

# RAMP トランザクションの一般化における 検索一貫性の問題について

徐 海燕<sup>†</sup> 古川 哲也<sup>††</sup> 史 一華<sup>†††</sup>

<sup>†</sup> 福岡工業大学情報工学部情報工学科 〒 811-0295 福岡市東区和白東 3-30-1

<sup>††</sup> 九州大学大学院経済学研究院 〒 819-0395 福岡市西区元岡 744

<sup>†††</sup> 西南学院大学商学部 〒 814-8511 福岡市早良区西新 6-2-92

E-mail: <sup>†</sup>xu@fit.ac.jp, <sup>††</sup>furukawa@econ.kyushu-u.ac.jp, <sup>†††</sup>shi@seinan-gu.ac.jp

あらまし 爆発的に増えているデータを高速に処理するためには、トランザクション処理の高性能化が重要な課題となっている。その中、分散環境で直列可能性を緩和した Read Atomic (RA) 隔離性という新たな隔離性レベルを高速に実現している Read Atomic Multi-Partition (RAMP) トランザクションがある。コーディネーションフリーで検索されるデータの一貫性を実現する RAMP トランザクションの技術をより多くのアプリケーションに適用できるようにするためには、トランザクションモデルを read only と write only トランザクションから、read と write 操作の両方を含む一般化されたトランザクションモデルへの拡張が不可欠である。本論文では、一般化されたトランザクションモデルにおいて、read from 関係で関連するトランザクションの変更結果に対する検索が fractured read にならないための RF 隔離性という RA 隔離性から拡張される性質を導入し、判定グラフや検索一貫性の性質について検討する。キーワード マルチバージョン直列可能性 MVSR, RA 隔離性, RAMP トランザクション

## 1 はじめに

分散データベースにおいてスケラビリティを実現するために、直列可能性を緩和した Read Atomic 隔離性という新たな隔離性レベルが提案され、それを実現する高性能な処理方式として Read Atomic Multi-Partition(RAMP) トランザクションが提案されている [1]。RAMP トランザクションはマルチバージョンに基づく効率的なプロトコルにより、Read Atomic 隔離性を保証できる。先進的なデバイスなどの適用によって、一層の高速化に関する研究が活発に行われている [4]。コーディネーションフリーで検索されるデータの一貫性を実現する RAMP トランザクションの技術をより多くのアプリケーションに適用できるようにするためには、トランザクションモデルを read only と write only トランザクションから、read と write 操作の両方を含む一般化されたトランザクションモデルへの拡張が不可欠である。例えば、参照整合性制約などの一貫性制約を満たすために参照しないとされないデータ項目が存在する。

本論文では、一般化されたトランザクションモデルにおいて、read from 関係にある 2 つずつのトランザクションに対する fractured read 現象を防ぐための RF 隔離性を導入する。RAMP トランザクションは、Read only と Write only トランザクションからなる並行実行を処理対象としているが、一つの作業が最初にすべて決められるとは限らず、複数個のステップからなることがあり、他のトランザクションの変更結果を参照した上で操作を行う必要のある場合もある。生成された結果が参照整合性制約などの一貫性制約を満たすことを確認するためにも、他のトランザクションの変更結果を参照する必要があ

る。本論文では、read from 関係でトランザクション間の関係が定義される場合に、read from 関係にある複数個のトランザクションの変更結果に対する検索に対応するための RF 隔離性を導入する。さらに、RF 隔離性の判定方法を与え、検索一貫性を満たす性質を示し、マルチバージョン直列可能性や RA 隔離性との関係を明確にする。

本論文は次のように構成される。基本的事項についての説明を 2 章で行う。3 章では、RF 隔離性を定義し、性質分析などについては 4 章で行う。RF 隔離性を満たさせるアルゴリズムの概要を 5 章で述べる。6 章は全体のまとめである。

## 2 基本的事項

トランザクション  $T$  は、アイテム  $x$  のバージョン番号  $k$  のバージョン  $x_k$  に対する検索操作  $r(x_k)$  や変更操作  $w(x_k)$  からなる集合である。便宜上初期のデータベースにおいて、アイテム  $x, y, \dots$  の初期バージョン番号を 0 で表し、それ以外の場合は、それを変更するトランザクションの時刻印がバージョン番号となる。

検索操作、または変更操作しか含まないトランザクションを read only、または write only トランザクションという。トランザクション集合  $\mathcal{T} = \{T_1, T_2, \dots, T_n\}$  上のスケジュール  $H$  は、各  $T_i \in \mathcal{T}$  の和集合である。

[定義 1] [3] トランザクション集合  $\mathcal{T}$  上のスケジュール  $H$  のマルチバージョン直列可能性判定グラフ  $MVSR(H)$  は、 $T_i \in \mathcal{T}$  を節点とし、次の 2 種類の枝を有するグラフである。

read from( $rf$ ) 枝：

$w(x_j) \in T_j, r(x_k) \in T_k (j < k)$  に対して  $T_j \xrightarrow{rf} T_k$  である。

version order( $vo$ ) 枝 :

$w(x_i) \in T_i, w(x_j) \in T_j, r(x_j) \in T_k (i < j, i < k, j < k)$  に対して,  $i < j$  ならば,  $T_i \xrightarrow{vo} T_j$ , そうでなければ,  $T_k \xrightarrow{vo} T_i$  である .

