

CONFERENCE DIGEST

ロボティクス・メカトロニクス講演会2012
2012 JSME Conference on Robotics and Mechatronics

ROBOMECH2012 IN HAMAMATSU

グリーン&ライフイノベーションで未来を拓くロボティクス・メカトロニクス

Robotics and Mechatronics
- Pioneer the Future with Green & Life Innovations -



浜名湖(弁天島)



浜松城



家康公像

May 27 Sun. - 29 Tue., 2012 ACT CITY HAMAMATSU

主催 一般社団法人 日本機械学会 ロボティクス・メカトロニクス部門
The Japan Society of Mechanical Engineers, Robotics and Mechatronics Division



DDS による高速通信・フレキシブルな RT ミドルウェアベースロボットシステム構造

Building flexible and high-speed RT-Middleware-based robot systems using DDS

BIGGS Geoffrey (産総研) 安藤 慶昭 (産総研) 神徳 徹雄 (産総研)

Geoffrey BIGGS, National Institute of Advanced Industrial Science and Technology
Noriaki ANDO, National Institute of Advanced Industrial Science and Technology
Tetsuo KOTOKU, National Institute of Advanced Industrial Science and Technology

Data Distribution Service (DDS) is an open specification from the Object Management Group for topic-based communications in distributed environments. It is particularly well-known for its type safety and suitability to real-time applications. Several high-quality implementations exist, and their compliance with both the wire protocol standard and the API standard makes them interoperable and interchangeable. This paper describes the implementation of a port type for OpenRTM-aist that uses DDS as the communications layer for data transfer between components. It demonstrates the speed brought by DDS through the implementation of a high-throughput point cloud processing application.

Key Words: Distributed system、システムアーキテクチャ、DDS

1. はじめに

現在、高度なサービスロボットの開発をより効率的に行うため、OpenRTM-aistをはじめとするソフトウェアプラットフォームの研究開発が盛んに行われている。一般にこうしたソフトウェアプラットフォームには、モジュール化のためのフレームワーク、通信ミドルウェア、開発ツールやライブラリ等が含まれる。

OpenRTM-aist において、データポートはコンポーネント間の相互作用を行うための最も基本的かつ重要な機能であり、以下の要求を満たすべきである。

- プロセッサタイムや帯域幅などのリソースを使いすぎではない。
- 制御アルゴリズムの時間制限を果たすことなどのためにリアルタイムの転送を行う。
- 開発者に制限を設けないように、開発者が定義するデータ型をサポートする。
- 開発者が通信方法を選択することを可能にする。

現在、OpenRTM-aist では、データポート間のデータ転送に CORBA[4] を使用している [2]。また、高速化のために TCP 上の独自プロトコルによるデータ転送も提案されている [3]。両者はともに TCP による伝送であるため、周期的な連続データ送信時には UDP 等と比較すると抵抗率である。

本稿では、OpenRTM-aist 用の新しいポート型を示す。新しいポート型は「DDS」(Data Distribution Service [1]) 標準の実装を利用してトピックベース通信を行う。ポートの実装を記述し、大量のデータを処理するシステムで使うことを報告する。

2. Data Distribution Service

Data Distribution Service は、OMG (Object Management Group) が提供する分散システムの高速度通信の標準である。同じく OMG 標準である CORBA 等の RPC 等とは異なり、トピックベースの一方通信により分散システムを構成する。

DDS はロボットシステムに適したいくつかの特徴を有する。

- トピックベースの接続は、RPC 等のインターフェースにより接続と比べると緩やかな連続である。接続関係を構成するのは、送信端点、データ型、受信端点のみであり、送信者・受信者ともに、相手を具体的に背体する必要はない。一方 RPC は、プロバイダとコンシューマ間の接続にセマンティクスとステートを持つ。

- DDS のデータ通信は型安全である。トピックのデータ型は OMG IDL により定義され強く型付けされている。
- DDS は QoS (Quality of Service) をサポートする。データサブスクリバはレート、レイテンシ、歴史などの必要になるデータの特性を指定することが可能である。データパブリシャは提供するデータについて同様の特性を指定することが可能である。エラー対応のために、パブリシャまたはサブスクリバは QoS 保証を守らない場合、DDS ミドルウェアはアプリケーションを通知する。
- DDS は信頼できない通信パスがある場合に強い。
- DDS 標準はリアルタイムシステム向きである。実装もリアルタイム制限以内に通信する機能が持つ。
- DDS 標準は API とプロトコルを定義し、標準に従う実装はより簡単に交換できる。

現在、DDS の実装は複数である。

- Real-Time Innovations の RTI Connext DDS¹。
- ACE 元に作られたオープンソースの OpenDDS²。
- PrismTech のオープンソース OpenSplice DDS³。

