WiP Abstract: Preliminary Evaluation for ROS2

Yuya Maruyama
School of Engineering Science
Osaka University

Shinpei Kato
Graduate School of
Information Science
Nagoya University

Takuya Azumi
Graduate School of
Engineering Science
Osaka University

Cyber-Physical Systems (CPS) represent next generation distributed and embedded systems. Becoming complicated and diverse, CPS applications aim to monitor and control complex real-time phenomena. Robot Operating System (ROS), an open-source middleware for robotics development, has been widely used for CPS applications (e.g. an autonomous car). ROS provides a Publish/Subscribe transport, many libraries (e.g. Point Cloud Library), and tools to help software-developers create CPS applications.

However, ROS does not support to meet real-time requirements and runs only on a few kinds of OSs. Therefore, ROS is not suitable for real-time and embedded systems. Facing this problem, ROS is going to be significantly upgraded as ROS2 [1]. ROS2 considers following use cases: real-time systems, non-ideal networks, small embedded systems, and cross-platform.

Existing ROS (hereinafter referred to as ROS1) reached the time to be reconstructed with improving user interfacing APIs and utilizing new technologies (e.g. Data Distribution Service (DDS) [2]). Replacing the ROS1's transport system, DDS is an industry-standard real-time communication system as end-to-end middleware that provides reliable Publish/Subscribe transport.

The next generation communication system of ROS, DDS, is suitable for CPS due to its various transport configurations such as DEADLINE, RELIABILITY, and DURABILITY. It has multiple implementations; some small/embedded solutions reduce their library size and memory footprints. Developed by different DDS vendors, several implementations are used in mission-critical environments and verified by NASA. Therefore, ROS2 built on DDS obtains its reliability and flexibility.

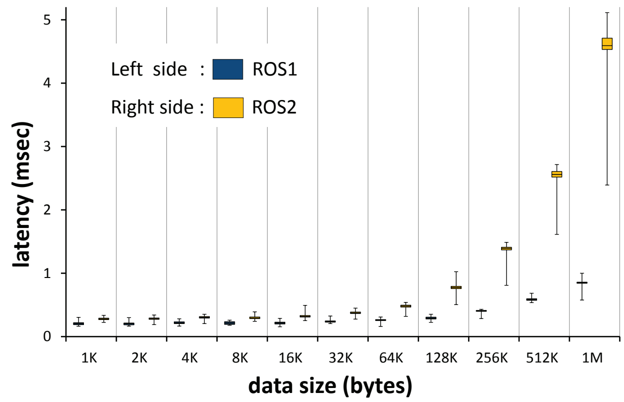

Figure 1: Comparison of transport latency variation in ROS1 or ROS2

Figure 1 shows data communication time between nodes on ROS1 or ROS2. ROS2 using DDS has various transport configurations, which causes the overhead examined in this research. ROS2 needs to convert data for DDS and abstracting DDS from ROS2 users. The overhead contains these transactions and the difference between ROS1 communication system and DDS. These influence varies by transport situations, data size, and DDS vendors.

In this research, we clarify the performance characteristics of currently-available data transport between nodes on ROS1 or ROS2 in various situations. Revealing the present capability of ROS2 depending on DDS vendors and DDS configurations, we explore and evaluate the facing constraints and potential of ROS2. The advantage and disadvantage of the current ROS2 are clarified with problems in the case ROS1 and ROS2 coexist. To the best of our knowledge, this evaluation is the first work for exploring the performance of ROS2.

1. REFERENCES

[1] Open Source Robotics Foundation (OSRF). ROS2. https://github.com/ros2.

[2] G. Pardo-Castellote. OMG Data-Distribution Service: Architectural Overview. In *Proc. of IEEE International Conference on Distributed Computing Systems Workshops*, pages 200–206, 2003.