$MVSR(H)$  が非巡回の場合,  $H$  は単一コピーマルチバージョン直列可能という [3]. 同じアイテムに対する変更操作は異なるバージョンを生成するため競合しないが, 同じアイテムに対する変更操作と検索操作間の競合関係は定められている . 本論文では,  $MVSR(H)$  に  $rf$  枝のあるトランザクション間に  $rf$  関係があるという .

[例 1] スケジュール  $H_1, H_2, H_3$  を

$H_1$ :  $T_1 \ w(x_1); w(y_1)$   
 $T_2 \ r(x_1); r(y_0)$   
 $H_2$ :  $T_1 \ w(x_1); w(y_1)$   
 $T_2 \ r(y_1); w(z_2)$   
 $T_3 \ r(x_0); r(z_2)$   
 $H_3$ :  $T_1 \ w(x_1); w(y_1); w(u_1)$   
 $T_2 \ r(x_1); w(z_2)$   
 $T_3 \ w(x_3); w(y_3)$   
 $T_4 \ r(y_3); r(z_2); r(u_1)$

とする . それらのスケジュールの判定グラフ  $MVSR(H_1), MVSR(H_2), MVSR(H_3)$  を図 1 に示している .  $MVSR(H_1)$  には,  $w(x_1) \in T_1, r(x_1) \in T_2$  より  $T_1$  から  $T_2$  への  $rf$  枝が,  $r(y_0) \in T_2, w(y_1) \in T_1$  より  $T_2$  から  $T_1$  への  $vo$  枝がある . 従って, 閉路が存在する .

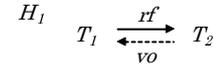
$MVSR(H_2)$  には, まず  $w(y_1) \in T_1, r(y_1) \in T_2$  による  $T_1$  から  $T_2$  への  $rf$  枝と,  $w(z_2) \in T_2, r(z_2) \in T_3$  による  $T_2$  から  $T_3$  への  $rf$  枝からなる  $T_1$  から  $T_3$  への経路がある . 次に, 経路の末尾の  $T_3$  から経路の先頭  $T_1$  へは,  $r(x_0) \in T_3, w(x_1) \in T_1$  による  $vo$  枝があるため, 閉路が存在する .

一方,  $MVSR(H_3)$  には,  $w(x_1) \in T_1, r(x_1) \in T_2$  より  $T_1$  から  $T_2$  へ,  $w(u_1) \in T_1, r(u_1) \in T_4$  より  $T_1$  から  $T_4$  へ,  $w(z_2) \in T_2, r(z_2) \in T_4$  より  $T_2$  から  $T_4$  へ,  $w(y_3) \in T_3, r(y_3) \in T_4$  より  $T_3$  から  $T_4$  への  $rf$  枝がある, 一方,  $w(x_1) \in T_1, w(x_3) \in T_3, r(x_1) \in T_2$  に対して,  $i(=3) < j(=1)$  でないため,  $T_2$  から  $T_3$  への  $vo$  枝,  $w(y_1) \in T_1, w(y_3) \in T_3, r(y_3) \in T_4$ , に対して,  $i(=1) < j(=3)$  であるため,  $T_1$  から  $T_3$  への  $vo$  枝がある . 時刻印順と逆順の枝が存在しないため,  $MVSR(H_3)$  には閉路が存在しない .

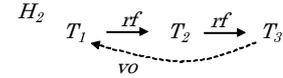
### 3 RF 隔離性

#### 3.1 RA 隔離性

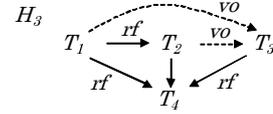
トランザクション  $T$  の検索操作と変更操作の実行結果からなる集合をそれぞれ  $T^R, T^W$  と記し, それぞれトランザクション  $T$  の検索結果と, トランザクション  $T$  の変更結果という . トランザクション  $T$  の変更結果を部分的にしか読み出していない場合, fractured read 現象が生じているという . 例えば,  $H_1$  において,  $T_1$  の変更結果  $T_1^W$  は  $\{w(x_1), w(y_1)\}$  である .  $T_2 = (r(x_1); r(y_0))$  は  $T_1$  の変更結果  $T_1^W = \{w(x_1), w(y_1)\}$  に



(a)



(b)



(c)

図 1  $MVSR(H_1), MVSR(H_2), MVSR(H_3)$

対して  $x_1$  しか読み出せていないため, fractured read 現象が生じている . 図 1(a) に示しているように  $MVSR(H_1)$  には  $rf$  と  $vo$  枝による閉路がある . 本論文では,  $MVSR(H)$  における  $rf$  と  $vo$  枝による 2 つのトランザクション間の閉路を RA 閉路と呼ぶ .

