

## ROS2 通信機構の最適化に向けた IPC フレームワークの比較評価

森田 錬 松原 克弥 鈴木 昭二

公立はこだて未来大学

## 1 はじめに

ロボットやドローンなどのシステム開発基盤として、ROS<sup>1)</sup>の採用が進んでいる。ROSは、複数ノードのグラフ構造によるプログラミングモデルを採用しているため、ノード間通信の実装がシステム性能に与える影響が大きい。最新のROS2では、国際標準仕様であるDDS<sup>2)</sup>を採用することで、システムの目的に応じてノード間通信機能の実装を選択できる機構を持つ [1]。

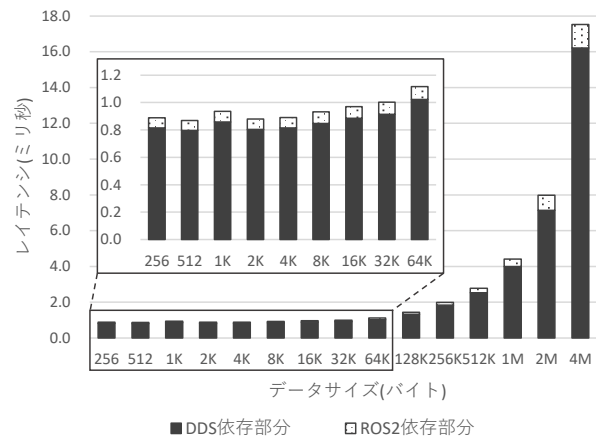
筆者らは、ROS2を用いて実装された制御システムにおいて、通信先やデータサイズなどの通信毎の特徴にあわせて複数のDDS実装を通信毎に動的に切り替える機構を実現した [2]。しかし、ROS2で現在使用可能なDDS実装では、すべてのデータ通信にUDPマルチキャストを用いている。それにより、通信先ノードがローカルホスト内にしか存在しない場合や受信するノードが存在しない場合にも冗長なネットワーク通信が発生することで、電力消費やレイテンシなどの通信性能に大きな影響を与えていることが推測される。

実際、ROS2システム<sup>3)</sup>上でデファクトスタンダードであるFast RTPS 1.7.0実装を用いた際の通信レイテンシの内訳を計測したところ、ROS2内の処理と比較して、DDS実装に依存する処理の影響が支配的であることがわかった (図1参照)。

本研究では、前述のDDS実装動的切替機構を活用して、ローカルホスト内ノード間通信に特化したDDS実装の実現を目指している。本稿では、本DDS実装の要となるIPC<sup>4)</sup>フレームワークの比較評価を行う。ROSの対象OSプラットフォームであるLinuxにおいて利用可能なUNIXドメインソケット [3]、OpenBinder<sup>5)</sup>とD-Bus<sup>6)</sup>を評価対象とする。

## 2 ROS2向けIPCフレームワーク評価基準

ローカルホスト内ノード間通信に特化したDDS実装を実現する観点から、評価対象とするIPCフレーム

図1: ROS2システム<sup>3)</sup>における通信レイテンシの内訳

ワークの評価基準を定義する。

第一の評価基準は、通信性能である。ロボットや組込み機器では、デッドラインが設定されたりリアルタイム制御が求められることが多い。デッドラインミスが起こった場合には、システム全体に致命的なダメージを与える。そのため、処理時間において高い割合を占めるレイテンシをなるべく小さくしなければならない。

二つ目は、メモリ・フットプリントの評価である。ロボットや組込み機器では、搭載される計算資源が限られてしまうため、メモリ・フットプリントの小さいIPCフレームワークが求められる。

第三の基準として、DDS仕様の実現可能性を評価する。ROS2のノード間通信機能を実現するDDSの通信モデルは、多対多の非同期メッセージングである。また、異種アーキテクチャ間通信に対応するためのシリアライズ機能を持つ。各IPCフレームワークの仕様と実装を分析することにより評価する。

## 3 評価

ROS2システム<sup>3)</sup>上でUNIXドメインソケット、OpenBinder<sup>5)</sup>、D-Bus<sup>6)</sup>およびROS2APIを利用して

**Preliminary Evaluation of IPC Frameworks for the ROS2 Inter-node Communication**  
Ren MORITA, Katsuya MATSUBARA and Sho'ji SUZUKI  
Future University Hakodate, 041-8655, Hakodate, Japan  
{g2118042, matsu, ssuzuki}@fun.ac.jp

<sup>1)</sup>Robot Operating System

<sup>2)</sup>Data Distribution Service

<sup>3)</sup>実験環境: Intel Core i5-6260U CPU, 8GB RAM, Ubuntu Linux 18.4.1 LTS (Kernel 4.15.18), ROS2 Ver. Bouncy Bolson