3. RTMDDS

DDS を利用する OpenRTM-aist 用の新しいポート型を示す。QoS を含めてすべての DDS の機能が利用できる。

ポートのアーキテクチャは図に示される。現在は RTI Connext DDS 以外の実装はサポートされていない。

二つのポートが含まれている：

- トピックにデータを送信する出力ポート。
- トピックからデータを受信する入力ポート。

DDS の DataWriter 経由でパブリシャにデータを渡すことによって出力ポートはデータを送信する。パブリシャは QoS を管理する。

DDS の DataReader がサブスクリバにデータを渡されることによって入力ポートはデータの受信を行う。サブスクリバは同様のトピックを利用しているパブリシャからデータを受信する。パブリシャとサブスクリバは接続されて、データ通信によってトピック機能を実現する。

RTMDDS の出力ポートと入力ポートは OpenRTM-aist の標準ポートと同様に生成される。

¹<http://www.prismtech.com/opensplice/products/open-source-dds>

²<http://www.opendds.org/>

³<http://www.prismtech.com/opensplice/products/open-source-dds>

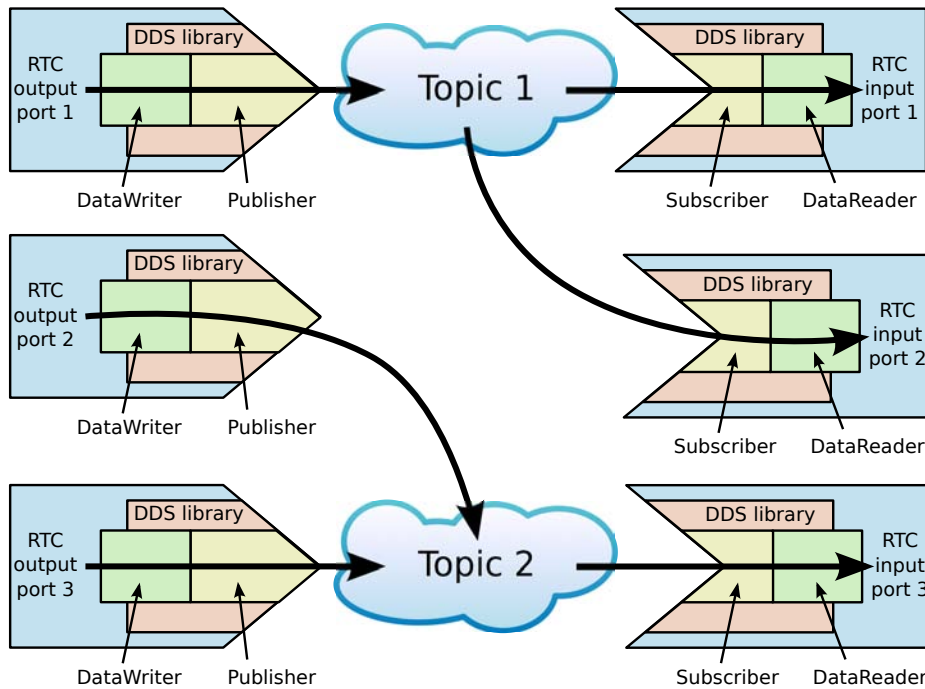


Fig.1 データ通信の為に RTMDDS のポートは内部に DDS を利用する。複数の出力ポートは一つのトピックに送信することや複数の受信ポートは一つのトピックから受信することも可能である。

3.1 トピック名

同様のトピックに送信または受信するポートは通信する。RTMDDS では、複数のポートは同時に一つのトピックに繋がることまたは一つのポートをトピックに繋がることが可能である。ポートをトピックに繋がることは RTSystemEditor も rtshell も利用することが可能である。

トピック名は普段開発者に決められる。例えば、オドメトリデータを通信するために「odometry」というトピックを生成する。選択されたトピック名はコネクションのプロパティで定義される。

代わりに、開発者がトピック名を RTMDDS に任せることが可能である。コネクションにある最初のポートの名の元にトピック名は作られる。

3.2 QoS 設定

RTI Connnext DDS では、QoS 設定は XML ファイルで定義することが可能である。RTMDDS は開発者に XML ファイルの選択するオプションを提供する。ポートの設定ファイル (rtc.conf に指定される) で XML ファイルを指定すると、ポートは接続される時点で QoS を設定する。

4. デモンストレーション

ポイントクラウド処理用のコンポーネント群に RTMDDS が利用されている。図 2 の用のポイントクラウドは大量のデータを持ち、提供するセンサーが高速で出力することが可能である。処理するパイプラインはできるだけ早く処理することが必要となる。図 3 に処理パイプラインの例が示される。

