4R－8

多重化データベースにおける仮想分割と再生成を用いた一貫性制御

千葉 佳史　多田 知正　樋口 昌宏　藤井 護

大阪大学 基礎工学部 情報科学科

1 はじめに

多重化データベースにおいて障害発生時に多重化度を維持する方法として再生成法 (regeneration)[3]が提案されている．また，ネットワーク分割が生じる仮定の下で多重化しているデータ間の一貫性を維持するアルゴリズムとして提案されている定数合意 (quorum consensus)[2]と再生成法を組み合わせた方式が提案されている[1]．定数合意はデータ項目の少なくとも過半数の複製にアクセスするため読み出しが遅くなるという欠点がある．また，定数合意に代わるアルゴリズムとして，定数合意のような問い合わせを行なわない仮想分割 (virtual partition)[2]が提案されている．本稿では，ネットワーク分割時に完全なデータベースを使用できるサイトがなるべく存在するように修正した仮想分割を再生成法に組みあわせた，定数合意と再生成法を組み合わせた方式より読み出しが速い方式を提案する．

2 準備

2.1 多重化データベース

多重化データベースでは各データ項目の複製が複数のサイトに配置されている．各サイトは多重化データベース中のどのサイトにどのデータ項目の複製があるかという情報を保持している．同じデータ項目に対する異なる内容を持つ２つ以上の複製が同時に有効にならないようにすることを一貫性制御という．サイト障害のみが生じ，ネットワーク分割は生じないという仮定の下では，一貫性制御は次のようにして行なわれる．トランザクションによるデータ項目の読み出しの場合は，トランザクションが発生したサイトに最も近い複製から値を読み出す．書き込みの場合は，全ての複製に対して書き込みを行なう．

2.2 再生成法

再生成法とは多重化データベースの多重化度を維持する方法として提案されている．多重化しているデータ項目の１つの複製がサイト障害により失われた場合，他の正常なサイトのうちの１つにそのデータ項目の複製を作ることでそのデータ項目の多重化度を保つ．従って，サイト障害により読み出し，書き込み不可能となるデータ項目がなくなることを防ぐことができる．すなわち再生成法により障害時のデータ可用性が上がる．再生成法は一貫性制御を考慮していないため一貫性制御を行なう方法と組み合わせて用いられる．

2.3 再生成法に定数合意を組み合わせたアルゴリズム

ネットワーク分割が生じた時に多重化データベースが複数の断片 (component) に分割される．この場合，１つのデータ項目について２つ以上の断片で読み出し，書き込みを実行するとデータベースの一貫性が破壊される．これを避けるため，トランザクションのデータ項目に対する読み出し，書き込みは，そのトランザクションが発生したサイトを含む断片中のサイトに過半数の複製が存在するデータ項目に対してのみ許可される．このことは読み出し，書き込みの前に各サイトに問い合わせ，アクセスを許可された複製の数が全体の過半数を越えている時のみ実行することで実現できる．この方法は定数合意と呼ばれている．

定数合意と再生成法を組み合わせた，サイト障害とネットワーク分割の両方に有効な方式が提案されている．しかし，定数合意はデータ項目に対する読み出し時，１つの複製の値を読み出すために，そのデータ項目の複製のある全てのサイトに問い合わせを行なうので，読み出しのオーバーヘッドが大きくなり，読み出しの速度が遅くなるという問題点がある．

3 提案方式

定数合意より読み出しの速度が速く，またネットワーク分割に有効な一貫性制御アルゴリズムとして仮想分割が提案されている．定数合意と仮想分割ではネットワーク分割時に各サイトはそのサイトが存在する断片に複製が過半数あるデータ項目に対する読み出し，書き込みが許可される．そのため，各サイトは一部のデータ項目へのみ読み出し，書き込みが許可され，全てのデータ項目に読み出し，書き込みを許可されるサイトがなくなるという状況が生じ得る．このことをデータベースの不完全化と呼ぶ．データベースの不完全化を

Consistency contorol with virtual partition and regeneration for replicated database
Yoshifumi CHIBA, Harumasa TADA, Masahiro HIGUCHI and Mamoru FUJII
Deperment of Information and Computer Sciences,
Faculty of Engineering Science, Osaka University
Toyonaka-shi, Osaka 560 Japan

なるべく回避するように仮想分割を修正した方法に再生成法を組み合わせることで完全なデータベースの可用性をより高くすることができる．以下，そのような方式について説明する．

