantly (because the CPU power was instead divided proportionally amongst those threads). Nonetheless, this demonstrates that the amount of information published (sent to broker) is not a bottleneck in this system.

Graph 3:

Graph 4:

The amount of information received by the subscribers varied per configuration. Simply adding more subscribers increased the latency slightly (while keeping the number of publishers equal to 1) likely due to overhead associated with connecting to the broker and filtering. However, adding 9 more publishers (while keeping the number of subscribers equal to 1) drastically increased the latency for receipt of information of a subscriber, suggesting that the broker

(likely its connection overhead to publishers and the task of processing publisher information) is the bottleneck for this system. This makes sense given that in our model, all information is sent through the broker.

Scenario for graph 5: Publish exactly one piece of information and record the duration from publishing to receipt from subscriber(s).

Graph 5.

This shows that when publishing exactly one piece of information, the latency for end-to-end is somewhat similar to the latency for receipt of information shown in Graph 4 above (though the latency for 1 pub,1sub,1brok is slightly higher than it was in the scenario where it measured over one second, likely due to overhead associated with testing only 1 message than a full pipeline of published message, as well as overhead associated with calculating the current times). We suspect that if we tested that model with 10 publishers, the latency would be drastically higher (similar to the result in Graph 4).