マルチメディア通信と分散処理 91－23
（1999. 1. 22）

動的ドメインリーダに基づく複製管理方式の実装，評価

川崎 智広†，佐藤 文明‡，水野 忠則‡
† 静岡大学大学院 理工学研究科
‡ 静岡大学 情報学部

従来の複製管理プロトコルはリード／ライト性能のトレードオフや，複製数の増加によって性能が急激に低下するなどの問題がある．それに対して，我々は複製をドメインというグループにわけてリーダというドメイン内の代表を通じて他のドメインと通信を行うドメインリーダ方式を提案し評価してきた．しかし，従来のドメインリーダ方式ではアクセスに偏りがある場合に通信に無駄が発生することから，アクセス頻度に基づきドメイン構成を動的に変更することで性能の改善をはかる動的ドメインリーダ方式を提案する．本報告では，動的ドメインリーダ方式の詳細と評価から本方式の有効性を述べる．

キーワード：一貫性保証，複製，複製管理プロトコル，分散データベース

An Implementation and Evaluation of Replica Control Protocol Based on Dynamic Domain Leader Concept

Tomohiro Kawasaki†, Fumiaki Sato‡, Tadanori Mizuno‡
†Graduate School of Engineering, Shizuoka University
‡Dept. of Computer Science, Shizuoka University

Conventional replica control protocols have the problem of read/write performaince trade-off and the problem of the rapid decrease of the performance. Therefore, we have already proposed and evaluated Domain Leader protocol. The protocol divides the replica into the groups named domain, and the communicates with each other through the representative in the domain. However,if there is imbalance of arrival(access) rate, the performance of the protocol decrease. Therefore, we propose Dynamic Domain Leader protocol to improve the performance by dynamically changing the domain structure. In this report, we describe about the concept of the Dynamic Domain Leader protocol,and evaluation results.

keywords : Consistency Management,Replica,Replica Control Protocol,Distributed Database

1 はじめに

現在，分散データベースシステムで用いるデータの信頼性と可用性を向上するためにデータを複製化する技術が盛んになってきている．データを複製化することにより，ユーザが近い場所にあるデータをアクセスすることができ，またデータが保存されているディスクがクラッシュしても，他の場所にあるデータをアクセスすることができるので，データを処理する際に応答時間の短縮とデータを利用できる可能性が高くなる．複製化の技術が進むにつれて多数の複製に対応できるプロトコルが必要になると考えられる．

そこで我々は，既にドメインというグループにわけてグループ内の一つをリーダとし，更新はそのサイト自身とリーダのみで更新するという

ドメインリーダ方式を提案し，評価してきた．

今回提案する動的ドメインリーダ方式は，ドメインのリーダを変更することでシステム全体のアクセス頻度にばらつきがある場合に対応できるように改良したものである．

以下第2章で，従来の複製管理プロトコルの問題点を検討し，ドメインリーダ方式の紹介をする．第3章では，ドメインリーダ方式を改良した動的ドメインリーダ方式ついての詳細を述べる．第4章では，実装し評価した結果より考察をする．第5章では，本稿のまとめと今後の課題について述べる．

2 ドメインリーダ方式

2.1 対象システムのモデル

ドメインリーダアルゴリズム [5, 6, 7] は，図1ように同一のデータアイテムを持つ複製をグループに分割し，各グループをドメインとして扱う．各ドメインはリーダという特殊なサイトがトランザクションの管理を行い，リーダ間の同時実行制御は，プライマリリーダという特殊なリーダが行う．

ドメインリーダでは，データの正当性と一貫性を保証し，リーダとリーダの通信を制御するために，二相コミット [3, 4] を用いる．データベースはアプリケーションやユーザが共有するので，データの一貫性を保証するために，書き込みトランザクションを実行する際にはデータベースをロックする必要があり，リード要求を処理する際には，ロックをかけなくても良いので，ロックは二相ロック [3, 4] を使う．

各サイトの複製データには，バージョン番号が含まれている．全リーダのバージョン番号は，書き込みトランザクション毎に一斉にかわる．これにより，データが新しいものであるか古いものであるかはデータ本体を見なくてもサイトのバージョン番号とリーダのバージョン番号だけで判断できるようになる．

