

ROS2/DDS における FPGA を用いた Publish/Subscribe 通信処理の初期検討

大川 猛^{†1} 菅田 悠平^{†1} 木戸 剛正^{†1} 若槻 泰迪^{†1}
大津 金光^{†1} 横田 隆史^{†1}

概要 : ROS2 (Robot Operating System version 2) は、知的ロボットの分散ソフトウェア開発プラットフォームとして普及が期待されている。ROS2 における通信レイヤとして用いられる DDS (Data Distribution Service) は、分散ソフトウェアにおいて、通信するトピックごとに細かな QoS (Quality of Service) ポリシーが設定可能であることが特徴である。一方、FPGA(Field Programmable Gate Array)は、知的ロボットの実現に必要な画像認識処理を、高い電力効率での並列処理が可能であるが、高性能な回路の設計が難しいという課題がある。本稿では、FPGA を用いて DDS に準拠した Publish/Subscribe 通信を行うための調査および初期検討結果について報告する。

キーワード : ROS2 (Robot Operating System), DDS (Data Distribution Service), FPGA, Publish/Subscribe 通信

1. はじめに

知的なロボットにおけるソフトウェア構築のための開発プラットフォームとして、ROS (Robot Operating System) が普及している。ROS 公式 Wiki[1]においては、センサ・アクチュエータを駆動するためのノードや SLAM (自己位置推定マッピング) 処理ノードなど多くのパッケージが配布され、世界中のロボット研究開発に役立っている。一方 ROS はプロトタイプ用途としては使い勝手が良いが、製品ロボットに用いるには通信の質保証に問題がある[2]。そのため、製品ロボットを構築可能な開発プラットフォームを目標として ROS2[3]が提案され、2017 年 12 月にリリースされた。ROS2 においては通信の QoS (Quality of Service) を保証するために、産業界で実績のある通信ミドルウェア DDS (Data Distribution Service) [4][5]を採用した。これにより、無線等通信リソースの限られた環境での複数ロボット動作、組込みマイコン環境での動作、リアルタイム制御等の実現が期待される。

一方、ロボットには低消費電力であることが求められる。組込みマイコンを用いた ROS/ROS2 ノードにおいては、画像処理等の計算量の大きい処理を行うには性能が不足する。これに対して FPGA(Field Programmable Gate Array)は、低消費電力・高性能のハードウェア処理が可能であるが、ロボットに組み込むための開発が難しい問題があった。そのため、我々は ROS 準拠 FPGA コンポーネントを提案した[6][7]。これは、低消費電力・高性能な FPGA 回路を、システム内の他の ROS ノードと通信可能とすることで、ロボットに FPGA を容易に導入することを可能とするものである。

こうした背景を鑑み、我々は ROS2/DDS において FPGA を導入することを可能とするための検討を行った。本稿では、FPGA を用いて DDS に準拠した Publish/Subscribe 通信を行うための調査・初期検討結果について報告する。

2. ROS2/DDS 通信の調査

2.1 特徴

ROS2 は通信ミドルウェアとして DDS を用いることで、無線等通信リソースの限られた環境での複数ロボット動作、組込みマイコン環境での動作、リアルタイム制御等を目標とする。これに対して、従来の ROS においては、1 体のロボットを対象とし、PC のような豊富な計算資源と高速で安定なネットワーク環境を前提としており、通信のリアルタイム性・品質保証、スケーラビリティに難があった。

Publish/Subscribe 通信においては、概念として通信に参加する「ノード」同士は直接通信するのではなく、名前付けられた「トピック」を介して通信する。このトピックの名前と所在 (IP アドレス・ポート) の対応付けは名前解決を行うネームサーバにより行われる。ROS においては `roscore` プロセスがその役割を担っていたが、`roscore` が故障するとシステム全体がクラッシュする問題があった(単一障害点)。一方、ROS2 においては名前解決のメカニズムは単一のプロセスではなく各参加ノードに分散されているため、耐故障性が高いという特長がある。このメカニズムを実現する通信の方式は、RTPS 仕様として明文化されている。

