ロボットミドルウェアのためのホットスタンバイ型

障害回復アーキテクチャの提案

A hot standby fault-recovery architecture for robot middleware

5115E001-0　安里 太緒　　指導教員　尾形 哲也　教授

ASATO Tao　　Prof. OGATA Tetsuya

概要: ロボットミドルウェアを用いた開発が普及しつつあるにもかかわらず，障害耐性を実現する設計論はあまり議論されていない．従来研究では高速な障害回復手法のホットスタンバイの実現のため，各コンポーネント間通信にプロキシを配置する必要があり開発コストが増加する課題があった．またロボットミドルウェアの通信方式である遠隔手続き呼び出しにも対応しておらず，適用できるシステムが限られていた．我々はホットスタンバイかつ遠隔手続呼出しに対応した障害回復アーキテクチャを提案した．我々は実行時にシステムの振る舞いを変更させるダイナミックワイヤリング技術がホットスタンバイ型障害回復の実現に有用であることを示した．また，引数を更新して要求を再送信する手法を提案し，遠隔手続き呼び出しのあるコンポーネントの障害回復を可能にした．我々は移動ロボットモデルコンポーネントを用いて評価実験を行い，本アーキテクチャの有用性を示した．

キーワード:障害回復，レプリケーション， 遠隔手続呼出し，ロボットミドルウェア

Keywords: Fault-recovery, Replication, Remote Procedure call, Robot middleware

１.序論

RT ミドルウェアや ROS といったロボットミドルウェアによるコンポーネント指向開発が開発コストを削減すると期待されているが，障害耐性のあるシステムを開発するにはサポートが不十分である.障害耐性とは障害が発生しても正常にサービスを提供し続ける能力を指し,サービスロボットといった人間の近くで作業を行うシステムを開発する際には重要な課題となる.

ロボットミドルウェアでは,レプリケーションと呼ばれる手法を用いたソフトウェア障害回復フレームワークが研究されている.レプリケーションは障害耐性を持たせたいプロセスに対してバックアップを作り,障害発生時に切り替えることで障害から回復する手法である.既存の研究では「ホットスタンバイ」[1]と呼ばれる高速な障害回復が提案されているが開発コストが増加する問題があった．また，非同期通信であるデータフローのみを対象としており,同期通信である遠隔手続き呼び出しに関しては考慮されていなかった.

これらの問題を解決するため，我々はロボットミドルウェアのダイナミックワイヤリング機能に注目したホットスタンバイアーキテクチャと，遠隔手続き呼び出しに対応するための要求再送信手法を提案した.

２.関連研究

２.１　ホットスタンバイアーキテクチャ

ホットスタンバイ型のレプリケーションでは実行時での通信の再構成が必要となる.ロボットミドルウェアにおける従来手法では,クライアントコンポーネントとの接続ごとにプロキシを追加実装することを提案している[2]．しかしながら,クライアントの開発者がシステム開発者と異なる場合,このプロキシの開発は困難なものとなる.

２．２　分散システムにおける要求の再送信

遠隔手続き呼び出しの実行中に障害が発生しバックアップに切り替わった場合,要求側コンポーネントは再度要求を送信する必要がある.この要求再送信問題に関して,データベースの分野ではトランザクションキューを用いる手法がある[3]．しかしながら，ロボットシステムはデータベースと異なり，物理的な状態を持つ．そのため障害前と同じ要求を送信した場合,意図しない振る舞いを引き起こす危険性がある.

３.提案アーキテクチャ

３.１ダイナミックワイヤリングによるホットス

タンバイの実現

我々はダイナミックワイヤリングを利用することでプロキシのいらないホットスタンバイを実現した．図 1 に提案アーキテクチャを示す．ダイナミックワイヤリングは一部のロボットミドルウェアで採用されている，通信の再構成によってシステムの振る舞いを変更する機能である[4]．これをバックアップコンポーネントの通信再構成に応用することで，データフローのみを持つコンポーネントであれば，クライアント側の追加実装のないホットスタンバイが可能となった．