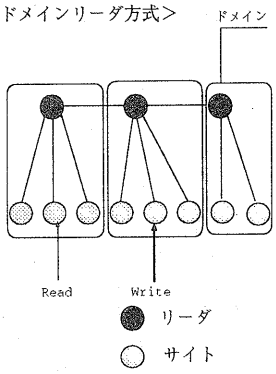

図 1: 動的ドメインリーダモデル

2.2 システムの動作

要求受け付けサイトで，自己の複製がロックされていると，以下の動作は読みだし，書き込みともにロックが解除されるまで待たされる．また，書き込みについてはいったんデータを読み出してから書き込むものとする．本節では，サイトを例にして到着した要求がどのような順序で処理されるかを説明する．

1. 読みだし動作
   - 読みだし要求到着:
     サイトのバージョン番号が最新のものかどうかリーダに問い合わせる
   - リーダからバージョン結果到着:
     リーダと同じ最新バージョンであれば当サイトのデータをそのまま読み出す．サイトが古いバージョンであればリーダからバージョン番号と一緒に最新データをもらい，データの更新をして読み出す．
2. 書き込み動作：
   - 書き込み要求到着 (サイト):
     書き込み要求をリーダに送信
   - サイトから書き込み要求到着 (リーダ):
     プライマリリーダに書き込み要求を送信
   - プライマリリーダから書き込み準備完了到着 (リーダ):
     他のドメインのリーダ全てに書き込み要求を送信

- 他のドメインのリーダ全てから書き込み準備完了到着 (リーダ):
  サイトに書き込み準備完了を送信
- リーダから書き込み準備完了到着 (サイト):
  リーダに書き込みデータを送信
- サイトから書き込みデータ受信 (リーダ)
  プライマリリーダと他の全てのリーダに書き込みデータを送信
- プライマリリーダと他の全てのリーダから書き込み完了到着 (リーダ) リーダはデータを更新してサイトに書き込み完了を送信
- リーダから書き込み完了到着 (サイト) サイトのデータを更新する.

プライマリリーダ内のサイトに要求が到着した場合は，ドメイン内のリーダがプライマリリーダでなくリーダでないサイトに到着した場合のトランザクション処理の手順に含まれるので省略する.

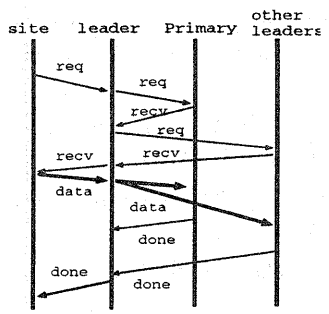

図 2: 通信手順

3 動的ドメインリーダ方式の提案

3.1 ドメインリーダ方式からの改良点

サイトとリーダで通信コストは下表のようになる.

サイトとリーダの通信コスト		
	site	leader
読みだし	2	1
書き込み	1＋ドメイン数	ドメイン数

表でも分かる通りドメインの中でリーダがサイトより通信コストが低いことがわかる. このシステムでコストパフォーマンス的に最も良いのは，リーダがドメイン内のサイトの中で最も発生率が高い時である. よって，発生率がドメイン内の他のサイト (リーダも含む) よりとても高いサイトについてはそのドメイン内のリーダとした方が良い. そのため，図 3のようにリーダにリーダの変更を行うかどうかを管理する変更テーブルを設ける. 変更テーブルを見て，ドメイン内の処理がある一定値に来た時にリーダを変更するかチェックし，チェックした結果が変更する条件をクリアしてればリーダを変更する.

しかし，リーダ変更にも通信コストがかかるため頻繁にリーダ変更を行うとドメインリーダ方式より性能を落とすことにつながってしまう. リーダの変更は，サイトとの発生率と比較して適度に行わなければならない.

3.2 リーダの変更手順

リーダの変更テーブルによって変更するかどうか決めるわけだが変更には以下の項目を必要とした.

- ドメインの総処理数
- ドメイン内の各サイト (リーダを含む) の処理数，ソケット情報

リーダの変更手順を図 4で表す.

変更チェックする以前の段階 :

リーダは各サイトからのメンバ登録を受けたら変更テーブルに登録し，一つのトランザクションを処理するごとにドメイン内の総処理数とリーダの変更テーブル内の該当するサイトの処理数を一つ増やしていく.

