

AltTF: Alternative lock-free TF tree

荻原湧志

December 17, 2021

概要

TF は ROS でよく使われるパッケージであり、これは有向木で座標変換を管理する。この木構造には Giant lock によるパフォーマンス低下の問題、線形補間による一貫性の欠落の問題があった。そこで、我々はトランザクション技術における 2PL を導入することにより、この問題を解決した。

1 序論

ROS はロボットソフトウェア用のミドルウェアソフトウェアプラットフォームであり、近年多くの研究用ロボットで使われている。

グローバル座標系の中でのロボットの位置、ロボットのローカル座標系の中でのセンサーの位置、センサーのローカル座標系の中での物体の位置、という風に別々に管理している。

この時、グローバル座標系での物体の位置を計算するにはグローバル座標系からロボットの位置の計算 * センサーの位置 * 物体の位置を計算すれば計算できる。

TF ではこのような座標系同士の変換を次の図のように有向森構造で管理できる。座標系の原点を frame と呼び、このフレームを node、座標系同士の位置関係を並行移動成分・回転成分で edge で表した木構造で管理する木構造で管理することにより、先程のようなグローバル座標系での物体 A の位置は、二つの frame 間のパスを辿ることで計算できるロボットの座標変換の情報は全てこの TF モジュールで管理される。

TF モジュールは ROS の多くのパッケージで使われているが、これには以下のような問題点がある。

Giant lock により、木構造が大きくなり多くのスレッドがアクセスするケースでは遅くなる。

(setTransform の non atomicity は実はまだ問題にはならない)

提供されているインターフェイスでは「最新」のデータを取るというメソッドでも実際には該当するパスのデータが完全に準備された状態の時間のデータを返している。

また暗黙的に線形補完が行われるため、データの一貫性がなくなる可能性がある。

1.1 貢献

そこで、データベースのトランザクション機構における並行制御のプロトコルの一種である 2PL を導入した。

その結果、224 コアのマシンではこの程度の性能差が出た。

また、データ一貫性を保証できるような setTransforms, lookupLatestTransform というインターフェイスを導入した。

1.2 論文構成

本論文の構成は次の通りである。2 章では既存の TF 森の構造とその問題点について述べる。3 章では提案手法である TF 森の再粒度ロックの導入とデータ一貫性のためのインターフェイスの提供について述べる。4 章では提案手法の評価結果を述べる。5 章では本研究の結論を述べる。6 章では今後の課題について述べる。

2 既存の TF 森の構造とその問題点

既存の TF の構造と問題点について

3 提案手法

提案手法と snapshot latest の概念の導入について時系列データにはいかなる変更も加えられないため serializability を導入する必要はないまた、最新のデータを見るには serializability が必要

4 評価

実験を行う

5 結論

データの正確性は必須

6 今後の課題

ここでは取り上げなかった TF 木の問題点として、部分的に

References

- [1] Simon J. Julier and Jeffrey K. Uhlmann "Building a million beacon map", Proc. SPIE 4571, Sensor Fusion and Decentralized Control in Robotic Systems IV, (4 October 2001)