# 弱い一貫性管理方式を用いた複製データのアクセス効率化

白田由香利†, 飯沢篤志†
(株) リコー 研究開発本部 ソフトウェア研究所
{shirota,izw}@src.ricoh.co.jp
(株) 次世代情報放送システム研究所
{shirota,izw}@ibl.co.jp

大規模な電子図書館システム、ビデオオンデマンド(VOD)システム、情報放送システムを構築する場合、利用者からのアクセスを効率良く行わせるためには、データ複製を行うことが有効である。本稿で想定する非常に大規模な分散データベース・システム(DBS)においては、複製も超多地点のデータベースサイトに作成される。しかし、データ複製を超多地点で行う場合、データの一貫性保持のためのコストが増大するという問題が起こる。2 相コミットプロトコルのような強い一貫性管理方式によりデータの一貫性保持を行うと、通信量の増大により実用化は困難となってしまう。これを解決するためには、弱い一貫性管理の適応が有効と考える。弱い一貫性管理方式にも各種の方式があるが、本稿では、対象となる応用分野ごとに、どのような方式が活用できるか、を具体的に論じていく。

# Efficient Access to Replica Data Using an Weak-Consistency Approach

Yukari Shirota† and Atsushi Iizawa†
Software Research Center, R & D Group, Ricoh Company, Ltd.
{shirota,izw}@src.ricoh.co.jp
Information Broadcasting Laboratories, Inc.
{shirota,izw}@ibl.co.jp

When constructing a large-scale digital library system, video demand (VOD) system, or information broadcasting system, data replication is useful to give users efficient access to the data. With a very-large-scaled distributed database system, a number of replica data are generated in a lot of distributed sites. As a result, it is costly to manage the global consistency among these many distributed replica data. The traditional approach of a strong-consistency control such as a two-phase commit protocol is inefficient and difficult to achieve across the internetwork. In contrast to strong consistency, a database replication system that provides weak consistency ensures greater availability of data. This paper presents some of the weak-consistency control methods used today and proposes which of the methods is the most appropriate for each target application.

† (株)リコーより(株)次世代情報放送システム研究所へ兼任出向中。
The authors are partly on loan from Ricoh Company, Ltd. to Information Broadcasting Laboratories, Inc.

## 1. はじめに

大規模な分散型データベースシステム(DBS)を構築する場合、利用者からのアクセスを効率良く行わせるためには、データ複製を行うことが有効である。本稿で対象とするDBSは「マルチメディア・コンテンツを提供し、その利用に対して課金を行うシステムである」とする。例えば、電子図書館システム、ビデオオンデマンド(VOD)システム、カラオケシステム、情報放送システムなどである。情報放送システムとは、人工衛星デジタル放送、地上デジタルテレビ放送、デジタルCATVなどのデジタル放送配信インフラを利用して、情報パッケージと呼ばれるマルチメディア複合データを各データベースサイトに配信するサービスである[飯沢 97]。情報パッケージの内容としては、放送番組表(EPG)、インデックス情報の他、選択性のある情報番組などがある。

本稿では非常に大規模な分散データベース・システム(DBS)上でデータ複製を行う際の、複製間の更新におけるデータ一貫性管理方式について論じる。

## 2. 対象となる分野の分析

上記のようなマルチメディアデータを提供する応用のためのDBSを構築する場合、その典型的DBとしては、以下の4種類が考えられる。

(1)コンテンツDB
- ・データサイズは超大
- ・複製が多数必要
- ・読み出す情報は多少古くても問題無し
- ・読み出し頻度は大
- ・更新伝播は遅延しても問題無し

(2)インデックス情報(2次情報)DB

インデックス情報とは、通常の書誌データなどの他、コンテンツを格納するサイトIDのような、それを参考にしてコストの安いアクセス経路を選択する場合などに使用できる情報を含む。
- ・データサイズは小〜中
- ・複製が多数必要
- ・読み出す情報は多少古くても問題無し
- ・読み出し頻度は大
- ・更新伝播は遅延しても問題無し

(3)利用者統計情報DB
- ・データサイズはシステム依存
- ・複製数は少なくてよい
- ・読み出す情報は多少古くても問題無し
- ・読み出し頻度は小
- ・更新伝播は遅延しても問題無し。