トランザクションの変更結果に対して, fractured read 現象が生じていないスケジュールは, RA 隔離性を満たすという [1][2]. Read Atomic 隔離性を満たせば, fractured read 現象を防ぐことができる .

トランザクションが read only, write only トランザクションモデルに限定し, 検索集合が Read Subset Items-Written (RISW) に制限された場合に, RA 隔離性を満たすスケジュール  $H$  は, マルチバージョン直列可能であり,  $MVSR(H)$  は非巡回である .

$H_1$  において,  $T_2$  が  $T_1$  の変更結果に対して fractured read 現象が生じているので,  $H_1$  は RA 隔離性を満たさない . 一方,  $H_2$  において, 2 つ以上の検索操作を含むトランザクションは  $T_3$  のみで,  $T_3$  は  $T_1$  の変更結果に対しても,  $T_2$  の変更結果に対しても fractured read 現象が生じていない . このため,  $H_2$  は RA 隔離性を満たす . 同様に  $H_3$  も RA 隔離性を満たす .

Read Atomic 隔離性を保証でき, 高いスループットとスケラビリティを実現できる新たな処理方式として RAMP トランザクションが提案された [1]. RAMP-fast アルゴリズムをはじめとする 3 つのアルゴリズムが提案されているが, ほとんどの場合で最も高速に処理を終了できる RAMP-fast アルゴリズムは,  $T^W$  によって変更されるバージョン集合を, メタデータ  $md$  として記憶し, 処理に活用している .

[例 2] スケジュール  $H_1$  に対して, RAMP-fast アルゴリズムの実行状況が次のようになる .

$H_1$ :  $T_1 \ w(x_1); w(y_1)$   
 $\langle x_1 : md \rangle = \{x_1, y_1\}$   
 $\langle y_1 : md \rangle = \{x_1, y_1\}$

$T_2$  が  $r(x_1); r(y_0)$  として実行される場合に,  $y_1 \in \langle x_1 : md \rangle$  によって,  $T_1$  の変更結果  $\{x_1, y_1\}$  に対する fractured read 現象が検出され,  $y$  が再検索され,  $T_2$  が  $r(x_1); r(y_1)$  として実行される .

### 3.2 RF 隔離性

トランザクションの変更操作は、write only トランザクションのように1つのトランザクションのみで結果を生成する場合と、 $H_2$  中の  $T_2$  や  $H_3$  中の  $T_2$  のように他のトランザクションの結果を参照した上で結果を生成する場合がある。変更結果が参照整合性制約などの一貫性制約を満たすことを確認するためには、トランザクションは他のトランザクションの変更結果を検索しなければならない。本論文では、次の仮定の下で議論する。

仮定：各トランザクション  $T_i$  は、 $T_i^W$  と一貫性上関連するデータは  $T_i^R$  で検索する。

以上の仮定の下では、複数個のトランザクションの変更結果をまとめて考える必要がある。そのための概念を導入する。

[定義 2]  $rf$  関係がある  $T_i$  と  $T_j (i < j)$  に対して、

$$T_i^W \cup T_j^W - \{w(x_k) \mid w(x_j) \in T_j, w(x_i) \in T_i, k = \min(i, j)\}$$

を、 $T_j$  の RF 変更結果  $T_{i,j}^W$  とする。

すなわち、2つのトランザクションの変更結果の和集合から、個々のアイテムのバージョンが唯一になるように、同じアイテムのバージョンが2つ存在する場合に、バージョン番号の大きい方、i.e.,  $j$  の方のみを残すようにしている。このため、 $T_{i,j}^W$  が  $T_j^W$  の超集合、i.e.,  $T_{i,j}^W \supseteq T_j^W$ 、である。

$H_2$  においては、 $T_1$  の変更結果  $T_1^W$  は  $\{w(x_1), w(y_1)\}$  であり、 $T_2$  の変更結果  $T_2^W$  は  $\{w(z_2)\}$  であり、 $T_2$  の RF 変更結果  $T_{1,2}^W$  は、 $\{w(x_1), w(y_1), w(z_2)\}$  となる。

個々のトランザクションの変更結果  $T_i^W$ 、または  $T_j^W$  は一貫性上関連する。 $T_j$  が  $T_i^W$  を参照している、すなわち、 $T_i$  と  $T_j (i < j)$  間に  $rf$  関係があるならば、仮定より、 $T_j^W$  と  $T_i^W$  は一貫性上関連する。 $T_{i,j}^W$  は  $T_j^W$  と  $T_i^W$  のまとめなので、次の結果が成り立つ。

