

REPORT

BhanuKiran. Chaluvadi

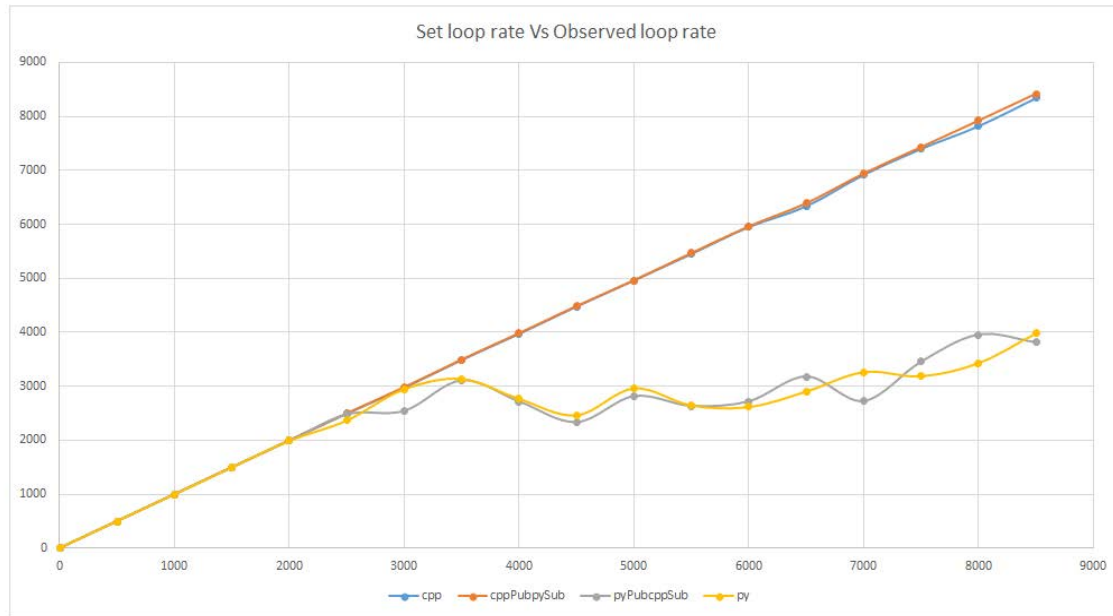
Performance Test: To find approximately the limit at which the desired rate does not coincide with real life for the 4 combinations

- cpp – Both publisher and subscriber in c++.
- cppPubpySub – Publisher in c++ and subscriber in python.
- pyPubcppSub – Publisher in python and subscriber in c++.
- py – Both publisher and subscriber in python.

Below is the table showing the median rate (hertz's) at which subscriber receives the message for initial consecutive observations for 15 seconds. Median is chosen to represent data over mean because of its appropriateness. .

Loop Rate	cpp	cppPubpySub	pyPubcppSub	py
10	10	10	10	10
500	500	500	500	500
1000	999	1000	999	999
1500	1499	1500	1497	1496
2000	1995	1998	1990	1991
2500	2493	2497	2479	2367
3000	2970	2982	2542	2949
3500	3488	3494	3109	3138
4000	3972	3982	2717	2767
4500	4478	4484	2334	2461
5000	4962	4962	2816	2964
5500	5454	5466	2635	2646
6000	5954	5961	2719	2617
6500	6339	6389	3181	2904
7000	6921	6940	2726	3266
7500	7403	7431	3455	3188
8000	7825	7920	3957	3432
8500	8344	8418	3815	3982

Plot:



The rate at which subscriber runs solely depends on the processor speed. Since subscriber loop does not have a loop parameter and the rate at which while loop repeats solely depends on processor and load on it. These values are purely relative to each computer. Better, the processor the better results.

From above graph, one can clearly observe that publisher and subscriber written in c++ outperformed their corresponding python version. I observed that c++ publisher and subscriber saturated at loop rate of 14000 than its counterparts written in python at 4000 on my computer. Not much distinction can be made between cpp and cppPubpySub, pyPubcppSub and py, this may be because the published string message is small. It may be possible to observe clear distinction between them if one starts publishing large data files like images.