2.2 RTPS 通信仕様の FPGA 化

DDS の通信仕様は、RTPS (Real-time Publish-Subscribe) プロトコル[4]として明文化されている。現在の最新バージョンは 2.2 で 2014 年に策定された^a。DDS のソフトウェア実装は、オープンソース・商用を含め複数入手可能であり、ROS2 からは実装を切り替えて使うことが可能である。

原理的には、FPGA が DDS の RTPS プロトコルに沿ったメッセージを送受信できれば、ROS2 のシステムに参加することが可能であると考えられる。しかしながら、RTPS v2.2 の仕様は A4 で 210 ページと膨大であり全てを FPGA 化することは容易ではない。

^{†1} 宇都宮大学
Utsunomiya University

^a なお、バージョン 2.1 は 2010 年に策定されているので、4 年間は変更がなかったということなので、仕様は安定していると考えられる。

3. ROS2/DDS 対応 FPGA コンポーネント実現に向けた初期検討

3.1 方針

膨大な RTPS の通信仕様をすべて FPGA 化することは困難であることから、最低限必要な機能に限定して実装を検討する。そのため、まずは ROS2/DDS のシステムを構築し、ノード間で行われる通信内容の分析を行うこととした。

3.2 DDS の通信内容分析環境

ROS2 動作時における通信内容分析を行うため、図 1 に示す ROS2/DDS 通信測定環境を構築した。システムは 2 台の PC (Ubuntu Linux 16.04) からなり、うち 1 台は仮想マシン環境とした。仮想マシン環境のホスト OS (Windows10) 上でパケットキャプチャ分析ソフトである Wireshark を動かし、2 台の Ubuntu Linux 16.04 間で行われる通信内容を分析した。使用した ROS2 のバージョンは ardent である。また、それぞれの PC 上には、ROS2 の配布物に含まれる examples の、talker ノードと listener ノードを動作させた。

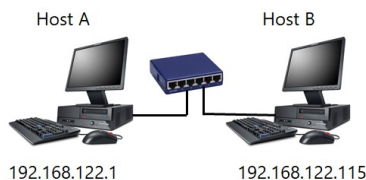


図 1 ROS2/DDS 通信測定環境の構成

Figure 1 Measurement setup for ROS2/DDS communication.

3.3 DDS の通信内容分析結果

ROS2 の talker/listener の例は、talker が”Hello World 0”, ”HelloWorld 1”・・・と文字列メッセージを publish し、listener がそれを subscribe して受信する、という内容である。

この通信内容分析においては、17 回の文字列メッセージ送信に対して、やり取りされたパケットは全部で 244 個であった。キャプチャされた RTPS パケットの例を図 2 に示す。このパケットは、”Hello World”を含む RTPS パケットであり、種別は INFO_DST, INFO_TS, DATA(p)である。

このパケット以外の RTPS パケット種別としては、HEARTBEAT, ACKNACK, DATA(w)のパケットが観測された。その他、IGMPv3 マルチキャストプロトコル、MDNS (マルチキャスト DNS) プロトコルのパケットが観測された。

