

P2P 型データ統合における保守的な 2 相ロックに基づく分散トランザクション制御

三宅 康太[†] 湧田 悠佑[‡] 佐々木 勇和[§] 肖 川[¶] 鬼塚 真^{||}
大阪大学[†] 大阪大学[‡] 大阪大学[§] 大阪大学[¶] 大阪大学^{||}

1 はじめに

データ統合は複数のデータベースに分散するデータを統合し、利活用を容易にする技術である。従来のデータ統合は、分散するデータベースからスキーマの異なるデータを集め、単一のグローバルスキーマを用いて統合ビューを作成する中央集権型のアーキテクチャで実現される [5]。一方で従来の手法とは対照的に、近年 P2P 型のデータ統合アーキテクチャが提案されている。このアーキテクチャでは、ピア同士が相互にデータを共有することによってデータ統合を行う。また、ピアのデータベースに存在するレコードの更新が他のピアのデータベースへ伝播する場合がある。

P2P 型データ統合アーキテクチャとして Dejima [1, 2] が提案されている。Dejima は同じく P2P 型アーキテクチャを採用する ORCHESTRA [3, 4] に影響を受けたアーキテクチャである。Dejima と ORCHESTRA の最も重要な違いは、Dejima では様々な応用を可能とするため、分散トランザクション制御によってデータの大域的な一貫性を保証する点である。しかし P2P 型データ統合アーキテクチャに特化した分散トランザクション制御手法は現在提案されていない。

そこで本論文では Dejima を対象として、効率的な分散トランザクション制御手法を提案す

る。提案手法は *Family Record Set* という概念を導入することで、分散するデータを単一の仮想ビューと見なし、この仮想ビュー上で保守的な 2 相ロックを実行する。

2 Dejima

Dejima は P2P 型データ統合アーキテクチャであり、ピアと Dejima グループの 2 つの構成要素からなる。ピアは企業やサービスプロバイダなど、Dejima に参加するクライアントを表す。各ピアは独自のスキーマを持つテーブルであるベーステーブルと、ベーステーブルから導出されるビューである Dejima テーブルを保持している。Dejima テーブルは双方向変換を利用した更新可能なビューである [6]。ピアはデータを共有したい他のピアと Dejima グループを形成し、同じ Dejima グループに属するピア間で Dejima テーブルを同期する。

Dejima において、更新は次のような手順で伝播する。

1. ピアにおいてベーステーブルが更新される
2. そのピアの保持する Dejima テーブルに更新が反映される
3. Dejima テーブルは更新可能なビューであるため、この Dejima テーブルを同期して保持している Dejima グループ内のピアのベーステーブルに更新が伝播する
4. 更新が伝播しなくなるまで 2~3 を繰り返す

Dejima ではこのような手順で更新が伝播し、データ統合が行われる。なお、本論文では Dejima テーブルは SPJU クエリで定義されることを想定している。

Distributed Transaction Management Using Conservative 2-Phase Locking in P2P-based Data Integration

[†] Kota Miyake, Osaka University

[‡] Yusuke Wakuta, Osaka University

[§] Yuya Sasaki, Osaka University

[¶] Chuan Xiao, Osaka University

^{||} Makoto Onizuka, Osaka University

3 提案手法

提案手法の説明にあたり、P2P 型アーキテクチャの特徴についてまとめる。まず、各ピアが使用しているデータベースの種類が異なるため、各データベースで使用されている同時実行制御も異なる。したがって、これらの違いに影響を受けず大域的なデータの一貫性を保証する手法でなければならない。また P2P 型アーキテクチャでは通常の分散データベースに比べ、より頻繁にデッドロックが発生する。これは更新が伝播するという特性上、最初のロック取得から最後のロック取得までに時間がかかり、デッドロックの発生確率が上がるためである。

そこで、*Family Record Set* (FRS) という新たな概念を導入することで、分散するデータを単一の仮想ビューと見なし、この仮想ビュー上で保守的な 2 相ロックを実行する手法を提案する。提案手法は各ピアのデータベースにレコードのロック機能があれば実行できるため、各ピアは様々な種類のデータベースを採用できる。また、トランザクション実行前にロックを取得するため、デッドロックによる進行中のトランザクションのアボートが発生せず、特に競合の多いワークロードにおいて効率的に動作する。

Dejima においてレコードは、ユーザによって直接挿入される Original Record と、更新伝播によって生成される Derived Record に大別される。FRS は、ある Original Record とその Original Record の更新伝播によって生成された Derived Record の集合である。Dejima テーブルは SPJU クエリで定義されるため、ある FRS に属するレコードが更新される場合、その更新は同じ FRS に属するレコードに伝播するが、一方で異なる FRS に属するレコードには伝播し得ない。すなわち、FRS は更新伝播における単位である。したがって、同じ FRS に属するレコードを一つのレコードと見なしたとき、分散する複数のベーステーブルを単一の仮想ビューとみなせる。提案手法は各レコードにそのレコードの属する FRS を示す識別子を与え、これを用いてトランザクション実行前に

各ピアにロックを要求することで、保守的な 2 相ロックを実行する。

4 結論

本論文では P2P 型データ統合アーキテクチャである Dejima において、特に競合が多い場合に効率的な分散トランザクション制御手法を提案した。この手法は FRS の概念を導入することで分散するベーステーブルを単一の仮想ビューと見なし、このビュー上で保守的な 2 相ロックを実行する。保守的な 2 相ロックに基づく手法であるため、実行中のトランザクションがデッドロックによって妨げられず効率的である。今後の課題としては、提案手法の有効性を評価実験によって明らかにすることが挙げられる。

参考文献

- [1] Y. Asano et al. Making view update strategies programmable-toward controlling and sharing distributed data. *arXiv preprint arXiv:1809.10357*, 2018.
- [2] Y. Asano et al. Flexible framework for data integration and update propagation: system aspect. In *2019 IEEE BigComp*, pages 1–5. IEEE, 2019.
- [3] Z. G. Ives, N. Khandelwal, A. Kapur, and M. Cakir. Orchestra: Rapid, collaborative sharing of dynamic data. In *CIDR*, pages 107–118, 2005.
- [4] G. Karvounarakis et al. Collaborative data sharing via update exchange and provenance. *ACM TODS*, 38(3):1–42, 2013.
- [5] M. Lenzerini. Data integration: A theoretical perspective. In *ACM PODS*, pages 233–246, 2002.
- [6] V. Tran, H. Kato, and Z. Hu. Programmable view update strategies on relations. *Proc. VLDB Endow.*, 13(5):726–739, 2020.