Delft University of Technology

Quantitative Evaluation of Embedded Systems

IN4390

Lab 1

Authors: Jiaxuan Zhang(5258162)
Yiting Li(5281873)
Group ID: (26)

December 7, 2020

1 Global Clarification

Several global configuration that are used while dealing with these 4 questions should be clarified:

1. Question 1 and 3 are implemented in the same computer, while Question 2 and 4 are implemented in another computer

2. During each experiment, abnormal data can be found in the beginning or in the middle. Reasons for unexpected value in the beginning may be that the build up of communication need time during the early stage of communication, it can be partly illustrated by "cold start" phenomenon. Reasons for unexpected value in the middle may be that some turbulence(2.g. some unrelated programs or tasks) occurred during the experiment. Method to eliminate the influence of these unexpected value can be discarding or substituting them with other data. Details will be mentioned in each chapter.

2 Question 1

2.1 Background and Obtaining Data

The goal of question1 is to check the relation between ROS communication latency in two nodes and data size, by deriving a box plot diagram. To get the data we need

1. install the bash in the root directory and modify the two .cpp in the /src file.

2. run the two nodes to obtain publishing time and subscribing time.

3. run the calculating cpp to get the transport time and store these data. Clean out all data then do more experiment.

Overall, 10 groups of data is ready for further analysing.

2.2 Modifying, Displaying and Interpreting Data

In fact, those data are not so smooth, we can do more measurement in one round or we can substitute those abnormal data with original median.

Then we can draw the box plot of transporting time of each data size, all the data has the same growing trend, as a result, only one example will be listed here.

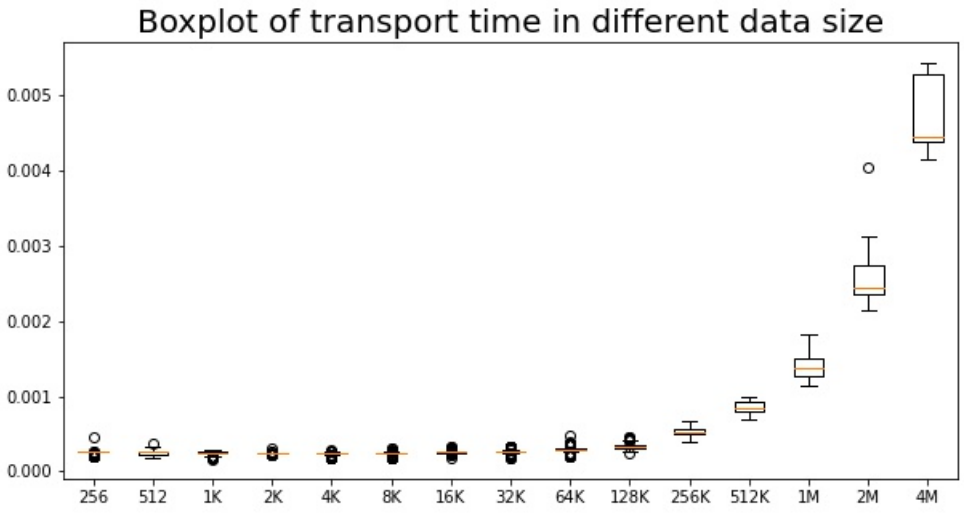


Figure 1: box-plot of transport time in different data size

The trend of transporting time is obvious, transport time grows similarly exponentially as data size increases. Some of the DDS policies can reduce the transport time, but the overall trend is the same, with the increasing of data size, ROS need more time to publish and subscribe them, that is the result of the increasing trend.

3 Question 2

3.1 Assumption

In order to calculate the Confidence Interval, we should first assumed a possible distribution for the data samples. Ttwo assumptions is raised here:

1. The possible distribution is Normal Distritbution with mean μ and variance σ^2
2. The samples are independent and identically distributed

So this question converts to calculation of the mean of the assumed distribution according to the given data set.

3.2 Implementation

The calculation of confidence interval is realized by Python script. It can be divided into two main steps.