(4)課金情報DB

DBの属性は上記(3)の利用者統計情報DBと同じ。異なる点は、課金情報が損失することは許されないのに対し、利用者統計情報は例え損失しても問題が少ないことである。

また課金情報の操作の殆どは加算と減算だけであり、つまり交換可能かつ結合法則が成り立つ操作であるので、更新途中の課金情報が損失しない限り、更新伝播は遅延したとしても、計算値は正しい値に収束する。よって電子図書館などコンテンツ参照サービスを提供するシステムの多くは課金情報の伝播が遅れても問題ないと言える。実際の課金システム構築において、更新伝播の最大予測時間内に行われる利用者の消費(コンテンツの参照など)額を見積もることにより、最悪の被害額は推定できるので、逆算により、必要となる更新伝播速度を決めることができる。

また想定するDBSでの典型的トランザクションとして以下がある。
- ・利用者によるコンテンツ検索(頻度:大)
- ・コンテンツ提供者によるコンテンツ及びインデックス情報の更新(頻度:小)
- ・課金情報及び利用者情報の収集(頻度:大)

  各利用者に対する収集頻度は小さいが利用者数が膨大なので、全体としてのトランザクション発生頻度は大きくなる。

以下では、上記のような性質をもつ応用に対して超大規模分散DBSを構築する場合を想定し、そこでのデータ複製間更新処理の方式について論じる。

3. データ複製間の更新伝播方式の分類

データ複製方式は信頼性、アクセス可能性、及び性能の向上など利点をもつが、反面一貫性保持に要する手間が大きく通信コストが高いことが問題となっている。Gray らは、デッドロックの確率及びそれに伴うトランザクションの失敗の確率は、サイト数が 10 倍になると 1000 倍になると論じている[Gray96]。現状のデータ複製間での更新伝播方式は、大きく以下の 2 つの要素で分類できる。

(1) 更新を行うことができるサイトはどこか

集中管理方式：オブジェクト毎に主複製サイトを決め、そのサイトでのみ更新可能とし、他のサイトでの更新は許さない。

分散管理方式：どのサイトでも更新が可能。しかし更新を行おうとするサイトが全部の複製との調整を行う必要がある。

(2)一時的なデータ矛盾を認めるか否か

強い一貫性管理方式：データの矛盾を認めず一つのトランザクションの中ですべての複製間の整合性を保証する。例えば 2 相コミットプロトコルがある。

弱い一貫性管理方式：応用の内容に特化して考えることにより、データの矛盾を一時的に認めるもの。一つのトランザクションの中でシステム全体の一貫性が保証されるのではなく、更新の伝播は別のトランザクションとして行われていく。

弱い一貫性管理方式は初め共有化メモリで発案されたものであるが[Cher86], 分散 DBS のトランザクション管理においても各種の研究がなされている。この方式は以下のように大別することができる[白田 97a]。

(a)マルチバージョン方式

(b)差異限定管理方式

(c)意味ベース方式

我々が対象とする応用においては、更新伝播方式として、集中かつ弱い一貫性管理方式が適していると我々は考える。その理由は以下の通り:

(1) 集中管理方式は分散管理方式に比べてプロトコルが簡単であること、

(2) 集中管理方式は負荷集中という問題があるが、更新頻度は小さいと予想されるので問題無いこと、

(3) 弱い一貫性管理方式の場合、更新のための通信コストが少なくてすむこと、

(4)弱い一貫性管理方式では一時的にデータの一貫性が壊れるが、OLTP と違い、読み出す情報は多少古くても問題ない。よって更新伝播が遅延しても、サイト内でローカルにデータ一貫性が保たれている限り、問題無い。

3. 更新伝播制御方式のレベル付け

我々は、応用に応じたデータ複製の更新操作に対する 3 つの要求のレベル付けを行い、其々のレベルに対する適切な一貫性管理制御方式を検討することにした。

レベル A：すべてのトランザクションにおいて強い一貫性管理を必要とする。トランザクションは直列化可能となる。

レベル B：更新処理を含むトランザクションでは強い一貫性を必要とするが、read－only トランザクションでは多少古いデータでも問題ないので、弱い一貫性管理でよい。1 コピー直列化可能性[Bern87]だけは保証してほしい。

レベル C：更新処理を含むトランザクションにおいても、弱い一貫性管理を行いたい。

上記のようにトランザクションの性格でレベル分けした場合、同一のデータに対して異なるレベルが設定されることのないように応用を設計することは、データの整合性を保つ上で必要となる。

レベル C のトランザクションでは、更新伝播は複数のトランザクションに分かれてサイト間に伝播する。よって更新の競合が検出されても既にコミットされているので、通常の方式のアボートを行うことはできないという問題が起こる。