[補題 1] トランザクションの変更結果  $T_j^W$  とその RF 変更結果  $T_{i,j}^W$  とは、一貫性上関連する。

このため、RF 変更結果に対して fractured read 現象が起きなければ、一貫した結果から検索していないことになる。

[定義 3]  $rf$  関係がある  $T_i$  と  $T_j (i < j)$  に対して、 $T_k^R$  が  $T_j^W$  の一部を検索し、RF 変更結果  $T_{i,j}^W$  中バージョンを検索していない、i.e.,  $r(x_j), r(y_p) \in T_k (p \leq j, w(y_q) \in T_{i,j}^W, p < q)$ 、ならば、 $T_k^R$  は  $T_{i,j}^W$  に対して fractured read 現象が起きているという。

$H_2$  において、 $T_3 = (r(x_0); r(z_2))$  は  $T_2^W$  から  $z_2$  を検索し、RF 変更結果  $T_{1,2}^W = \{w(x_1), w(y_1), w(z_2)\}$  から  $x_1$  を検索しておらず、 $T_{1,2}^W$  に対して fractured read 現象が生じている。図 1(b) に示しているように、 $MVSR(H_2)$  には、 $T_1$  から  $T_2$ 、 $T_2$  から  $T_3$  への  $rf$  枝と、 $T_3$  から  $T_1$  への  $vo$  枝がある。

本論文では、 $MVSR(H)$  における  $rf, rf, vo$  枝による閉路を RF 閉路と呼ぶ。RF 閉路は、3つのトランザクション間の  $rf, rf, vo$  関係による限定された閉路である。

トランザクションの検索操作は、1つのトランザクションの RF 変更結果に留まらないことがある。例えば、 $H_3$  における  $T_4$  の検索は、 $T_3$  の RF 変更結果と  $T_2$  の RF 変更結果に及ん

でいる。

$MVSR(H)$  グラフから、 $vo$  枝を除いたグラフを  $H$  の  $rf$  関係グラフという。明らかに、 $rf$  関係グラフは有向非巡回グラフである。有向非巡回グラフは、個々の経路における問題に分解できる。

$T$  によって検索されるバージョン集合が  $T_{i_1}$  を始点、 $T_{i_m}$  を終点とする  $rf$  経路に及んだ場合に、検索はそれぞれのアイテムにの最大値のバージョン番号のバージョンを生成したトランザクションに対する検索に細分される。問題もそれぞれのトランザクションの RF 変更結果に対して fractured read 現象が生じていない問題に細分される。

一方、それぞれのアイテムに対する変更中の最大値からなる集合  $\{w(x_{i_k}) \mid w(x_{i_j}) \in T_{i_j}, 1 \leq j \leq m, k = \max(i_j)\}$  を、経路変更結果  $P_{i_1, i_k}^W$  という。

$T$  によって検索されるバージョン集合が複数の経路に及んだ場合に、異なる経路変更結果に同一のアイテムが存在すれば、バージョン番号も一致しなければならない。言い換えれば、他の経路の変更結果を潜在的な検索結果として加えても、RF 変更結果に対して fractured read 現象が生じていないことを要求する。

これで RF 変更結果を対象とする RF 隔離性を導入できる。

[定義 4]  $T$  によって検索されるバージョン集合が RF 変更結果に対して fractured read 現象が生じてなければ、 $T$  は RF 隔離性を満たすという。

すべての  $T_i \in \mathcal{T}$  が RF 隔離性を満たす時、 $H$  が RF 隔離性を満たすという。

$H_3$  においては、 $T_4$  の検索は  $T_3$  の変更結果  $T_3^W = \{w(x_3), w(y_3)\}$  と  $T_2$  の RF 変更結果  $T_{1,2}^W = \{w(x_1), w(y_1), w(z_2)\}$  に及ぶ。両者の結果はアイテム  $x, y$  においてはバージョン番号が一致していないので、 $H_3$  は RF 隔離性を満たさない。

[例 3]  $H_3$  と  $T_3$  のみが異なる  $H_4$  においては、 $T_1$  と  $T_2$  間に、 $T_2$  と  $T_3$  間に  $rf$  関係があるので、 $T_1, T_2, T_3$  という  $rf$  経路が存在する。 $T_4$  の検索は  $T_3$  の RF 変更結果  $T_{1,3}^W = \{w(x_3), w(y_3), w(z_2)\}$  に対する検索  $\{r(y_3), r(z_2)\}$  と、 $T_2$  の RF 変更結果  $T_{1,2}^W = \{w(x_1), w(y_1), w(u_1), w(z_2)\}$  に対する検索  $\{r(z_2), r(u_1)\}$  に細分される。 $w$  どの RF 変更結果に対しても fractured read 現象が生じてないので、 $T_4$ 、すなわち、 $H_4$  は RF 隔離性を満たす。

$H_4$  と  $T_4$  のみが異なる  $H_5$  においては、 $T_4$  の検索はトランザクション  $T_3$  の変更結果  $T_3^W$  と、 $T_0$  の変更結果  $T_0^W$  に留まっており、いずれに対しても fractured read 現象が生じていない。 $T_4$ 、すなわち、 $H_5$  は RF 隔離性を満たす。しかし、図 2(b) に示されているように、 $MVSR(H_5)$  には  $T_1, T_2, T_3, T_4$  間の閉路が存在する。

