Wii Fit Board と Sphero SPRK+による直感的なロボット制御のための MROS2 の活用

1020259 中村碧 指導教員:松原克弥

Aoi Nakamura

概要:ロボットソフトウェア開発において、ROS(Robot Operating System)の利用が増えている。ROS はクラウドサーバと連携し、分散型ロボットシステムを構築するのに役立つ。しかし、各機能モジュール(ノード)の配置はシステム稼働前に決定する必要があり、予測困難な状況変化で最適なノード配置が変わる可能性がある。クラウドとロボット間でのCPU アーキテクチャの違いにより、稼働中のノードの再配置は技術的に難しい。この研究では、ROS ノードの動的配置機構の実現に向けた、組み込みデバイス向け ROS 2ランタイム実装である mROS 2-Posix を評価する。mROS 2-Posix が ROS 2 と比較してマイグレーションに優位性があるかを明らかにし、動的配置機構の実現への影響を実験結果で示す。

キーワード:ロボティクス,mROS2

Abstract:

Keywords: Roboethics, mROS2

1 背景と目的

さまざまな産業向けやエンターテイメント関 連のロボットシステム, 自動車の自動運転技術 や IoT システムの構築においても、これらのソ フトウェア開発をサポートするフレームワーク として Robot Operating System (以下, ROS) の普及が増加している[1]。ROS のプログラミン グモデルは、システムの各機能を独立したプロ グラムモジュール(ノード)として設計するこ とにより、汎用性と再利用性を向上させて、各 機能モジュール間のデータ交換を規定すること で、効率的かつ柔軟なシステム構築を可能にし ている。たとえば、カメラを操作して周囲の環 境を撮影するノード,画像からオブジェクトを 識別するノード、オブジェクトのデータを基に 動作制御を実行するノードを連携させることで、 自動運転車の基本機能の一部を容易に実装でき る。

ROS のプログラミングモデルは、ロボット /IoT とクラウドが協力する分散型システムにおいても有効である。ロボットシステムのソフトウェア処理は、「センサー」「知能・制御系」「動力系」の3要素に分けられる[2]。クラウドロボティクスにおいて[3]、主に知能・制御系のノー

ドを高い計算能力を持つクラウドに優先して配 置することで、高度な知能・制御処理の実現を 促進し、さらに、必要な情報をクラウドに集約・ 保存することで、複数のロボット間での情報共 有と利用を容易にする。一方で,現行の ROS 実 装においては、各ノードの配置をシステム起動 時に設定する必要があり、先述のクラウドロボ ティクスのクラウドとロボット間の最適なノー ド配置を事前に設計する必要がある。しかし、実 際の環境で動作するロボットは、ネットワーク の状況やバッテリー残量の変動など、システム 運用前に予測することが困難な状況変化に対応 する必要があり、設定したクラウドとロボット 間のノード配置が最適でなくなる可能性がある。 このような状況変化に対する対応として、ノー ドを動的に再配置するマイグレーション機能の 実現が求められているが, 多くの場合でクラウド とロボット間の CPU アーキテクチャが異なり、 実行中のノードをシステム運用中にマイグレー ションすることは技術的に困難である。

菅らは、WebAssembly (以下、Wasm)を用いることで、クラウドとロボット間での実行状態を含む稼働中ノードを動的にマイグレーションする手法を提案した[菅さんの論文]. 課題として、

ROS 2を Wasm 化したことでオーバーヘッドが増大し、マイグレーション後、ノードの実行時間が大幅に増えてしまう問題が残った. 柿本ら[は、組込みデバイス向け ROS 2ランタイム実装である mROS 2-POSIX を採用し、ROS ランタイムの Wasm 化にともなうオーバヘッド増加に対処した. しかし、採用された mROS 2-POSIX はごく小規模なアプリケーション上でしか評価実験は行われておらず、今後も mROS2 POSIX を採用していく上でアプリケーションの規模を増加させた評価実験は必要不可欠である.

本研究では、クラウドとロボット間での実行 状態を含む稼働中ノードの動的マイグレーショ ンの実現に向けた mROS 2-Posix の評価を行い、 mROS 2が ROS 2と比べて動的配置機構の実現 のソフトウェア基盤としてどの点で優位性があ るのか明らかにすることを目指す.

2 mROS2-POSIX

そもそも mROS 2[4] は高瀬らが研究開発を 進めている ROS 2 ノードの軽量実行環境であ る. mROS 2 は中規模の組込みデバイス上で実 行されるプログラムについて, 汎用デバイス上 のROS2ノードと自律的に通信する機能を提供 している。このソフトウェア基盤によって、分 散型のロボットシステムへの組込み技術の導入 が促進できる. 組込みデバイスは計算資源が限 定的であるが、これを活用することによってリ アルタイム性の向上および消費電力の削減を図 ることができる. そして, mROS 2 が POSIX[] に対応したのが mROS 2-POSIX である. 本章 では、組込みデバイス向けの高校率なROS2通 信方式およびメモリ軽量な実行環境を提供して いる mROS 2-POSIX について、ソフトウェア 構成を中心に解説する.