リーダの変更決定 :

ドメイン内の総処理数が変更値に到達した

場合，リーダの処理の割合と一番多く処理したサイトの処理数の割合が2倍以上あるか比べ，2倍以上はなれている場合にリーダの変更を行う．

リーダの変更：

- ドメインの全サイトと他のドメインのリーダにリーダ変更を伝える
- リーダはドメインの全サイトと他のドメインのリーダから変更準備完了を受信
- リーダは新しくリーダになるサイトに変更準備完了とキューに溜った処理を送信
- 新しくリーダになるサイトは，全リーダとドメイン内の全サイトにリーダの変更の設定するように要求しその間にリーダはサイトの設定をする
- 新しくリーダになるサイトはドメインの全サイトと他のドメインの全てのリーダから変更終了のメッセージを受信してリーダの変更を終了する

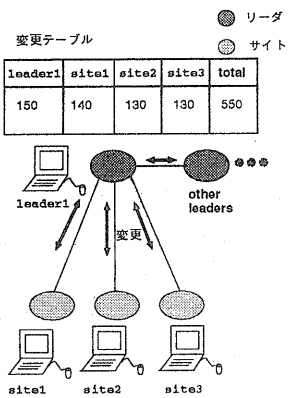

図 3: リーダ変更

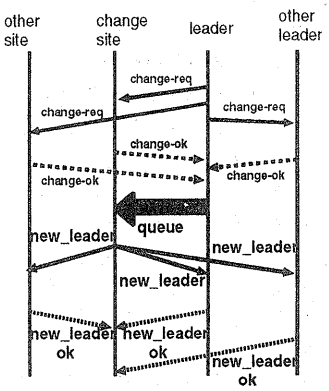

図 4: リーダ変更手順

4 実装と評価

4.1 実装環境

ドメインリーダ方式と動的ドメインリーダを以下の実装環境で実装し評価した．

1. 通信手段はソケットで，通信プロトコルは UDP を使用する
2. マシンは Sun Spark station4 を 20 台でオペレーティングシステムは全て SunOS 5.5 である
3. ネットワークは Ethernet 10Mbps, 同一セグメントの LAN を使用する
4. ネットワークの負荷の状態はほとんどかかっていない
5. 複製の種類は 1 種類である

4.2 サイト数の増加に対する変化

既にドメインリーダ方式では，シミュレーションでサイト数の増加に対して性能の低下が緩やかであることは分かっている．図 5では，実装でもシミュレーションと同様の結果が得られた．

4.3 リーダ数の比較

サイトの数が同じでドメイン数の構成をかえて最適なリーダ数を求めた．図 6よりリーダの数が 2 の時，最も性能がよかった．また，最適なリーダ数を割出すのには不十分ではあるが，リー

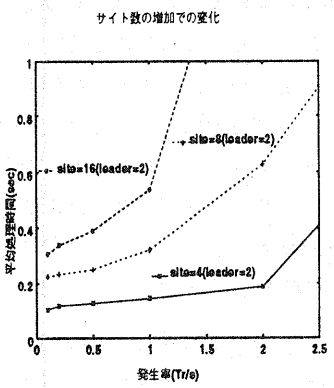

図 5: サイト数の増加に対する変化

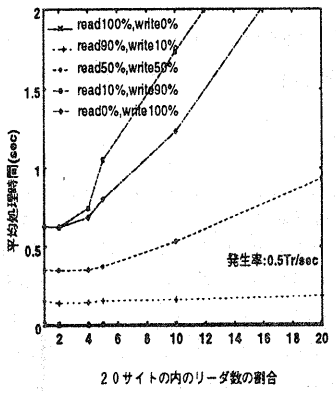

図 6: リーダ数の比較

ダ数をより多くすると，書き込みトランザクションのコストが増えてしまい，性能が悪くなっていることが分かる．これは，既にシミュレーションで確認しているがドメインリーダ方式には最適なリーダ数があり，最適な値に設定する必要があることが実装でも確認された．

4.4 リーダ変更率の比較

ドメインの中でリーダの処理数と最も処理数の多いサイトの処理数があまりかわらなければリーダを変更することで逆にドメインリーダ方式よりも性能が落ちてしまうことが考えられる．動的ドメインリーダ方式をリーダの変更率 (1/ 変更チェックを行う処理数) をかえてドメインリーダ方式と比較した．全サイト数は 20 で一つのドメインのサイト数は 4，全体のリーダ数は 5 である．

リーダを含むすべてのサイトの要求発生率はドメインの中で一つだけ 1 秒間に二つのトランザクション要求を発生し，それ以外のサイトは一つの要求を発生するようにした．また，1000 秒経つとローテーションで 1 秒間に二つのトランザクションを発生させるサイトをかえるようにした．