1. load data and transform data to pandas data structure with function `pandas.DataFrame()`
2. use function `researchpy.summary_cont()`to obtain the confidence interval of parameters.

3.3 Result and Explanation

The result is as shown in Table 1. According to the mean value of data, the upper bound and lower bound are symmetrical, that is, the difference between upper bound and lower bound of 80% confidence interval is less than that of 98% confidence interval. The reason for that is the confidence interval indicates the possibility that the evaluated parameter is located in an interval, so the higher the confidence, the wider the interval for the same parameter with same data set.

0.8		0.98	
0.000921	0.000962	0.000904	0.000979

Table 1: Q2: Confidence Interval of μ

4 Question 3

4.1 Background and Obtaining data

The impact of background workload on the latency of ROS's communication should be evaluated. To achieve that, real-time priorities of ROS nodes and background workload need to set.

Background workload can be set easily by running the given artificial_load package, which will run infinitely until a stop command.

Setting real-time priorities can be done easily by only not define real-time in .cpp file. The other step is similar to Question1.

4.2 Modifying, Displaying and Interpreting Data

Same as the Question1, some data are so abnormal that brings a huge bias to final result, we use the same way to eliminate those data. By doing that, we can get a nicer data set, one example is shown as follow:

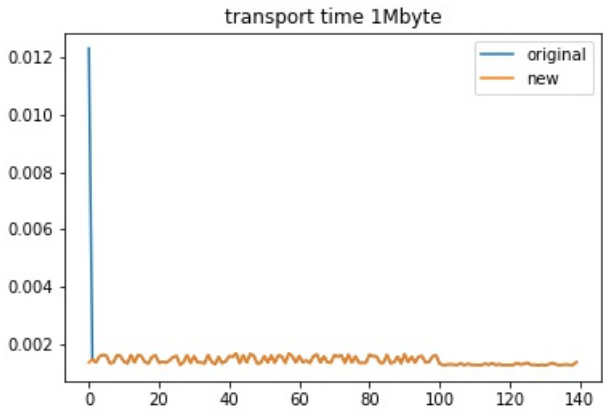


Figure 2: comparison of original and modified data

After experiment and modifying data, 4 groups of data is presented, we can draw a box-plot to analyse them, the diagram is shown as follow:

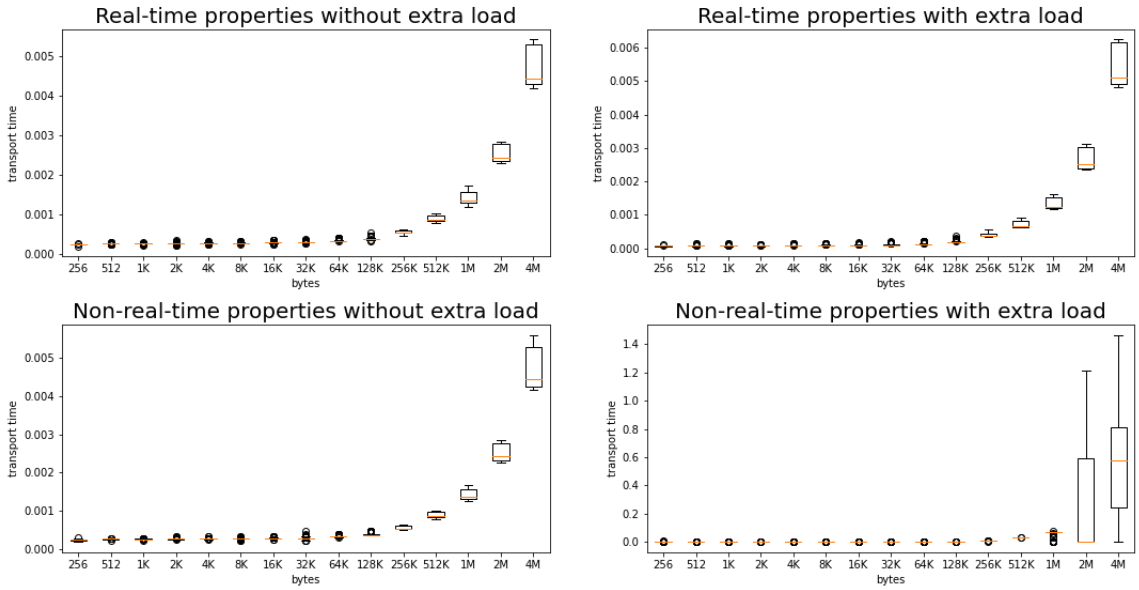


Figure 3: comparison of all data

From the Figure 3, the increasing trend of transport time is also easy to notice, which means that transport will increasing when data size that is been conveyed increases.

Operating system can handle interrupt according to their priorities, one with high priority can be handle while those with low priority have to wait in the queue. We can find that the number of transport time is different.

1. when there is no other running work, the execution time is similar, the non-real-time group's transport time is slightly larger than real-time one. Because the relatively sufficient resources enable both programs to be executed quickly.

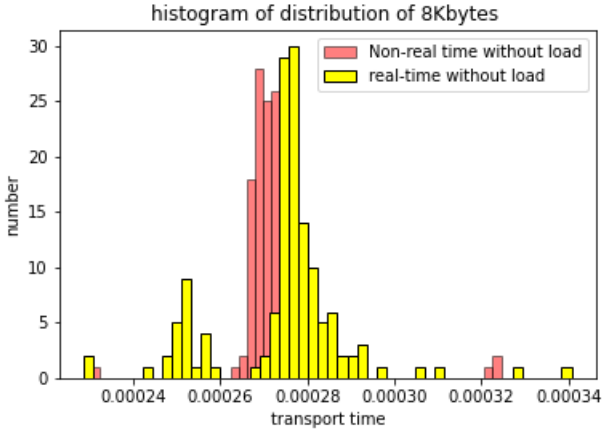


Figure 4: comparison of all data

2. when face a huge background workload, the non-real-time nodes runs much slower than real-time one. Because the real-time one has higher priority, so it won't wait too long in the queue. The result is shown as follow:

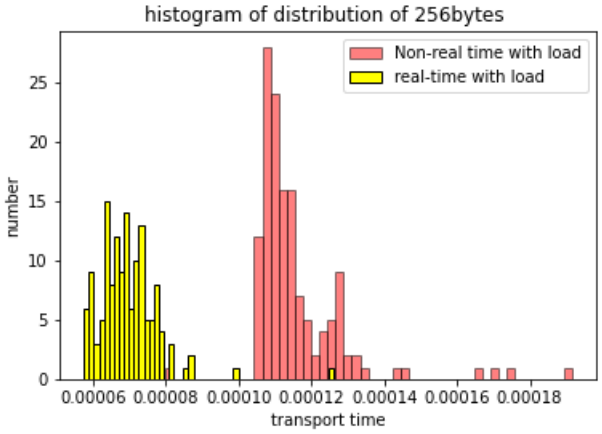


Figure 5: comparison of all data

3. when a node has high priority, its transport time doesn't vary much. Shown as follow:

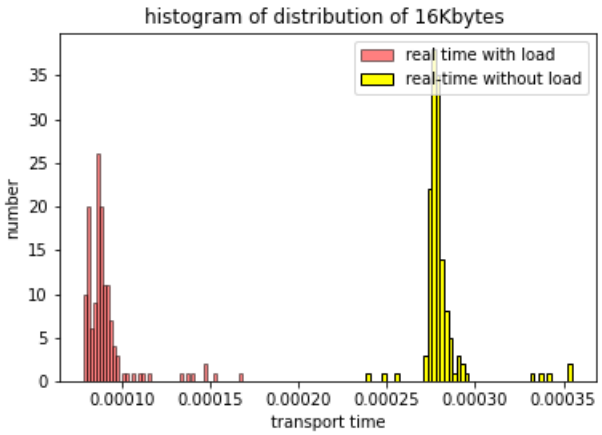


Figure 6: comparison of all data

We conclude that, if there is no background workload, non-real-time node and real-time node has a little difference. If there is background workload, transport time of a non-real-time node is much higher than transport time of a real-time node.