図1 (a) は、mROS 2-POSIX のソフトウェア 構成を示している.上層から順に、まず mROS 2-POSIX アプリケーション層は、ユーザが実装する ROS 2 ノードに相当する.mROS 2-POSIX API 層および通信ライブラリ層は、ROS 2 の topic に相当する API および通信機能を提供する階層である.本階層は、ROS 2 のネイティブ なクライアントライブラリである relcpp と互換性を保つように設計している.なお、mROS 2 通信ライブラリでは、relcpp のうち pub/sub 通信の基本的な機能のみ実装されている.利用可

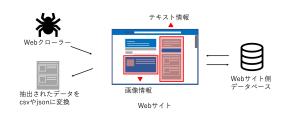


図 1 mROS 2と mROS 2-POSIX の内部構造

能な機能は制限されているものの、組込み技術を 導入する ROS 2 開発者は、汎用 OS 向けのプロ グラミングスタイルを踏襲しながら C++によっ て mROS 2 のアプリケーションを実装すること ができる.

Real Time Publish-Subscribe (以下, RTPS) プロトコルスタックには C++実装の embeddedRTPS[]が採用されている.メモリ領域の静的 確保など組込みデバイスでの稼働を想定して設計 されており、RTPS における Simple Participant Discover Protocol (以下, SPDP) および Simple Endpoint Discover Protocol (以下, SEDP) が 実装されているため、通信の自立性が実現でき る. UDP については組込み向けの C 実装である lwIP[]が採用されている. また, これらの階層は CMSIS-POSIX(Cortex Microcontroller System Interface Standard Portable Operating System Interface) API に依存している. 通信層の embeddedRTPS および lwIP は CMSAIS-POSIX に依存しており、図1(b) に示す mROS 2の CMSIS-RTOS[] を互換した層になっている. 最 下層にはハードウェアを抽象化したライブラリ がある.

mROS 2-POSIX の実行方式は、mROS 2が対応しているリアルタイム OS との相違がある. リアルタイム OS では、組込みマイコンを実行資源の管理対象として、タスク単位でアプリケーションが実行される. POSIX においてはタスクに相当する概念はプロセスであり、そこから生成されるスレッドを実行単位として処理が進行している.

したがって,mROS2-POSIX は図 2 に示す実行方式を採用している.mROS 2-POSIX の実行単位であるノードは POSIX のスレッドに対応付けらており,mROS 2 の通信ライブラリ機能を

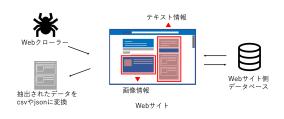


図 2 mROS 2-POSIX の実行方式

担う実行資源は POSIX のプロセスとして捉え, これを管理対象としている. 組込みマイコンで の通信処理におけるイベント割込みについては, POSIX 準拠 OS におけるブロッキング API の発 行に相当させて処理されている.

mROS 2と mROS 2-POSIX の機能構造の対応として、タスク管理機能は、スレッドの生成や開始などの POSIX 資源の管理機能に対応付けれている。メッセージキューおよびミューテックスによるスレッドの同期・通信および排他制御は、POSIX では Pthread によって対応している。メモリ管理機能は Alloc/Free で、時間管理は OS 時刻管理機能で対応している。 lwIP における UDP パケット処理は、POSIX における OS 資源の操作に関する機能で実現されている。

3 評価方針

mROS 2-POSIX を評価するにあったって, mROS2と同様のアプリケーションをROS2に 実装し、比較評価する. mROS 2 は、ROS 2 のネ イティブなクライアントライブラリである relepp と互換性を保つように設計されており、ROS2に 同様のアプリケーションを実装した場合でも期待 される動作をすることができる. 評価に使用す るアプリケーションは Wii Fit Board と Sphero SPRK+を用いたロボット制御アプリケーショ ンである. アプリケーションは、Wii Fit Board はユーザの体重移動を検知し、Sphero SPRK+ を前後左右に動かすことで、ユーザの体重移動 に追従するように動作する. 図3にアプリケー ションの構造を示す. Wii Fit Board に乗って いるユーザの座標をパブリッシュするノードで ある publisher _ wiiboard と、 wiiboard という トピックにサブスクライブし、受け取った値を もとにSphero_SPRK+を動作させる wiibaord $_{\rm sphero}$ $_{\rm sprk}$ の 2 つのノードにわけられる.

評価項目としては,mROS 2-POSIX が ROS 2 と比べてどの点で優位性があるのかを明らかに するため,以下の項目を設定した.

- 通信性能の評価
- メモリサイズの評価

通信性能の評価については、1回の通信をロボットがLEDを点灯するまでの時間とし、それを100回行い、その平均と最大値と最小値と標準偏差を取ることで性能比較を行う.

メモリサイズの評価については, アプリケーションのバイナリを size コマンドで計測しそれぞれtext, data, bss のサイズを比較する.

4 評価アプリケーションの実装

評価アプリケーションの実装として, mROS 2-POSIX と ROS 2 の両方で同じ機能を持つノードを作成する必要がある.

- 5 進捗と計画
- 6 結言
- 7 関連研究
- 8 知能システムコースにおける本研究の位置づけ

参考文献

- [1] 立教大学、"立教大学シラバス検索システム"、立教大学、[Online]. Available from: https://www.rep-rikkyo.com/class_rooms?day=c&hour=4. Accessed: 2023-10-25.
- [2] 東京学芸大学 (非公式), "東京学芸大学教室検索システム", 東京学芸大学. [Online]. Available from: https://u-gakugei-uoa.pages.dev/akitan/. Accessed: 2023-10-25.
- [3] SeleniumHQ, "Selenium Documentation", Selenium, 2023. [Online]. Available from: https://www.selenium.dev/documentation/en/. Accessed: 2023-10-25.