処理パイプラインのすべてのステップは単独のコンポーネントであり、PCL (Point Cloud Library [6]) からの一つのアルゴリズムを実装する。高効率で分解によって変わらない通信がある用に、コンポーネント間の通信は DDS が面倒する。DDS のおかげで処理パイプラインのいくつかのステップを別コンピューターに実行することは可能であり、ランタイムで処理パイプラインで使われているアルゴリズムを変更することが可能になる。

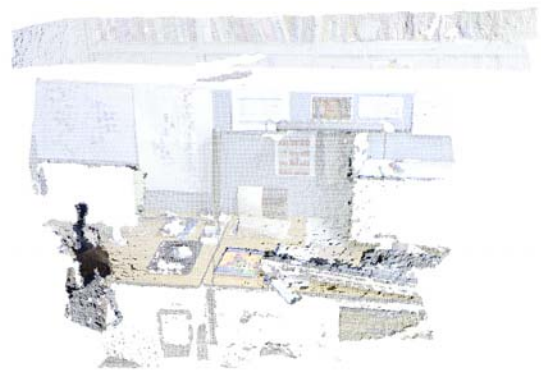


Fig.2 Microsoft Kinect からのポイントクラウドを処理するための処理パイプラインに DDS ポートは使われた。

高効率にポイントクラウドの処理を行うために、DDS メッセージの最大データサイズをあげることを含めて、いくつかの QoS 設定が必要となる。処理パイプラインは最新データだけを処理するつもりでトピックが保存する歴史は削減された。

5. おわりに

Data Distribution Service はコンポーネントベースロボットシステムの通信にとって大事な技術である。高効率、タイプセーフ及びトピックベース通信を提供する。トピックベースの特性によって RPC 通信で不可能ようにコンポーネントのつながりはより弱くなって再利用がより簡単になる。ロボットで、特に OpenRTM-aist で、よく使われるデータフロー通信にふさわしい。

現在、トピックベース通信を使うロボットミドルウェアは ROS 以外に存在しない。しかし、DDS のフィーチャーと ROS のフィーチャーは比べ物ではない。ROS は DDS のような QoS 機能やリアルタイム機能を提供しない。

RTMDDS は OpenRTM-aist を利用する RT コンポーネ

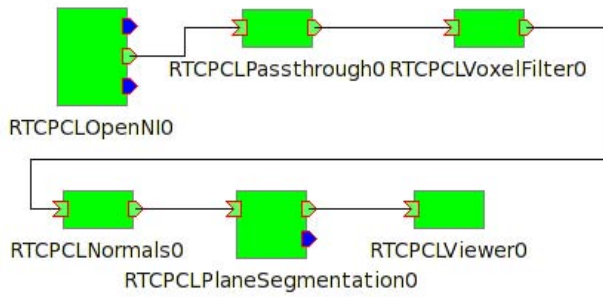


Fig.3 ポイントクラウドの処理を行うパイプライン。高効率とネットワーク透明性の為に各接続は DDS を利用する。

ントに新しいデータポートを提供する。RTMDDS によって、高パフォーマンスシステムの開発に RT コンポーネントは DDS のトピックベース通信と QoS 機能を利用することが可能である。DDS をサポートするロボットミドルウェアには OpenRTM-aist は最初である。

参考文献

- [1] (2012) OMG DDS Specifications. [Online]. Available: http://www.omg.org/technology/documents/dds_spec_catalog.htm
- [2] N. Ando, T. Suehiro, and T. Kotoku, "A software platform for component based rt-system development: Openrtm-aist," in *SIMPAR '08: Proceedings of the 1st International Conference on Simulation, Modeling, and Programming for Autonomous Robots*. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 87–98.
- [3] 安藤 慶昭, 神徳 徹雄, 安 芳次, 久保田 貴也, 大川 猛, 平野 聡, "RT コンポーネントの InPort/OutPort データ転送方法の多様化 - Raw TCP/IP Socket によるデータ転送 -", 計測自動制御学会 システムインテグレーション部門 講演会 2006 (SI2006), p.3B2-1, 2006.1.
- [4] (2012) CORBA 3.2. [Online]. Available: <http://www.omg.org/spec/CORBA/3.2/>
- [5] (2012) Common Object Request Broker Architecture (CORBA) Specification, Version 3.2: Part 1: CORBA Interfaces. [Online]. Available: <http://www.omg.org/spec/CORBA/3.2/Interfaces/PDF/>
- [6] (2012) PCL – Point Cloud Library. [Online]. Available: <http://pointclouds.org/>