第 2 節で分析したような応用システムにおいてはレベル A のトランザクションは無い、と考えてよいだろう。原則的には全トランザクションはレベル B に属するが、性能向上のためレベル C が必要である場合が一部発生すると考えられる。

例えば利用者やシステムが検索に用いるインデックス情報や利用者統計情報の更新である。これらは一時的にデータ一貫性が崩れてもさほど問題は無い。例えば、間違ったインデックス情報でアクセスパスを計算しても、アクセスしてみれば間違いが判明するように、その処理結果が永続的に利用されるものではないからである。

## 4. 実現アルゴリズム

本節ではレベルB及びCのトランザクション処理の実現アルゴリズムについて述べる。

### 4.1 レベルBの更新処理

初めに、レベルBの更新処理ではどのような方式が適しているか考察してみる。

強い一貫性管理の更新アルゴリズムは従来から研究されており、データ複製における強い一貫性管理の制御方式としても、2相ロック‡、楽観的2相ロック、グローバルタイムスタンプに基づく楽観的方式、ベーシックタイムスタンプオーダリング、など各種がある。Careyらの評価では「完全に複製されたシステムでは、楽観的2相ロックが最も適している」という結果が得られている[Care91]。我々が想定する応用では、読み出しに比較して更新の頻度が極端に小さく、トランザクション間の競合は殆どないと仮定できるので楽観的制御方式が適切であると考える。

しかし楽観的2相ロック方式では、確認相で全ての複製サイトにロックをかける必要がある§。実際の運用では、ダウンしているサイトの可能性もあるため、楽観的2相ロック方式は、実現が困難であると言える。ダウンしているサイトがあっても処理が続けられるため方式としては以下の2つが考えられる。

(a)重み付き投票方式**[Giff79]

(b)集中管理でグローバルタイムスタンプに基づく楽観的方式

(b)で集中管理方式を用いた理由は第3節で述べたプロトコルの簡単であるという理由の他に、集中管理によりダウンしているサイトを無視できるようにするためである。集中方式では、一貫性の確認は主複製サイトにだけ問い合わせるだけで済むため、他の複製サイトにWlockをかけにいく(a)に比較して効率がよい。よって超多地点のサイトを含むことを考慮すると、(a)よりも(b)の方式が通信コストの増加が少なく、想定する超大規模分散DBSに適していると我々は考えている。

また(b)では、複数種類のオブジェクトにアクセスする必要があるトランザクションでは、初めにローカルサイトにそれらの複製を作成してから実行を始めることとする。これにより他のサイトと交信無しで処理を実行することが可能となる。(b)の方式の詳細については[白田 97b]の山彦方式のプロトコルを参照して頂きたい††。以後(b)をPTO(Primary-based Timestamp Optimistic)方式と呼ぶ。

### 4.2 レベルCでの更新処理

上記のPTO方式は、writeが含まれるトランザクションに対しては厳格つまり強い一貫性管理に基づく方式であった。レベルCでは更新トランザクションも緩和して弱い一貫性管理にする。

---

‡ データ複製においても、2相ロックプロトコルにより一つのトランザクションの中ですべての複製を更新するならば、直列化可能となる。

§ 集中管理に基づく2相ロック方式の場合は、トランザクションがreadしたい時、主複製に対してRlockをかけてから、ローカルデータを読み出す。writeの場合も同様で、主データにWlockをかけてから全複製データをwriteする。ロックは主複製にのみ行えばよいので負荷は軽減できる。しかし、全てのサイトでwriteできなければアボートしなくてはならない。

** 重み付き投票方式では、以下の条件を満たす一定数を満たすサイトが参加すれば、その中に必ず最新版の値が含まれることが保証されている。

Read-quorum + Write-quorum > n

Write-quorum + Write-quorum > n.

但し、nは複製の数を表す。

†† 山彦方式は楽観的制御方式の確認相に放送型配信機構を利用することを特長とする方式であるが、基本的なグローバルタイムスタンプのプロトコルには変わりはないからである。

これにより全体の効率をさらに向上させることができる。

レベルCの更新処理でもレベル B 同様、集中管理方式にして、衝突の検出は主複製サイトで行うこととする。衝突とは、例えば、サイト a とサイト b で同じデータオブジェクト O の複製を保持していて、両者のトランザクションで同じ時刻に O を更新し、両方のトランザクションがコミットした後、その更新ログが主複製サイトに届くような場合である。その際主複製サイトは先に届いた更新ログを承認し、後から届いた方の更新は却下することとする。