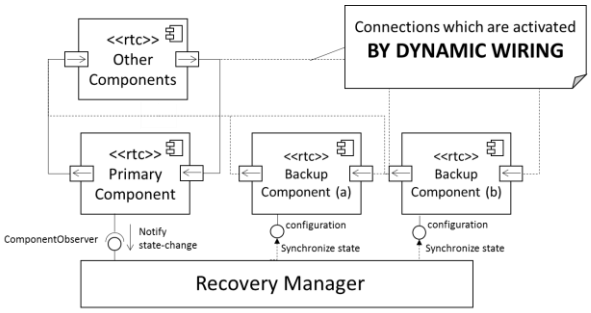

図 1 ロボットミドルウェアのための
ホットスタンバイアーキテクチャ

３．２ 引数更新型要求再送信機能の拡張

我々は遠隔手続呼出しを持つコンポーネントのレプリケーションに対応するためのアーキテクチャを提案した．クライアント側コンポーネントはこのアーキテクチャに従って１．要求の実行結果の監視，２．引数の更新，３．要求の再送信の三つの機能を実装することで障害回復直後のロボットの状態を反映させた要求の再送信を行うことが可能となった．

４．要求再送信機能の評価実験

引数を更新する要求再送信手法の有用性を示すため，移動ロボットシミュレータを用いた評価実験を行った．目標軌道を遠隔手続呼出しで受け取ると，移動ロボットに軌道を追従する速度命令を送信する制御コンポーネントを障害回復の対象とした．ロボットが軌道上のカーブ手前を通過した際に障害を発生させ，制御コンポーネントをバックアップに切り替える．その際に，ロボットの位置に合わせて軌道を更新した場合としなかった場合における，目標軌道と実際の軌道の誤差を測定した．障害から回復するまでの時間を 1 秒と３秒に設定し，各パターンで１０００回試行を行った．結果を図 2 に示す．

引数を更新しなかった場合，更新した場合と比較して誤差が増加した．これは障害発生中にもロボットが走行し続け，目標軌道から離れた地点から軌道追従を再開したことによる．一方軌道を更新した場合，障害回復直後のロボットの地点から目標地点までの軌道が再探索されるため，誤差の増加が抑えられた．軌道を更新することによって，目標軌道から離れた領域の走行を抑えることが可能となった．

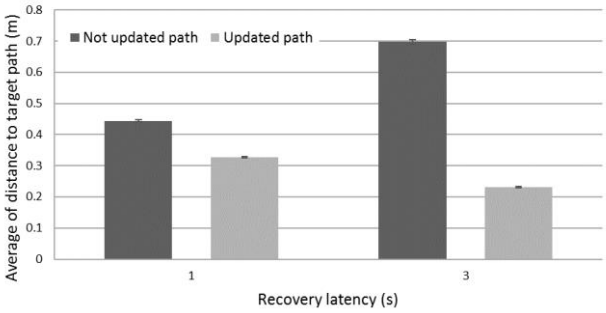

図 2 目標軌道との誤差の比較

５．結言

我々はロボットミドルウェアのための障害回復アーキテクチャを提案した．本研究における貢献は次の二点である．

1. ダイナミックワイヤリングを利用したホットスタンバイ型障害回復モデルを提案し，開発コストを削減する方法を示した．

2. ロボットの遠隔手続き呼び出し通信にレプリケーションを適用する際に要求の更新が必要であることを示し，障害回復が適応可能なロボットミドルウェアシステムを拡張した．

引数更新機能の実装はクライアント側に委ねられているため，この機能を自動生成するためのアーキテクチャの開発が今後の課題である．

参考文献：

[1] Fedoruk, Alan, and Ralph Deters. "Improving fault-tolerance by replicating agents." Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 2. ACM, 2002.

[2] Lauer, Michaël, et al. "Towards Adaptive Fault Tolerance: From a Component-Based Approach to ROS." CARS 2015-Critical Automotive applications: Robustness & Safety. 2015.

[3]McGee, William C. "The information management system IMS/VS, Part V: Transaction processing facilities." IBM systems journal 16.2 (1977): 148-168.

[4] Elkady, Ayssam, and Tarek Sobh. "Robotics middleware: A comprehensive literature survey and attribute-based bibliography." Journal of Robotics 2012 (2012).