$H_4$  と  $T_4$  のみが異なる  $H_6$  においては、 $T_4$  の検索は  $T_3$  の RF 変更結果  $T_{1,3}^W = \{w(x_3), w(y_3), w(z_2)\}$  に対する検索  $\{r(y_3), r(z_2)\}$  と、 $T_2$  の RF 変更結果  $T_{1,2}^W = \{w(x_1), w(y_1), w(u_1), w(z_2)\}$  に対する検索  $\{r(z_2), r(u_0)\}$  に細分される。前者は fractured read 現象が生じていないが、後者は fractured read 現象が生じている、このため、 $H_6$  は RF 隔

離性を満たさない．図 2(c) に示されているように  $MVSR(H_6)$  には  $T_1, T_2, T_4$  間に RF 閉路が存在する．

- $H_4$ :  $T_1$   $w(x_1); w(y_1); w(u_1)$   
 $T_2$   $r(x_1); w(z_2)$   
 $T_3$   $r(z_2); w(x_3); w(y_3)$   
 $T_4$   $r(y_3); r(z_2); r(u_1)$
- $H_5$   $T_1$   $w(x_1); w(y_1); w(u_1)$   
 $T_2$   $r(x_1); w(z_2)$   
 $T_3$   $r(z_2); w(x_3); w(y_3)$   
 $T_4$   $r(y_3); r(u_0)$
- $H_6$   $T_1$   $w(x_1); w(y_1); w(u_1)$   
 $T_2$   $r(x_1); w(z_2)$   
 $T_3$   $r(z_2); w(x_3); w(y_3)$   
 $T_4$   $r(y_3); r(z_2); r(u_0)$

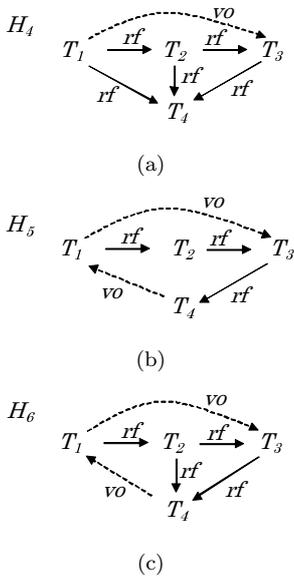


図 2  $MVSR(H_4), MVSR(H_5), MVSR(H_6)$

明らかに，RF 隔離性を満たせば，RA 隔離性も満たす．RA 隔離性を満たさないなら，RF 隔離性も満たさない．具体的に， $H_1$  は RA 隔離性も RF 隔離性を満たさず， $H_2, H_3$  と  $H_6$  は RA 隔離性は満たすが，RF 隔離性は満たさない． $H_4$  と  $H_5$  は RA 隔離性も RF 隔離性も満たす．

RA 隔離性は 1 つのトランザクションの変更に対する検索に関するものであり，RF 隔離性は rf 関係のある親子トランザクションの変更結果に対する検索まで拡張している．検索集合は，個々のトランザクションの変更結果と，rf 関係のあるトランザクションの親子変更結果に対する検索に細分できるので，RA 隔離性の場合の RISW 制限は不必要になる．

## 4 RF 隔離性の性質

本章では，RF 隔離性を満たす  $H$  の判定グラフ  $MVSR(H)$  の性質と RF 隔離性を満たすことの意義について議論する．

### 4.1 検索一貫性との関係

本論文の仮定の下では，補題 1 で示しているように，トラン

ザクションの変更結果  $T_j^W$  とその RF 変更結果  $T_{i,j}^W$  とは，一貫性上関連する．バージョン間の一貫性情報が利用できれば，各トランザクションによって検索されるバージョン集合間の検索一貫性は，次のように定義できる．

[定義 5]  $T^R$  中に一貫性上で関連するバージョンに対して fractured read 現象が生じなければ， $T^R$  は検索一貫性を満たすという．

トランザクション集合  $\mathcal{T}$  上のスケジュール  $H$  におけるすべての  $T_i \in \mathcal{T}$  が検索一貫性を満たす時， $H$  は検索一貫性を満たすという．

直列可能性は，意味論情報がない場合における検索一貫性を保証する基準である．補題 1 でまとめたバージョン間の一貫性上関連する情報を，検索一貫性の判定に利用できる．

[定理 1] 検索操作が本論文の仮定を満たす場合では，スケジュール  $H$  が検索一貫性を満たすための必要十分条件は， $H$  が RF 隔離性を満たすことである．

必要性：検索一貫性を満たさないとすると，補題 1 より RF 変更結果に対する fractured read 現象が生じることになる．したがって， $H$  が RF 隔離性を満たさないことになる．矛盾である．