この場合、既にトランザクションはコミットにより完了してしまった後に衝突が検出されるので、通常のアボートはできない。そこで我々は、以下の解決方法を提案する。

主複製サイトにおける衝突検出及びコミット/アボート処理のアルゴリズムは以下のようになる。複製データはオブジェクト O に対して n 個あると仮定し、それを $O_1$,..., $O_n$ と表し、その主複製データを $O_0$ と表す。トランザクションTが保持するオブジェクト O の、最終更新タイムスタンプを t(O, T)と表わす。

*コミットを完了したトランザクション T は、コミット完了後に主複製サイト宛てに以下の情報から構成されるコミット情報を送信する。

・サイト ID 及びトランザクション ID
・write したオブジェクト(これを Oi とする)の情報: (OID, t(Oi,T), 更新後の値)
・read したオブジェクト(これを Oi とする)の情報: (OID, t(Oi,T))

*主複製サイトは、非主複製サイトからのコミット情報に対して承認判定を行う。アクセスした全てのオブジェクトに対して $t(Oi,T) \geqq t(O_0)$ が成立すれば承認され、一つでも不成立であれば却下される。

*主複製サイトは承認/却下メッセージをトランザクション T に送信する。

*却下されたトランザクションは、既に更新されていたオブジェクトに対して、自サイトにおいて更新をかける。最新値を保有するサイトは(1)主複製サイトと、(2)最後の更新を行なったサイト(これを最終更新サイトと呼ぶ)、の 2 つがある。我々は、負荷を主複製サイトに集中させないため、最終更新サイトから最終バージョンのデータをコピーした方が効率が良いと考える。その利点は次の通り: マルチメディアデータはサイズが大きいので、複製を作ることはコストがかかり問題である。提案の方式では、主複製上のトランザクションは複製を作らなくても済むため、応答時間が短縮される。

更新頻度の小さい応用では、write が衝突する可能性が少ない。そのような応用で、通常のアボート用にデータの複製を作ることは効率的でない。よって応用がデータの強い一貫性を求めない場合には、上記レベル C の実現アルゴリズムが適していると考える。

### 4.3 レベルB及びCでの更新伝播遅延

上述したレベル B 及び C の更新アルゴリズムでは、主複製サイトが更新を承認した後、他の全サイトにその更新内容を伝播させる必要がある。その際、弱い一貫性管理の手法を用いて、複製間の差異をどこまで許すか、という許容範囲を条件として与え、更新伝播回数を減らすこととする。このような差異限定管理方式は[Alon90, Kris91, Pu91, Side96]などで各種の研究が行われている。我々はそれらの制約条件を想定する応用に対して以下のように適応することを提案する。

(a) リフレッシュする周期を設定する。
主複製サイトはデータ更新情報をキューに入れておき、一定周期ごとに更新情報をブロードキャストする。受け側のサイトはこの更新情報が送られてくることにより、主複製サイトがダウンしていないことも知ることができる。

(b) キューにたまっている更新情報の数が一定数を超えたら伝播する。

(c) 更新トリガーとなるデータ領域を決めておく。
スキーマ中特に重要であり、すぐに更新を公

知したい場合である。例えば、書誌情報のうち、タイトルと作者と 2 つの属性に修正が行われた場合は、即時に更新を伝播する。

(d) データ更新伝播の閾値を設定する。
データの更新の度合いに対して閾値を設定しておく。例えば、コンテンツの 1/3 以上が更新あるいは追加されたら、更新伝播する、あるいは利用者からの閲覧回数が 100 を超えたら伝播するなど、である。

(e) バージョンの違いにより伝播する。
本の改定版などで、マイナーな誤字の直し程度であれば、更新伝播を待つ。しかし、大きな差異のある改訂版の出た場合は、即時更新を伝播する、などの場合である。

上記の差異限定ルールは組み合わせて設定することができる。

### 4.4 read−only トランザクション処理

前述の弱い一貫性管理方式による更新伝播遅延により、主複製サイト以外のサイトではデータの一貫性が崩れている可能性がある。しかし想定する応用では、読み出しに関しては最新のデータである必要がないものが殆どである。よってレベル B 及び C のいずれの場合でも read−only トランザクションは他サイトとの間で一貫性チェックを行なうことなくローカルに処理を実行することにする。これにより性能の向上が図れる。また、こうした一貫性管理の緩和を行なっても 1 コピー直列化可能性は保証される。つまり、他での更新が起こる前に読み出していた、と考えることで一つの資源上での一連の処理として直列化できる。