このように、簡単なやり取りであっても大量のパケットがやり取りされるため、FPGA 化する際の最低限必要な機能をまず切り出す必要がある。

```

126 52.39662.192.168.122.115.192.168.122.1 RTPS 138 INFO_DST, INFO_TS, DATA
127 53.39691.192.168.122.115.192.168.122.1 RTPS 138 INFO_DST, INFO_TS, DATA
128 53.39694.192.168.122.115.192.168.122.1 RTPS 138 INFO_DST, INFO_TS, DATA
129 54.38953.192.168.122.1 192.168.122.115 RTPS 110 INFO_DST, HEARTBEAT
130 54.38961.192.168.122.1 239.255.0.1 RTPS 110 INFO_DST, HEARTBEAT
131 54.38964.192.168.122.1 192.168.122.115 RTPS 110 INFO_DST, HEARTBEAT
132 54.38968.192.168.122.1 239.255.0.1 RTPS 110 INFO_DST, HEARTBEAT
133 54.38972.192.168.122.1 192.168.122.115 RTPS 110 INFO_DST, HEARTBEAT

Frame 126: 138 bytes on wire (1104 bits), 138 bytes captured (1104 bits) on interface 0
Ethernet II, Src: RealtekU_26:a6:ae (92:34:00:26:a6:ae), Dst: fe:54:00:26:a6:ae (fe:54:00:26:a6:ae)
Internet Protocol Version 4, Src: 192.168.122.115 (192.168.122.115), Dst: 192.168.122.1 (192.168.122.1)
User Datagram Protocol, Src Port: 54951 (54951), Dst Port: 7411 (7411)
Real-Time Publish-Subscribe Wire Protocol

0000 fe 54 00 26 a6 ae 52 54 00 26 a6 ae 08 00 45 00 .T.&.rT.&....E.
0010 00 7c be 3c 40 00 40 11 06 6f c0 a8 7a 73 c0 a8 z1.<@..o..zs..
0020 7a 01 d6 a7 1c f3 00 68 76 3f 52 54 50 53 02 01 z.....h vRTPS..
0030 01 0f 01 0f fa f3 98 11 00 00 00 00 00 0e 01 .....zs.....
0040 0c 00 01 0f 0a 8f 10 36 00 00 00 00 00 09 01 .....V.....
0050 08 00 0e e2 18 5b 68 1d 55 61 15 05 2e 00 00 00 .....Lh..Ua.....
0060 10 00 00 00 0e 04 00 00 0e 03 00 00 00 00 01 00 .....:.....
0070 00 00 00 01 00 00 0f 00 00 00 48 65 6c 6c 6f 20 .....:..Hello
0080 5f 6f 72 6c 64 3a 20 31 00 00

```

図 2 キャプチャされた RTPS パケットの例

Figure 2 Example of captured packet of RTPS.

4. おわりに

本稿では、FPGA を用いて DDS/RTPS に準拠した Publish/Subscribe 通信を行うための調査および初期検討の結果を紹介した。まず、調査の結果として DDS/RTPS の仕様は膨大であり、FPGA で全てを実現することは開発コストとハードウェアコストの両面から困難であると考えられる。また、通信内容分析実験で得られた知見は、RTPS プロトコルを FPGA で実装する際には、膨大な仕様の中から必要最小限の機能を切り出すことが必要であるということである。今後、具体的な ROS2/DDS 準拠 FPGA コンポーネントの設計を進めていく予定である。

参考文献

- [1] ROS Wiki, <http://ros.org/wiki>
- [2] Yuya Maruyama, Shinpei Kato, and Takuya Azumi, "Exploring the performance of ROS2," Proceedings of the 13th International Conference on Embedded Software, ACM, 5 pages, 2016.
- [3] ROS2 Wiki, <https://github.com/ros2/ros2/wiki>
- [4] DDS INTEROPERABILITY WIRE PROTOCOL SPECIFICATION VERSION 2.2," <https://www.omg.org/spec/DDS-I-RTPS/>
- [5] G. Pardo-Castellote, "OMG data-distribution service: Architectural overview," Proc. of the 23rd International Conference on Distributed Computing Systems Workshops, IEEE, pp. 200-206. 2003.
- [6] Takeshi Ohkawa, Kazushi Yamashina, Hitomi Kimura, Kanemitsu Ootsu, Takashi Yokota, "FPGA Component Technology for Easy Integration of FPGA into Robot Systems," IEICE Transactions on Information and Systems (Special Section on Reconfigurable Systems), Vol.E101-D, No.2, pp.363-375, Feb. 2018. DOI: 10.1587/transinf.2017RCP0011
- [7] 菅田 悠平, 大川 猛, 大津 金光, 横田 隆史, "ハードウェア TCP/IP スタックを使用した ROS 準拠 FPGA コンポーネントの通信性能評価" 組込みシステムシンポジウム 2017 論文集, pp.62-69, 2017.