分散システムにおけるデータの複製管理方式

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管理において，読取りアクセス性能および可用性の観点から，複製の複数配置が行われる．その場合には，複製間の一貫性維持のための通信および更新処理が必要となり，全体の性能，特に更新性能に対する影響が懸念される．また，近い将来予想されている企業内外を共有データで連携する場合には，不均質なシステム間での相互連携動作が必要である．一方，無線通信を含むシステムでは，ネットワークとの接続状態に依存しない可用性が求められる．これらに対し本論文は，2つの複製管理方式を考察した．第一は，従来に比べて著しく多数の複製を持つ場合を想定した複製管理方式で，企業組織などの単位で特定サイトをドメインリーダと指定し，データの更新はドメイン内全サイトの複製更新の代わりにドメインリーダの複製のみを更新する．これにより一貫性維持に要する時間を削減する．数値計算によりその有効性を評価した．第二は，一貫性維持の時間削減およびネットワークから切離し時の可用性維持を目標とする複製管理方式である．サイトごとにデータ更新時の上限値を設けて，その範囲内でサイト独自の更新を許容し，アクセスの閑散時に全体の一貫性を回復するという方式で，シミュレーションにより有効性を評価した．この方式はデータ内容の意味情報を利用しているので，適用範囲には制限があるが，性能向上およびネットワーク分断時の可用性の向上に有効である．

Some Proposals for Replication Control Method in Distributed Systems

YOHTARO MIYANISHI,[†] KENJI NAKAMURA,[††] FUMIAKI SATO[††] and TADANORI MIZUNO[††]

Replicating data and placing them several sites is a usual method to increase read performance and to increase data availability in distributed data management systems. In such systems, overall performance, especially write performance, will be decreased because keeping consistency among replications derives additional traffics and additional write operations. Recently, enterprise integration by shared data is becoming an important topic for future industry. In such circumstances, each system should be able to work cooperatively among extremely heterogeneous data management systems. On the other hand, the network using radio telecommunication is becoming popular but its connection is rather unstable. Stable data availability is desirable in such network. We propose two types of replication control protocols for such needs. One is "domain leader" method which is devised for the case that there are extremely many replicas rather than conventional concept of replications. A certain site is designated as a domain leader which has rights and duties about the replicas among the sites in its domain. Instead of updating all replicas in the domain, the leader's replica is to be updated. The performance will be improved, as we show the results of numerical calculation. The other one is "modestly optimistic concurrency control" method. The upper limit of updating is set for each site and each data item. The updating request within the upper limit is permitted to execute independently at each site. The overall consistency will be recovered when the requests arrive rarely, e.g. at midnight. The performance will also be improved, as we show the results of simulation. As this method utilizes the semantic information of data contents, the scope of application will be rather limited, but it is effective for the improvement of performance and for increasing the availability when the network happens to be divided.

† 三菱電機
Mitsubishi Electric Corp.
†† 静岡大学
Shizuoka University

1. はじめに

分散システムにおけるデータ管理システムを設計する際に，データの一貫性，アクセス性能，および信頼

性・可用性が重要な留意点である．読取りアクセス性能の観点からアクセス要求の到着地点への複製の配置や，信頼性の観点から複数地点への複製の配置が行われる．その場合，複製間の一貫性維持のためのトラフィックおよび書込み処理が発生し，全体の性能，特に書込み性能に対する影響が懸念される．

また，最近の通信インフラストラクチャは高速化，大規模化の傾向とともに，無線通信による移動体通信も含めた多様化の傾向が見られる．無線による通信を行う場合には，ネットワークの接続状態はしばしば接続，切離しの状態の間を変動することが予想される．そのような状況でもシステム全体としては，連続的に使用できることが望ましい．

一方，将来の企業形態として，企業内，企業間，および製品の提供者・使用者間を有機的かつダイナミックにデータを介して連携を行い，作業者間での協調作業および企業内データの有効活用をはかり，諸々の企業活動の効率を高めていくことが提唱されている．このような場合には不均質なシステム間で，大規模なシステム間の相互連携動作が求められる．

本論文は，今後出現すると予想される複製管理についての上記のニーズを，

- ネットワークと切り離しても，ある程度の処理が継続できるための複製管理，
- 一貫性維持のためのオーバヘッドが過大とならない複製管理，
- 個別システムの内部に立ち入らないデータの複製管理，すなわち不均質なシステムを前提とする複製管理，

として把握し，これらの観点から，2つの新しい複製管理方式について考察する．

2. 関連する研究

近年，分散データベースシステムにおいて，アクセスの並行処理制御やデータ相互間，複製間の一貫性維持のために，ロック操作を用いた二相ロック（2PL）方式や，二相コミット（2PC）方式が実用化されている．

データの配置に関しては，従来から多くの研究がなされてきた[1)~6)]．従来の研究のほとんどは，複製はデータの可用性を満たす最小の数におさえるという考えである．これに対して本論文で検討する方式の1つは，複製は可用性だけはなく，後に述べるような事情により，多数存在するという前提での検討を行う．