その結果，図 7で見て分かる通り，ドメインのトランザクション処理数が 10000 または 5000 の時にリーダ変更をしてしまうと今現在，最も発生率の高いサイトではないサイトをリーダと変更してしまうためリーダ変更にかかるコストを含めて，性能が落ちてしまっている．しかし，それ以上に変更率をあげてみると性能が良くなっていることが分かる．これは，発生率の高いサイトとリーダを変更することによって，性能が良くなることが分かった．

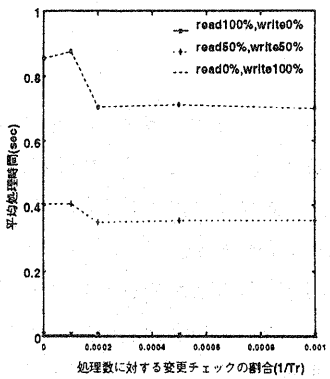

図 7: リーダ変更率の比較

4.5 占有率での比較

リーダを変更するべきサイトのイベント数が，システム全体のイベント数の中で占有の割合が高い場合，そのサイトの通信時間がシステム全体の通信時間と大きくかかわってくる．したがって，ドメイン内での占有率が高ければ高い程そのサイトをリーダに変更すればシステム全体の通信時間が大きく縮小されると考えられる．

図8では，最もイベントの発生頻度が高いサイトのドメイン占有率が20リーダに変更することで通信時間が大きく縮小されたことが分かった．

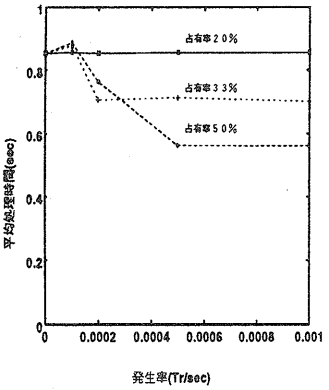

図 8: 占有率での比較

5 まとめ

ドメインリーダ方式がまだシミュレーションでしか評価されていなかったが実装した結果，複製の増加に対する性能の低下が緩やかになり，リーダ数を最適値に設定すればシステムの性能がかなり向上することが分かった．

また，イベントの発生頻度の高いサイトとリーダを変更する動的ドメインリーダ方式ではドメイン内のサイト (リーダを含む) で発生頻度にバラツキがある場合ドメインリーダ方式より通信時間が短縮できることも確認できた．

しかし，今回の実装環境ではサイト間の通信路はすべてつながっていて，しかも通信遅延はほとんど無かった．実際の通信路はすべてのサイト間で完全につながっているわけでもなく通信時間もばらばらである．したがって発生頻度は高いが他のドメインのリーダとの通信時間が長ければそのサイトをリーダに選んでしまうと逆に性能が落ちてしまう．以下の項目を今後の課題として研究を進めて行く．

- リーダの変更テーブルに通信時間の項目を追加してモバイル端末など通信速度の違うサイトでも対応できるようにして評価
- リーダ数の最適値の割りだし
- 定数合意方式や $\sqrt{N}$ アルゴリズム等の別の複製管理方式との性能比較

参考文献

[1] M.Stonebraiker.: "Concurrency Control and Consistency of Multiple Copies of Data in Distributed INGRES," IEEE Trans. Software Eng,vol.SE-5,no.3,May,1979,pp188-194

[2] D.K.Gifford.: "Weighted voting for replicated data," in Proc. 7th ACMSIGOPS Symp. Oper. Syst. Princip.,CA,Dec.10-12,1979,pp.150-159.

[3] 中川路 哲男 著.: "OSI 分散トランザクション処理技術解説" (株) ソフト・リサーチ・センター

[4] 斎藤 忠夫 監修, 石坂 充弘 編著.: "情報通信プロトコル" オーム社

[5] 中村健二, 宮西洋太郎, 佐藤文明, 水野忠則.: "ドメインリーダに基づく複製管理方式のモーバイル環境適用への評価" 情報処理学会研究報告, 情処研報 Vol 96,no.288,pp.43-48,1996 年 10 月.

[6] 宮西洋太郎, 中村健二, 佐藤文明, 水野忠則.: "分散システムにおけるデータの複製管理方式" 情報処理学会論文誌,vol.37,no.5,1996 年 5 月.

[7] 中村, 宮西, 佐藤, 水野.: "ドメインリーダに基づく複製管理方式の評価と改良" 情報処理学会論文誌,Vol.39,No3,pp.716-724