5 Question 4

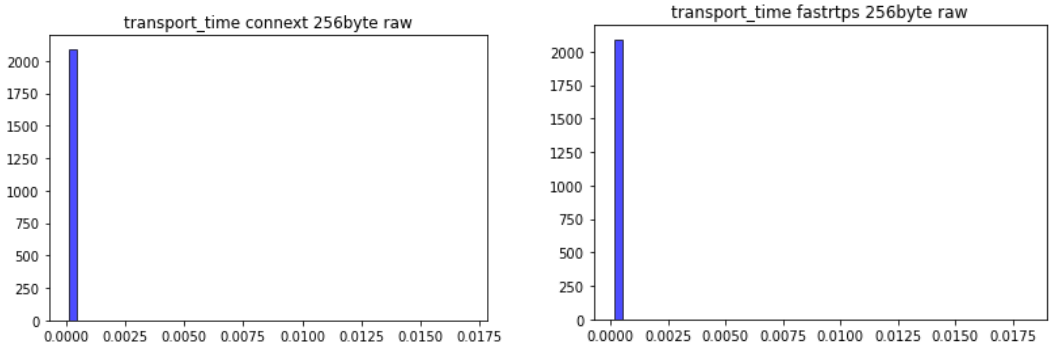
5.1 Background and Main Methods

In this Question, we need to compare the performance of 6 groups of configuration and by the result of histogram and ANOVA, a recommended configuration should be given.

First the source code is revised and the samples number is increased to 300 in order to get more precise analysis. Second, in order to get rid of random influence based on environment, we repeat measurement 7 times. The histogram is drawn mainly with `matplotlib` package of python and the ANOVA is implemented by `pandas` and `researchpy` packge.