十分性： $H$  が RF 隔離性を満たさないとすると，それは潜在的検索を含まない場合と含む場合に分けられる．前者は RF 変更結果に対する fractured read 現象が生じていることを意味し，補題 1 より検索一貫性を満たさない．後者は検索している異なる経路における検索に一致するバージョン集合が存在しない，言い換えれば，検索一貫性を満たす検索結果が存在しないことを意味する．いずれの場合も矛盾である．

### 4.2 判定グラフ $MVSR(H)$ との関係

RF 隔離性と判定グラフ  $MVSR(H)$  は，表 1 に示しているように，関係がないように見える．RF 隔離性を満たさない  $H_3$  の  $MVSR(H_3)$  は非巡回であり，RF 隔離性を満たす  $H_5$  の  $MVSR(H_5)$  は巡回である．これは，両者の基づく前提の違いによるものである．

表 1 各スケジュールの性質

	RA 隔離性	RF 隔離性	$MVSR(H)$	
			非巡回	非巡回
$H_1$	×	×	×	×
$H_2$	○	×	×	×
$H_3$	○	×	○	×
$H_4$	○	○	○	○
$H_5$	○	○	×	×
$H_6$	○	×	×	×

$MVSR(H)$  が非巡回であることは， $H$  が単一コピーマルチバージョン直列可能であるための必要十分条件である [3]．直列可能性は，個々のトランザクションが一貫したデータベースから実行していれば，その実行されたデータベースも一貫であるという前提に基づいている．本論文ではその前提を設けておらず，各トランザクションが検索一貫性を満たせば，その RF 変

更結果が一貫しているという前提に基づいている。

一方、表 1 中の  $H'$  は、潜在的な検索も含めた場合のスケジュールを意味する。例えば、 $H_3$  において、潜在的検索を含めると、 $T_3$  は  $r(y_3); r(x_3); r(z_2); r(u_1); r(x_1); r(y_1)$  になり、 $MVSR(H'_3)$  には図 3 に示しているように RA 閉路が存在することになる。

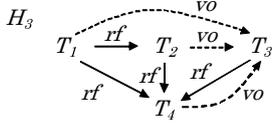


図 3  $MVSR(H'_3)$

逆に、 $H_5$  の場合、 $MVSR(H_5)$  に閉路が存在しても、どのトランザクションの RF 変更結果に対する fractured read 現象が生じてないので、RF 隔離性を満たすことになる。

ただし、 $H$  が RF 隔離性を満たすならば、トランザクションの変更変更に対しても RF 変更結果に対しても fractured read 現象が生じない。前者は、RA 閉路として、後者は RF 閉路として表されるため、次の結果が成立つ。

[定理 2] スケジュール  $H$  が RF 隔離性を満たすための必要十分条件は、潜在的な検索も含めたスケジュール  $H'$  の  $MVSR(H')$  に RA 閉路も RF 閉路も存在しないことである。

必要性：RF 隔離性を満たさないなら、RF 変更結果に対して、fractured read 現象が起きているか、異なる rf 経路に対する検索結果が一意にまとまらないかのいずれかになる。

前者の場合は、 $T$  の変更結果を検索しており、RF の変更結果と異なったバージョンを検索していることになる。検索すべきバージョンより古いバージョンを検索していれば、RF 閉路が生じることになる。検索すべきバージョンより新しいバージョンを検索していれば、潜在的検索による RA 閉路が生じることになる。

一方、後者の場合は潜在的検索による RA 閉路が生じることになる。矛盾である。

十分性：潜在的検索によらない RF 閉路があるとすると、RF 変更結果に対して、fractured read 現象が起きているとなり、RF 隔離性を満たさない。一方、潜在的検索による閉路があるとすると、潜在的検索が RF 変更結果に対して、fractured read 現象が起きていることになる。同じく RF 隔離性を満たさない。矛盾である。

### 4.3 スケジュールの正当性との関係

まず、RF 隔離性と単一コピーマルチバージョン直列可能性 MVSR の関係について検討する。

$MVSR(H)$  中の閉路を分類するために、各  $T_i \in \mathcal{T}$  を、 $T_i$  と  $T_i^R$  にしたスケジュール  $H'$  について検討する。 $H$  に各  $T_i \in \mathcal{T}$  の検索操作からなる部分トランザクション  $T_i^R$  を加えただけなので、 $MVSR(H)$  に閉路が存在するならば、 $MVSR(H')$  にも閉路が存在し、その逆も成り立つ [6]。 $MVSR(H')$  中の閉路

は、部分トランザクション  $T_i^R$  を含む場合と含まない場合に分けられる。

スケジュール  $H_1 \sim H_3, H_5 \sim H_6$  は前者の例である。後者の例を次に示す。

[例 4] スケジュール  $H_7$  の  $MVSR(H_7)$  は図 4 に示しているように  $T_2$  と  $T_3$  間に閉路が存在するが、 $T_2$  も  $T_3$  も 1 つの検索操作しか含んでいないので、 $T_i^R$  を含まない閉路である。

$H_4$ :  $T_1$   $w(x_1); w(y_1)$   
 $T_2$   $r(x_1); w(x_2)$   
 $T_3$   $r(y_1); w(y_3)$