各サイトは，そのサイトが通信可能であるとみなしているサイトの集合をビュー（view）として保持しておく．ビューと実際に通信できるサイトの集合が異なることを検出した時にビューは更新される．時刻 t に更新されたサイト A のビューは

$$V(A,t) = \{s \mid \text{サイト } s \text{ は時刻 } t \text{ に } A \text{ と通信可能}\}$$

で定義される．

また各サイトはどのサイトにどのデータ項目の複製があるかという情報を局所的に保持する．この情報を次のように定義する．

$$DB = \{(s,i) \mid \text{サイト } s \text{ にデータ項目 } i \text{ の複製が存在する}\}$$

ネットワーク分割時にデータベースの一貫性を保証するため定数合意と同じく１つのデータ項目については１つの断片でのみ読み出し，書き込みができるようにする．また，データベースの不完全化を避けるため，ビューと実際に通信できるサイトの集合が異なることを時刻 t に検出したサイト A はまず各データ項目 i について次の条件 (1) が成り立つかどうかを調べる．ただし $DS(i) = \{s \mid (s,i) \in DB\}$ とすると，

$$|DS(i) \cap V(A,t)| > |DS(i)|/2 \qquad (1)$$

条件 (1) を満たすデータ項目の集合を $C(A)$ とする．次に $C(A)$ に含まれるデータ項目が全データ項目数の過半数であるかを調べる．すなわち，データベース中の全データ項目数を n として $|C(A)| > n/2$ が成り立つ場合はそのサイト A を含む断片中のサイトに１つでも複製が存在するデータ項目に対する読み出し，書き込みを許可する．サイトが読み出し，書き込みを許可されたデータ項目の集合を可用データ項目集合とすると，サイト A の可用データ項目集合は次のように定義される．ただし各データ項目 i に対して $X(A,t,i) = \{s \mid s \in V(A,t) \cap DS(i)\}$ とする．

$$AD(A,t) = \begin{cases} \{i \mid X(A,t,i) \neq \phi\} & \text{if } |C(A)| > n/2 \\ \{i \mid X(A,t,i) = DS(i)\} & \text{otherwise} \end{cases}$$

以上のようにすることでどこかの断片では，その断片にある全てのデータ項目に対する読み出し，書き込みが許可される．

データ項目 i に対する読み出し，書き込みは i が $AD(A,t)$ に含まれることを確認し，$DS(i) \cap V(A,t)$ に含まれるサイトの i の複製に対して実行する．読み出し，書き込みを実行する時にビューと実際に通信可能なサイトの集合が異なることを検出した場合はビューを更新する．それにともない可用データ項目集合を更新する．また，そのビューに含まれるサイトのビューと可用データ項目集合を更新する．さらに，断片において可用データ項目集合中のデータ項目の複製の数が障害により減少している場合は，再生成法で複製を一定数まで増やす．それにともない DB を全てのサイトで更新する．このようにすることで断片中のデータ項目の多重化度を保ちデータ可用性を上げる．

4 定性評価

提案方式 (rv+)，再生成法と定数合意を組み合わせた方式 (rq)，再生成法と通常の仮想分割を組み合わせた方式 (rv) の３つの方式の特徴を読み出しのオーバーヘッド (read overhead)，障害回復のオーバーヘッド (recover overhead)，完全なデータベースの可用性 (availability) の項目について比較した．結果を以下に付す．

定性評価			
	rv+	rq	rv
read overhead	○	×	○
recover overhead	×	○	×
availability	○	×	×

提案方式はデータの読み出し時に局所的な情報を用いるため読み出しのオーバーヘッドが定数合意よりも小さくて済む．一方，障害が生じた時には全てのサイトでビューと可用データ項目集合を更新しなければならないため回復のオーバーヘッドは定数合意より大きくなる．提案方式はビュー更新時になるべく１つの断片で全てのデータ項目へのアクセスが許可されるようにするため完全なデータベースの可用性は高い．

5 まとめ

提案方式は障害回復のオーバーヘッド以外の点では他の方式より優れている．したがって，障害の発生することが少ない多重化データベースに用いた場合は効率的である．今後提案方式の詳細な定量評価を行なう予定である．

参考文献

[1] N. R. Adm and R. Tewari: “Regeneration with Virtual Copies for Distributed Computing Systems”, *IEEE Trans. Software Eng.* vol.19, no.6, pp. 594-602, June 1993.

[2] P. A. Bernstein, et al.: *Concurrency Control and Recovery in Database Systems.* Addison-Wesley, 1987.

[3] C. Pu,J. D. Noe,and A. Proudfoot: “Regeneration of replicated objects: A technique and its Eden implementation”, *IEEE Trans. Software Eng.* vol.14, no.7, pp. 936-945, July 1988.