複製管理のための並行処理制御については，大きく分けて3つの方式が研究されてきた[7)]．

- ロックを用いる2PLまたは2PC方式
- タイムスタンプを用いる方式
- 楽観的並行処理制御方式[8)]

また，ネットワーク分断時のデータの管理として，投票・定数方式が研究されてきた[9)]．

次に，データアクセス要求の種類がデータ更新である場合について，各方式の問題点を整理する．

ロックを用いる方式は，ロック後に全複製の更新を行うため，ネットワークへの負荷，複製保持サイトへの更新の負荷，および1つの更新要求が完了するまでに時間を要するという問題がある．

タイムスタンプ，楽観的並行処理制御は，衝突がほとんど発生しないという前提の方式であり，もし衝突が検知されると，事後の無効処理（UNDO）が必要となるという問題がある．

投票・定数方式はネットワーク分断時に半数以上のサイト（ノード）を含む部分と半数未満の部分の部分ネットワークに分断されるが，過半数側の方がデータを維持するという方式である．過半数側に入れなかったサイトでは，データを使用できなくなるという問題がある．すなわち，一貫性は維持されるが，可用性は低下する．

本論文では，複製の一貫性維持における性能の面からの複製管理方式，およびネットワーク分断時の可用性の面からの複製管理方式について述べる．

本論文で述べる複製管理方式の1つに類似する研究として，Triantafillou[10)]の研究がある．この研究は複製の一貫性維持の性能問題を扱ったものであり，リーダと呼ばれる1つの複製との間で2PCを行うことにより，オーバヘッドを減少させるという方式である．これは本論文での方式に類似する考えであるが，本論文での方式は，実際のネットワークは地域や組織単位で境界すなわちドメインが形成される場合が多いので，地域または組織単位ごとにリーダを特定することにより複製管理の処理を簡略化できる，という考えに基づいている．

本論文は，3章，4章で，考察した方式をそれぞれ述べ，まとめを5章で行う．

3. ドメインリーダ方式

3.1 ドメインリーダ方式の概要

この方式は，近い将来に予想されている企業間連携[11)]におけるデータの相互利用を想定している．このような環境では，各企業のシステムは個別に発展してきているので，全体としてはまったく不均質なシステムとなっている場合が大半であると予想される．このようなシステムでの相互運用は，初期の段階で

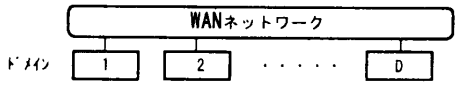


図1 対象分散システム（全体）
Fig. 1 Total system.

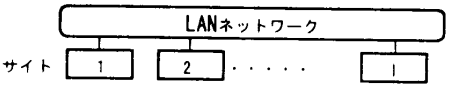


図2 対象分散システム（ドメイン内）
Fig. 2 System in a domain.

は，メッセージ受渡し型が主な形態であると予想されるが，企業間の連携が進み，システムの目的指向が強まるに従って，データの直接的な相互利用のニーズが高まってくると予想される．たとえば現在でも，同一CADシステムを用いた均質システムではあるが航空機の国際協同開発において，設計データを共有して，並行的に解析，設計作業を行うという実施例も報告されている．文献11)では，CALS (Continuous Acquisition and Lifecycle Support：生産・調達・運用支援統合情報システム）の考えが紹介されているが，上記のようなデータの相互利用の形態は，CITIS (Contractor Integrated Technical Information Service）と称され，論理的な連携を実現するデータベースはIDB (Integrated Data Base）と称されている．

本方式は，このような不均質で，複製を各システムで任意に保有しているような環境において，ある組織単位に，ドメインリーダと称する特定のサイトに処理の権限，義務を集約することにより，複製の一貫性維持に要する時間を軽減し，不均質さにも対応しようとする管理方式である．

3.2 対象システムと対象データ

検討の対象となる分散システムを**図1**および**図2**に示す．図1は全体システムを示し，WAN型のネットワークにD個のドメインが接続している．図2は各ドメイン内のシステムを示し，LAN型のネットワークにI個のサイトが接続している（Iの値はドメインによって異なるが，簡単のため同一とする）．各ドメインには，1つのドメインリーダが存在するものとする．ここでのドメインは，企業ならば1つの部門とか工場の単位を想定し，ディスクデータのテープへのバックアップなどの複製管理を行う組織の単位を想定している．また，前述のように従来は複製の数を可用性を満たす範囲で少数におさえるという考えが通常であったが，本方式では，複製は至るところに存在するという前提である．

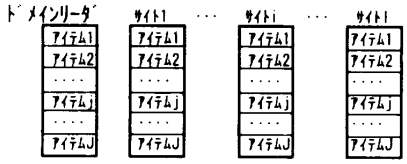


図3 対象データベース
Fig. 3 Objective database.