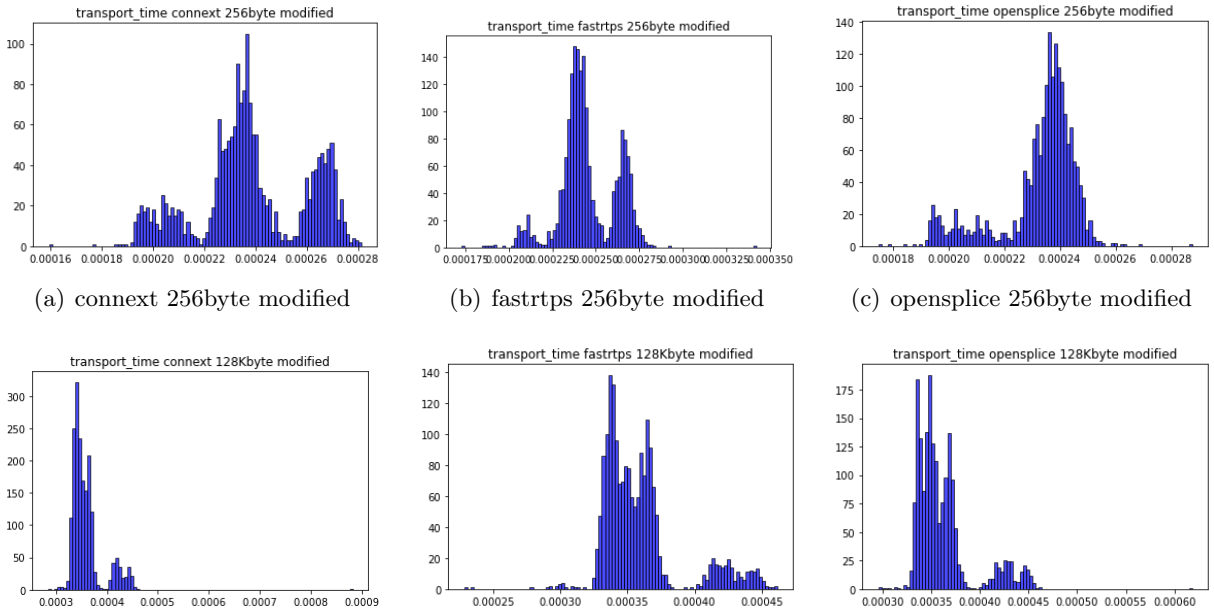
5.2 Histogram

While drawing histogram with the initial data, one of the result is shown as in Figure 7. There are some extreme values in the data set, after repeat the experiment several times, this extreme value always appear as the early stage of the data. Reasons for that were explained in the 1. In order to make the comparison result more accurate, we deleted the first 30 sampling points of each group of data. The result is shown in Figure 8



(a) connext 256byte raw (b) fastrtps 256byte raw

Figure 7: Raw Data



(a) connext 256byte modified (b) fastrtps 256byte modified (c) opensplice 256byte modified

(d) connext 128Kbyte modified (e) fastrtps 128Kbyte modified (f) opensplice 128Kbyte modified

Figure 8: Modified Data

5.3 ANOVA

By using Python package `researchpy` and `pandas`, ANOVA is done. The result of F test is shown in Table 2, In order to make the result in Sum Column more readable, each data has been multiplied by 100 when reading the initial data. The multiplication won't affect the F-test result, because F-test only care about the ratio between different components.

	Sum	degree(freedom)	F-value	p-value
DDS	0.00049	2.0	38.7112	1.759177e-17
Message	0.418128	1.0	66283.4579	0.0
DDS:Message	0.001152	2.0	91.3009	4.6313e-40
REsidual	0.071232	11292.0	NaN	NaN

Table 2: Q4 ANOVA

The final column in Table 2 implies the possibility we will refuse a correct hypothesis by mistaken according to the samples. From the table, the significance level of DDS is $1.759177 \times 10^{(} - 17)$ which means the possibility of false positive is $1.759177 \times 10^{(} - 17)$, i t is very small,so we have great confidence to say the initial hypothesis that $\alpha_1 = \alpha_2 = \alpha_3 = 0$ is wrong, and moreover the DDS has a significant impact on the latency with a significance level $1.759177 \times 10^{(} - 17)$.

5.4 Results and Suggestion

From Table 2,we would see that the message size has the most significant influence. Besides message size and DDS method, the interaction between message size and DDS method also has a significant impact. In order to compare these parameters in detail, the calculation of 95% confidence interval of each parameter is shown in the Table 3. The result of ANOVA is also proved here, the higher bound of the confidence interval of 256Byte Message Size is smaller than the lower bound of 128KByte, which implies the difference between different message size is apparent. The interval of DDS methods partly overlapped, so the difference of that is less apparent than message size.

Although the delay is slightly higher, it seems that a message of 128Kbyte is able to carry more information than 256 bytes. But for robot control, our robot always needs to get environmental information quickly and accurately, and make timely response. That is to say, timeliness really matters. Meanwhile, we always need atomic and temporal instruction more than complex and predictive instructions. Therefore the 256-byte configuration is even better.

At the same time, the upper limit of 95% confidence interval for OpenSplice is lower than the lower limit of FastRTPS. Therefore, we have 95% confidence to assert that the latency of OpenSplice is lower than that of FastRTPS. Although the confidence interval of OpenSplice and Connext overlap, the lower and upper bounds of OpenSplice are the lowest when calculating the 95% confidence interval. Therefore, OpenSplice is preferable.

From the above reasons, my recommended the combination is DDS = OpenSplice, and the message size is 256byte.

DDS			Message	
connext	fastrtps	opensplice	256Byte	128Kbyte
[0.0296,0.0301]	[0.0300,0.0304]	[0.0295,0.0300]	[0.0238,0.0239]	[0.0359,0.0361]

Table 3: Q4 Confidence Interval