### 4.5. 新規登録トランザクション

本稿の対象とする応用システムでは、新規のデータを一括して登録することが多い。こうした新規登録トランザクションは、既にあるデータを参照しないで独立に処理することが可能である。よって上記のレベル B 及び C の更新処理の中で、こうした新規登録処理は別扱いとし、その楽観的制御方式の確認相の処理を省略することが可能である。つまり他のトランザクションと競合する可能性が無いので、確認を行わず、読み出し相が終わった後、直接書き出し相を実行することができる。

既存のデータと独立なデータを新規登録する応用は、独自のトランザクションレベル（例えば、新規登録処理フラグを設定する）を設けて、そのレベルであるトランザクションは一切競合チェックを行わないとすることにより効率化が図れる。

## 5. まとめ

本稿では、電子図書館システムや VOD システム、情報放送システムのようなマルチメディア・コンテンツを提供し、その利用に対して課金を行うシステムを超大規模分散 DBS として実現する場合の複製データの更新処理方式を検討した。応用の特徴として、更新が少ない、データサイズが大きいことから、並行処理制御方式としては集中的かつ弱い一貫性管理方式が適していると考えた。また、主複製サイトで更新を承認した後、全複製サイトへその更新結果を伝播する際も、想定する応用では伝播が遅延しても問題が無いので、複製の差異の許容限界を指定する条件を与える差異限定管理が効率的であると考えた。今後は、トランザクションのアクセス対象となるオブジェクトの集合が、各サイト上にローカルに存在するようにするため、各サイトのアクセスパターンからデータ複製の最適配置を決定する方式を検討していきたい。

## 参考文献

[Agra93] D. Agrawal and S. Sengupta: "Modular Synchronization in Distributed, Multiversion Databases: Version Control and Concurrency Control," *IEEE Trans. on Knowledge and Data Eng.*, Vol. 5, No. 1, pp. 126-137, Feb. 1993.

[Alon90] R. Alonson, D. Barbar a, and H. Garcia-Molina: "Data caching Issues in an Information Retrieval System," *ACM Trans. on*

*Database Sys.*, Vol. 15, No. 3, pp. 359-384, Sept. 1990.

[Bern87] P. A. Bernstein, V. Hadzilacos, and N. Goodman: *Concurrency Control and Recovery in Database Systems,* Addison-Wesley, 1987.

[Care91] M. J. Carey and M. Livny: "Conflict detection tradeoffs for replicated data," *ACM TODS*, Vol. 16, No. 4, pp.703-746, Dec. 1991.

[Cher86] D. R. Cheriton: "Problem-Oriented Shared Memory: A Decentralized Approach to Distributed System Design," *Proc. 6th Intl. Conf. Distributed Computing Systems*, pp. 190-197, 1986.

[Giff79] D. K. Gifford: "Weighted Voting for Replicated Data," *Proc. 7th ACM Symp. on Operating Systems Pinciples*, pp. 150-159, 1979.

[Gray96] J. Gray, P. Helland, P. O'Neil, and D. Shasha: "The Dangers of Replication and a Solution," *Proc. ACM SIGMOD 96*, pp. 173-182, Montreal Canada, June 1996.

[Kris91] N. Krishunakumar and A. J. Bernstein: "Bounded Igonorance in Replicated Systems," *Proc. 10th ACM SIGACT-SIGMOD Conf. on Principals of Database Sys.*, pp. 63-74, May 1991.

[Pu91] C. Pu and A. Leff: "Replica Control in Distributed Systems: An Asynchronous Approach," *Proc. 1991 ACM SIGMOD Conf. on Management of Data*, pp. 377-386, May 1991.

[Side96] J. Sidell, P. M. Aoki, A. Sah, C.Staelin, M. Stonebraker, and A. Yu: "Data Replication in Mariposa," *Proc. 12th Intl. Conf. on Data Engineering*, pp. 39-42, Houston, Nov. 1990.

[飯沢 97] 飯沢篤志、浅田一繁、白田由香利: 「情報放送のための超大規模分散データベースシステム」、情報処理学会研究会報告、97-DBS-113-44、1997。

[白田 97a] 白田由香利、飯沢篤志: 「情報放送システム上での並行処理制御方式の考察」、情報処理学会研究会報告、97-DBS-113-45、 1997。

[白田 97b] 白田由香利、飯沢篤志、矢野隆志: 「放送型配信機構上での並行処理制御方式:山彦」、ADBS'97 論文集、pp. 15-22、東京、1997。