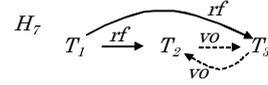


図 4  $MVSR(H_7)$

前者の閉路はトランザクションの検索されるデータに問題があることを意味する。本論文では、定理 2 に示しているように検索一貫性に関わる閉路を除外している。

後者の閉路については、本論文では、それが存在しても各トランザクションの実行が検索一貫性を満たすように保証する立場である。また、検索されるバージョン集合に対しては、Read Subset Items-Written (RSIW)[2] のような制限を設けていない。

次に、RF 隔離性と RA 隔離性の関係について議論する。

RA 隔離性は、トランザクションが read only, write only トランザクションモデルに限定し、検索されるバージョン集合が RISW に制限された場合に、 $MVSR(H)$  が非巡回、すなわち単一コピーマルチバージョン直列可能であることを保証している。代わりに、本論文で提案している RF 隔離性は、トランザクションモデルに対しても、検索集合にも制限を設けない場合に、検索一貫性を保証している。各トランザクションが検索一貫性を満たせば、一貫する変更結果と、一貫する RF 変更結果を生成できる。

## 5 RF 隔離性アルゴリズム

まず、 $T^W$  によって変更されるバージョン集合を  $md$  に、 $T^R$  によって検索されるバージョンの  $md$  を  $rf$  に、メタデータとして、記憶する。

RF 隔離性アルゴリズムを付録に添付している。主な関数は次の 2 つである。

$check(v)$  関数:  $v$  を終点とする rf 経路を探索し、RF 隔離性に影響しうるバージョンの集合を返す。

$merge()$  関数: 異なる rf 経路上のバージョン集合間が合併可能かどうかをチェックし、可否を返す。

[例 5] スケジュール  $H_3 \sim H_5$  についての実行状況を次に示す。

$H_3$   $T_1$   $w(x_1); w(y_1); w(u_1)$   
 $\langle x_1 : md \rangle = \{x_1, y_1, u_1\}, \langle x_1 : rf \rangle = \{\}$   
 $\langle y_1 : md \rangle = \{x_1, y_1, u_1\}, \langle y_1 : rf \rangle = \{\}$   
 $\langle u_1 : md \rangle = \{x_1, y_1, u_1\}, \langle u_1 : rf \rangle = \{\}$   
 $T_2$   $r(x_1); w(z_2)$   
 $\langle z_2 : md \rangle = \{z_2\}, \langle z_2 : rf \rangle = \{x_1, y_1, u_1\}$   
 $T_3$   $w(x_3); w(y_3)$   
 $\langle x_3 : md \rangle = \{x_3, y_3\}, \langle x_3 : rf \rangle = \{\}$   
 $\langle y_3 : md \rangle = \{x_3, y_3\}, \langle y_3 : rf \rangle = \{\}$   
 $T_4$   $r(y_3); r(z_2); r(u_1)$  が実行される時,  
 $T_{3,2}^W = \{x_3, y_3\}$   
 $T_{2,1}^W = \{x_1, y_1, u_1, z_2\}$

それ以上の  $rf$  経路が存在しないので,  $T_4$  が  $r(y_3)$  を行えば, その検索は  $\{x_3, y_3\}$  の部分集合でなければならないため,  $r(y_3)$  に調整される.  $T_4$  が  $r(z_2)$  を行えばその検索は  $\{x_1, y_1, u_1, z_2\}$  の部分集合でなければならないため,  $r(z_2); r(u_1)$  に調整される. いずれかを選択することになる.

一方,  $H_4$  においては,  $T_2$  までは  $H_3$  と同じであるが,  $T_3$  から差が出る.

$T_3$ :  $r(z_2); w(x_3); w(y_3)$   
 $\langle x_3 : md \rangle = \{x_3, y_3\}, \langle x_3 : rf \rangle = \{z_2\}$   
 $\langle y_3 : md \rangle = \{x_3, y_3\}, \langle y_3 : rf \rangle = \{z_2\}$   
 $T_3$ :  $r(y_3); r(z_2); r(u_1)$  が実行される時,  
 $T_{3,2}^W = \{x_3, y_3, z_2\}$   
 $T_{2,1}^W = \{x_1, y_1, u_1, z_2\}$

$T_1$  を始点,  $T_3$  を終点とする  $rf$  経路が存在し,  $P_{3,1}^W = \{x_3, y_3, z_2, u_1\}$  であるため,  $T_4$  がその部分集合から検索しているので,  $T_4$  は調整なしで実行できる.

$H_5$  においては,  $T_3$  までは  $H_4$  と同じであるが,  $T_4$  によって検索されるのは  $T_3$  と  $T_0$  の変更結果である. これ以上まとめできないので,  $H_5$  は調整なしで実行される.

異なる終点から得られる結果は同一のアイテムが存在すれば, バージョン番号も一致する場合にしか許可しないので, 次の結果がある.