<sup>4)</sup>Inter-Process Communication, プロセス間通信

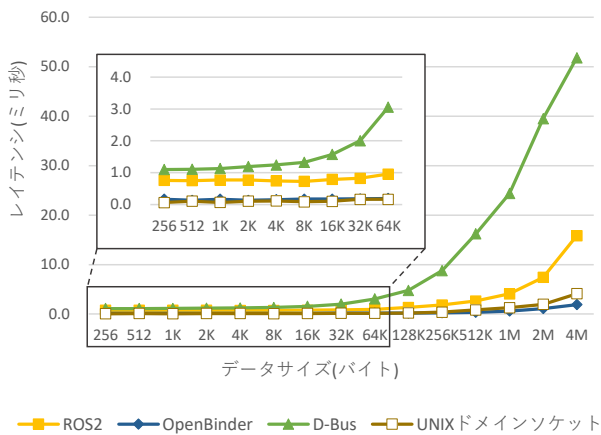


図 2: 通信性能の計測結果

ノード間通信を行う実験プログラムを実行し、通信性能とメモリ・フットプリントを計測した。通信性能の計測では、文字列型のデータを 256B から 4MB までの範囲でサイズを変化させつつ、10Hz の頻度で 100 回通信した際の平均 RTT を記録した。また、比較対象として、既存 DDS 実装のひとつである Fast RTPS 1.7.0 の通信レイテンシも計測した。メモリ・フットプリントの計測では、先の通信性能の計測に用いた送信側プログラムを実行し、256B のデータを送信している際の物理メモリの消費量を ps コマンド出力の RSS 値から取得した。DDS 仕様対応の評価は、各 IPC フレームワークのドキュメントとソースコードから、DDS 仕様として必要な Pub-Sub 通信モデルの実現可能性と送受信データのシリアライズ機能の有無を確認した。

### 3.1 通信性能

通信性能の計測結果を図 2 に示す。OpenBinder と UNIX ドメインソケットのレイテンシは、ROS2 のレイテンシより常に小さく、データサイズが増加した場合も変化量が小さい。D-Bus は、どのデータサイズにおいても、既存の ROS2 実装よりもレイテンシが大きい。

### 3.2 メモリ・フットプリント

メモリ・フットプリントの計測結果を表 1 に示す。OpenBinder と UNIX ドメインソケットは、D-Bus と比較してメモリ・フットプリントが非常に小さい。D-Bus は、他の IPC フレームワークの約 15 倍のメモリ・フットプリントとなり非常に大きい。

### 3.3 DDS 仕様対応

D-Bus では、dbus-daemon を介して通信することで Pub-Sub 通信を実現できる。また、DDS で利用される

<sup>5)</sup>OpenBinder Version 8, <http://www.angryredplanet.com/~hackbod/openbinder>

<sup>6)</sup>D-Bus 1.12.2, <https://dbus.freedesktop.org/>

表 1: メモリ・フットプリントの計測結果

IPC フレームワーク	物理メモリ消費量
UNIX ドメインソケット	1,864 KiB
OpenBinder	1,924 KiB
D-Bus	29,200 KiB

データ型に対応したシリアライズ機能も持つ。

OpenBinder は、同期型もしくは非同期型の一対一通信が可能である。しかし、複数の受信ノードが存在する多対多通信が実現されていない。しかし、位置非依存 ID から通信先アドレスへの変換を行う管理ノードにおいて、1ID に対して複数アドレスに対応する機能を加えることで多対多通信を実現できると考える。また、DDS で利用されるデータ型に対応したシリアライズ機能を持つ。

UNIX ドメインソケットも、多対多通信には対応していない。また、位置に依存した通信先を明示的に指定するため、多対多通信を実現するには、一つの通信を複数に複製する中継ノードを実装する必要があり、通信オーバーヘッドも大きくなる。また、シリアライズ機能も持たない。

## 4 おわりに

本稿では、ROS2 通信基盤の最適化を目的として、ローカルホスト内のノード間通信に最適な DDS 実装のための通信フレームワークを選定するために、Linux において利用可能な IPC フレームワークである UNIX ドメインソケット、OpenBinder と D-Bus の通信性能とメモリ・フットプリントを比較評価し、また、各 IPC フレームワーク上での DDS 仕様の実現可能性を検討した。計測結果と DDS 仕様対応の検討結果から、今回対象とした IPC フレームワークのなかでは、ローカルホスト内ノード間通信手段として OpenBinder が最適であることを確認した。今後、OpenBinder による Pub-Sub 通信方式の実現手法を考案し、OpenBinder を用いた DDS 実装を実現したい。

## 参考文献

- [1] Maruyama, Y., Kato, S., and Azumi, T.: Exploring the performance of ROS2, Proc. EMSOFT '16, pp.110, ACM (2016).
- [2] 森田錬, 松原克弥: ROS2 における通信特性に応じた DDS 実装の動的選択機構の実現, 研究報告組込みシステム, Vol.47, No.2, pp.1-6 (2018).
- [3] 藤田智哉: aibo における ROS の利用とソニーの取り組み, ROSCon JP 2019(オンライン), 入手先 <<http://roscon.jp/>> (参照 2019-01-05).