

電子情報通信学会研究会サービス コンピューティング研究会 (SC) 研究会の発表を終えて

西田 紗知
Sachi NISHIDA

情報メディア学専攻修士課程 2年

1. はじめに

私は、3月27日に会津大学で開催された電子情報通信学会サービスコンピューティング研究会(SC)研究会に参加し、「VDM++によるBASEトランザクションのデータ整合性評価」という題目で口頭発表を行った。

2. 研究内容

クラウドコンピューティング環境、特にパブリッククラウドでのトランザクション処理におけるデータ整合性維持は完全な並行処理を行うBASEという原則に基づいている。これは、システムの可用性と拡張性とのトレードオフにより整合性の条件を緩和した結果と考えられる。このことにより、データベースアクセスの順序関係が従来に比べ複雑化し、トランザクションの振舞いの予測が困難となる。また、クラウドではデータベースの分散が隠蔽され、さらに非同期にレプリケーションされるため、一時点で同一データベースレコードに複数の観測値が存在し、上記の振舞いはさらに複雑なものとなる。このため、クラウド環境におけるトランザクション処理でのデータ整合性は、ACIDに比べはるかに煩雑な問題となる。本論文では形式仕様記述によりこの問題を厳密に記述・分析する手法を提案する。

3. クラウドにおけるデータ整合性

クラウドコンピューティングは可用性・拡張性の向上のため並行処理と多重化に重点が置かれ、トランザクション処理におけるデータ整合性に関するいくつかの制約が緩和されている。整合性という用語は、様々な文脈において異なった意味で使われてい

るため、まずこの概念を明確に定義する必要がある。アプリケーションレベルでは、データの意味に基づく整合性を指す。これは、データの値に対する制約として、述語論理式により表現することができる。述語論理式を構成するには、対象となる個体集合である領域 D 、述語および関数記号の集合、定数および変数の集合、および記号の具体的実装を示す解釈 I を定める必要がある。制約の対象は、個々の属性値、レコード、もしくはデータベース全体のいずれでもよい。領域 D は

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

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 番目のレコード、そして $a_{ij}^{(k)}$ はレコード r_{ij} の k 番目の属性値を表す。述語および関数に関しては、データベース操作の基本となる共通なもの、アプリケーション領域固有のものが考えられる。前者の代表的なものは“exists”, “between”, “count” など SQL で定義されているものであり、後者は所要量展開、安全在庫量、原価計算など特定の業務のみで使われるものである。前者は再利用可能なものとしてライブラリー化できるが、後者は個別に再定義する必要がある。以上のものが、すべて定義されると、整合性のための制約式は冠頭標準形(PCNF)の述語論理式で表すことができる。

$$Q_1 \dots Q_n \left(\bigvee_{j,i} P_{ij} (t_1^{(ij)} \dots t_{m_{ij}}^{(ij)}) \right)$$

ここで、 Q_i は、限量子“ \forall ”または“ \exists ”付の変数、すなわち、 \forall_{x_i} もしくは \exists_{x_i} 、 P_{ij} は述語、 $t_k^{(ij)}$ は関数および変数・定数より構成される項である。

4. VDM++によるBASEトランザクションの仕様記述

VDMはシステムの構造や機能に関する仕様や設計を集合論と述語論理に基づき、厳密かつ抽象的にモデル化、記述し、分析・検証を行う手法である。従来使われていたVDM-SLをオブジェクト指向に基づいたモデル化を扱えるように拡張されたものである。VDM++はC++やJAVAのようなオブジェ

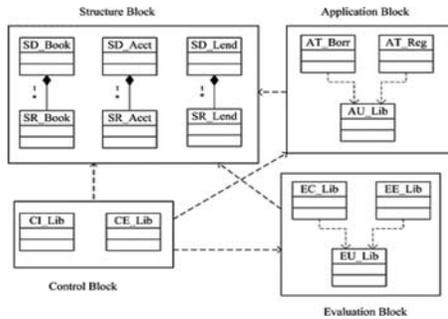


図1 VDM++仕様記述クラス構成

クト指向プログラミング言語に似ているが、より抽象的な構文を用いて、変数や操作といったシステムの構造および機能を記述できる。クラウドにおけるトランザクション処理でのデータ整合性を検証・評価するには、データベース構造とトランザクションの振舞いを正確に記述する必要がある。データ整合性評価のためには、まず、データベース構造、トランザクションの振舞い、データ整合性制約の述語論理式とその真偽判定、などをクラスにマップする必要がある。クラス定義に際し、可読性と再利用性を考慮して、クラスを

(Structure Block) データベース構造を記述する構造ブロック

(Application Block) アプリケーションの処理内容を記述するアプリケーションブロック

(Control Block) モデルの初期化とトランザクションの実行順序を行う制御ブロック

(Evaluation Block) データ整合性制約の定義と、処理結果の整合性検証を行う評価ブロック

の4つのブロックに分類する。

これらのブロックはJavaなどのパッケージに相当するが、VDM++には該当する概念がないため、便宜的にブロックにグループ化している。

5. VDM++によるデータ整合性評価

VDM++によるBASEトランザクションシステムの仕様記述が完了すれば、Overture Toolのような支援ツールにより、実行をシミュレートすることが可能となる。前節の手順では、データベースに含まれる属性値や、処理されるトランザクションの種類などが固定され、データ整合性評価のカバー範囲

が限定されたものとなる。より柔軟で効率的なシミュレーションを可能とするために、評価ブロックに、シミュレーションのシナリオ記述を行うためのクラス“Scenario s”を追加する。このクラスの主な役割は、データベース初期化に必要な情報の提供、実行すべきトランザクションの組合せの決定と生成、実行時の適切な並行度の維持などである。このために、以下のようなインスタンス変数とレコード型定義を含む。

(1) データベース id, 含まれるデータベースレコード数, key および属性値生成のパラメタ (分布形状など), レプリケーション数と同期の遅延, などを示す“DBGen”というレコード型。

(2) 上記“DBGen”のリスト。各要素が1データベースの生成に対応している。

(3) “select”, “update”などのデータベース操作, 対象のデータベース id, アクセスキー, 関連する属性値を含むレコード型“DBAccess”。これは単一のデータベースアクセスを表す。

(4) トランザクションを表すレコード型“Transaction”。これは、トランザクション id と上記“DBAccess”のリストの組として定義され、トランザクションからデータベースへのアクセスパターンを表現する。

(5) “Transaction”型のリスト。これは、シミュレーションに使用するトランザクションのセットを表す。

(6) 整数型のリスト。要素数が並行処理されるスレッド数, 各要素が実行中のトランザクション id を表す。“Scenario s”クラスの記述に基づくシミュレーションを実行中, 各トランザクションの commit/abort 処理で“EE s”クラスの整合性評価メソッドが呼び出され, “EC s”クラスの整合性制約式を表すメソッドを用いて, データ整合性の状況を確認する。ここで, 整合性制約が満たされない場合, その状況をレポートする。

6. 結論

本論文により、クラウド環境において、データ整合性概念の基本原則となる“BASE”に従うトランザクション処理を仕様記述言語VDM++によりモデル化しシミュレーションすることで、アプリケーションシステムのクラウド環境への適合性をデータ整合性の観点から評価する手法が示された。VDM++支援ツールによりシミュレーション可能であり、既存もしくは新規アプリケーションがクラウド環境でデータ整合性を維持できるかを評価することができる。