[定理 3] RF 隔離性を満たすトランザクションからなるスケジュールにおける RF 隔離性アルゴリズムに従う実行結果は, RF 隔離性を満たす.

証明: RF 隔離性アルゴリズムに従う得られた結果が RF 隔離性を満たさないとする. 問題あるならば, 関数  $check(v)$  で得られる結果にあるか, あるいは合併後の結果に問題があるかのいずれかになる.  $check(v)$  関数では,  $md$  関連を利用して  $RA$  隔離性を満足させ,  $rf$  関連に沿っている経路を作成しながら  $Rset$  中のバージョンとの違いを比較し, 異なる場合は  $rf$  関連に従って修正している. 後者は, 同一のアイテムが存在すれば, バージョン番号も一致する場合にしか合併してない. このため, 実行結果は RF 隔離性を満たす.

## 6 おわりに

本論文では, トランザクションモデルを read only と write

only トランザクションから, read と write 操作の両方を含むトランザクションモデルへの拡張した場合に, read from 関係にあるトランザクションの変更結果に対して fractured read 現象が生じないための RF 隔離性を導入した, RF 隔離性の判定問題は, 判定グラフ MVSR(H) における 3 つ以内のトランザクション間の閉路問題に帰着した. さらに, RF 隔離性を満たすことが, 各トランザクションは, 自分の変更結果と一貫性上関連するデータは検索するという仮定のもとで, 検索一貫性を満たすための必要十分条件にであることを示した.

各トランザクションが正しく実行できるためには, 検索されたデータが一貫しなければならぬ. read only, Write only トランザクションしか存在しない場合には, RF 隔離性が RA 隔離性に一致するが, 検索操作と変更操作の両方が存在する場合には, 検索するアイテムの範囲に制限を設けない状況の下で検索一貫性を保証できる.

RF 隔離性を満たすことが必要十分条件となる応用例のまとめや RF アルゴリズムの実現などが, 今後の課題である.

## 文 献

- [1] Bailis, P., Fekete, A., Hellerstein, J.M., Ghodsi, A. and Stoica, I.: Scalable atomic visibility with RAMP transactions, SIGMOD, pp.27-38 (2014).
- [2] Bailis, P., Fekete, A., Ghodsi, A. Hellerstein, J.M., and Stoica, I.: Scalable atomic visibility with RAMP transactions, ACM Transactions on Database Systems (TODS) Vol. 41 Iss. 3, No. 15 (2016).
- [3] Bernstein, P. A., Hadzilacos, V., and Goodman, N.: Concurrency Control and Recovery in Database Systems, Addison-Wesley (1987).
- [4] 村田直郁, 川島英之, 建部修見, RDMA の適用による RAMP トランザクション処理の高速化, 情報処理学会論文誌データベース (TOD) Vol. 10, No. 2, pp.19-30 (2017).
- [5] 徐海燕, 古川哲也: ワークフロートランザクションの隔離性, 情報処理学会論文誌 (TOD) Vol.44, No. SIG 8 pp.55-64 (2001).
- [6] 徐海燕, 古川哲也, 史一華: 一貫性と隔離性の細分による並行実行の正当性の検証, 情報処理学会論文誌: データベース, Vol. 2, No. 1, pp. 22-32 (2009).

## 付録

### RF 隔離性アルゴリズム

‡  $T^R$  で検索された全てのアイテムのそれぞれのバージョン

$Rset = \{x_{1k_1}, x_{2k_2}, \dots, x_{mk_m}\}$

‡ 一貫性上関連する全対象の集合

$Rall = \{\}$

一貫性の  $Check$  を通った検索データの集合

$R_{確定} = \{\}$

function  $main(Rset)\{$

  for ( $max\ k : x_{jk}$  in  $Rset$ ) $\{$

$Rtmp = Check(x_{jk})$

    if ( $merge(Rtmp) == 0$ ) $\{$

      Print “部分 RF 隔離を満たす:”.  $R_{確定}$

    } $\}$  else  $\{$

```

    Rall に Rtmp を追加
  }
}
Print “RF 隔離を満たす：” . R確定
}

```

```

function Check(v){
  Rset から v を削除 , R確定 に v を追加
  foreach ( xjk1 in (v : md - v) )
    if ((xjki in Rset) and k1 ≠ ki){
      xjk1 を読み直す , Rset から除去
      R確定 に追加 ,
    }
    Rtmp に xjk1 を追加
  }
  for (v : rf から最大バージョン xjk1) {
    if (xjki in Rset){
      if (k1 ≠ ki) {
        xjk1 を読み直し ,
        Rset の xjki を xjk1 で替える .
      }
      Check(xjk1)
    } else {
      if (xjk2 in Rtmp){
        Rtmp に xjk1 を追加
      }
    }
  }
  Return(Rtmp)
}

```

```

function merge(Rtmp){
  foreach (xjk1 in Rtmp) {
    if ((∃xjk2 in Rall) and k1 ≠ k2) {
      return(0)
    }
  }
  Return(1)
}

```