

経営情報学会秋季全国発表大会の発表を終えて

西田 紗知

Sachi NISHIDA

情報メディア学専攻修士課程 2015 年度修了

1. はじめに

私は、11月28日に沖縄コンベンションセンターで開催された経営情報学会秋季全国発表大会に参加し、「クラウドにおけるデータ整合性」という題目でポスター発表を行った。

本稿では、発表した研究内容と発表を通じて得られたものについて述べていく。

2. 研究内容

2.1 概要

クラウドコンピューティングは、単なるデータの保存だけでなく、企業情報システムの処理基盤としても重要なものとなりつつある。しかし、この環境でトランザクション処理を行う場合、BASE というクラウド特有のデータ整合性概念を考慮する必要がある。

本研究は、モデリングとシミュレーションにより、システム設計段階でBASEによるデータ整合性への影響を検証・評価し、クラウドでのトランザクション処理の信頼性を向上させることを目指している。モデリングおよびシミュレーションには時間オートマトンに基づく検証ツールのUPPAALを用い、時相論理式の一種であるTCTLによりデータ整合性の検証・評価を行う。

2.2 クラウドにおけるトランザクション処理とデータ整合性

クラウド環境でのトランザクション処理では、システムへの負担の少ない、緩和されたデータ整合性原則である“BASE”というものが提案されている。BASEでのトランザクション動作は以下の性質を持つ。

1. Basically available 特性により、すべてのデータ資源は排他制御なしに利用可能。
2. Soft state により、透過性のない、すなわち異なる内容のレプリケーションを参照・更新する可能性。
3. Eventual consistency により、各瞬間ではなく、ある時間間隔内で整合性が保証される。

このような特性を持つトランザクションが同時並行的に実行される状況で、データ整合性が維持できるかが、クラウド環境でのデータベーストランザクション処理稼働可否を決定する重要な要因となる。この整合性評価を適切に行うには、整合性とは何かを厳密に定義されている必要がある。

データベースにおいて、整合性とは、データベース内のレコードもしくはエンティティの値自体もしくはその存在についての制約であるため、対象となるデータベースのレコード集合を定義域とする述語論理式での記述が可能である。これは形式的には、以下のように定義される。

1. 定義域は、データベース全体、個々のレコード、そして個々の属性を含むため

$$D = \left(\bigcup_i D_i\right) \cup \left(\bigcup_{i,j} r_{ij}\right) \cup \left(\bigcup_{i,j,k} a_{ij}^{(k)}\right)$$

となる。ここで D_i は i 番目のデータベース、 r_{ij} はデータベース D_i の j 番目のエン트리あるいはレコード、 $a_{ij}^{(k)}$ は r_{ij} の k 番目の属性である。

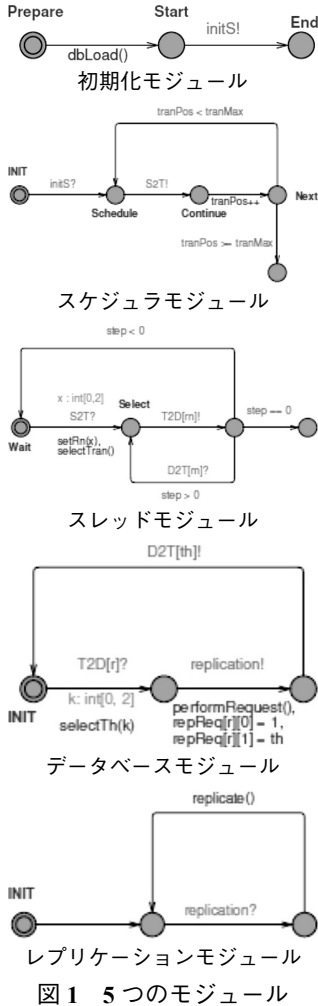
2. 整合性は、述語論理式の prenex 積標準形 (PCNF) で

$$Q_1 \cdots Q_n \left(\bigvee_j \bigwedge_i P_{ij}(t_1^{(ij)}, \dots, t_{m_{ij}}^{(ij)}) \right)$$

と表される。ここで、 Q_i は数量詞「 \forall 」と「 \exists 」をもつ変数である。

2.3 UPPAAL によるデータ整合性検証モデル

UPPAAL は、時間オートマトンを基礎とするシステムシミュレーションおよびモデル検査のためのツールであり、状態に相当するロケーション、状態遷移に相当するエッジ、およびそれらに関連付けられる時間制約や状態変数の更新を行うための関数を



組み合わせてモデルを構築する。モデルはパラメタライズ可能なテンプレートとして定義し、これをインスタンス化することで、シミュレーションが可能となる。複数のテンプレートをチャンネルにより接続し、同期を取ることが可能である。また、分岐時間時相論理 (Timed Computational Tree Logic {TCTL}) によるモデル検査を行うことができる。

本論文ではトランザクション処理の機能的側面に注目し、システムを図1に示す5つのモジュールに分割してモデル化した。

2.4 UPPAAL によるデータ整合性評価

前節のUPPAALモデルは、適切なデータベース用配列の定義と、トランザクション特性の登録によ

りシミュレーション可能となる。この結果は、2つの方法で評価することができる。

1つは、シミュレーション結果として出力された変数値を調べ、データベースが最終的に整合性制約を満たす値になっているかを調べる方法である。この場合、出力結果を一度外部ファイルに書き出し、Javaなどのプログラムで、第2.2節で述べた制約論理式を結果にあてはめ、整合性の評価を行う。この方法の利点は、シミュレーション自体は短時間で終了するため、効率的に整合性評価が行えることである。欠点は、整合性制約の変更や追加ごとに評価プログラムの書き換えが必要となる点、およびシミュレーションはトランザクション動作の1インスタンスを示すだけなので、繰り返しシミュレーションを行う必要がある点である。

もう1つの方法は、UPPAALのベリファイア機能を使う方法である。事前に制約条件を分岐時間時相論理式 (TLTL) 式として登録し、モデルがこれを満たすかどうかを検査する。ただし、本研究では時間制約を考慮しないため、TCTL式ではなくCTL式を使うことになる。この方法の利点は、トランザクション動作の全てのインスタンスがUPPAALによって調べられ、抜けのない整合性評価が行える点、および整合性評価のユーザープログラムを必要としない点である。欠点は、全状態空間探索のための計算量が膨大なものとなり、大規模なモデルには適用が難しい点である。

3. おわりに

これまで5度の口頭発表を行った成果が、落ちて発表することができた。だが、ポスター発表は今回が初めてだったので、口頭発表との違いに戸惑う場面もあった。特にポスターでは、口頭発表と比べ、視覚資料に盛り込める内容が少なく、補足的に口頭で説明を行うだけでは、説明しにくいと感じた。また、今回質問や意見で、鋭い指摘をいただき、非常に参考になった。これらを修士論文および公聴会で生かしたいと考えている。