図3には検討対象のデータベースを示す．対象となるデータは，J個のデータアイテムからなり，データアイテムj単位にアクセスされ，またアクセス要求は地域的に分散した複数ドメインの中の1つのサイトiに到着し，サイトiまたはサイトiの所属するドメインリーダでアクセスの管理，すなわち調整者の役割，がなされるものとする．

3.3 システム各要素の動作

データのアクセスには読取り，書込み，更新（読取り後新たなデータを作成して書込みを行う）の操作があるが，動作の記述として，読取り，書込みの場合を述べる．

(1) **要求受付サイトの動作**

要求受付サイトで，自己の複製がロックされていると，以下の動作は読取り，書込みともに，ロックが解除されるまで待たされる．

- 読取り要求受付時

データの読取り要求を受け付けると，当サイトは自己の所属するドメインリーダに，自己の複製の現在状態を送信して，最新状態か否かを問い合わせる．当サイトの複製が最新状態ならば，最新状態であるという情報のみがドメインリーダから返答され，データの転送は行わない．最新状態でなければ，最新データをともなう返答がなされる．

- 書込み要求受付時

データの書込み要求を受け付けると，各サイトは自己の所属するドメインのリーダに書込み要求およびデータを転送し，自己のドメインリーダとの間で2PC処理を行う．動作の管理，すなわち調整者の役割はドメインリーダが行うものとする．

自己のドメインリーダがドメイン内またはドメイン間の2PC処理にすでに入っている場合は，通常の2PC処理と同様に，動作を待たされる．

当該サイトからは，自己が属するドメインリーダのみと2PC処理を行っているように動作する．

(2) **ドメインリーダの動作**

ドメインリーダの複製がロックされていれば，ロックが解除されるまで，以下の処理は待たされる．

● 読取り要求の場合

要求受付サイトの複製の現在状態を判断し，最新状態になっていれば，最新状態であるという情報のみ返答する．最新状態になっていなければ，状態情報とともにデータを返答する．

● 書込み要求の場合

ドメインリーダは他のドメインリーダとの間で，2PC処理を行う．各ドメインリーダの複製はドメイン内の複製を代表する．

すなわち，あるサイトに到着したアクセス要求は，そのサイトの所属するドメインリーダに中継されて，全体のドメインリーダ間で2PC処理によるデータの書込みが行われる．リーダ間で2PC処理による処理が完了すると，要求到着サイトに対しては，このアクセス要求は完了したもの，すなわちコミット完了と返答する．

各ドメインリーダは引き続き，自己の複製をロックのままで，ドメイン内のサイトにデータを転送する．

各ドメイン内の全サイトから完了の通知があると，またはタイムアウトすると，各ドメインリーダはドメイン内の処理も完了したとして，自己の複製のロックを解除する．

もし，自己のドメイン内の全サイトへのデータ転送において，未完了またはタイムアウトがあっても，すでに要求サイトへのコミット完了を行っているので，このアクセス要求はアボート扱いにはできない．

したがって，あるドメイン内では，ドメインリーダのデータは全体と一貫しているが，サイトは必ずしも一貫しているとは言えない．ネットワークが異常の場合やサイト自体が異常の場合には，最新のデータは伝達されていない可能性がある．

このような場合でも，ドメインリーダはドメイン内の各サイトのデータの現在状態を管理していて，状態に応じた動作（転送の実行または不実行，最新状態の回答）を行う．

(3) 要求受付以外のサイトの動作

● 書込みの場合

各サイトが所属するドメインリーダからデータが転送されてくると，自己の複製をロックし書込みを行う．書込みが完了すると所属するドメインリーダに完了を通知する．

上記の動作の概要を図4に示す．

3.4 ファイルサーバ方式との比較

上記の動作は，各サイトに複製を持たず，ドメインリーダのみに1つの複製を持つ従来のファイルサーバ方式と類似してくるが，ファイルサーバ方式と比較して，本方式の特徴を述べる．

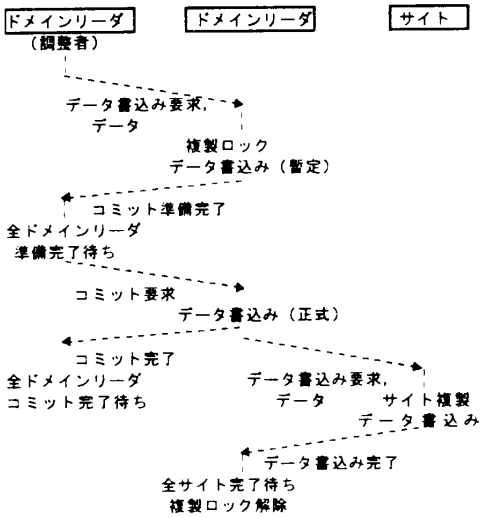


図4 システムの動作
Fig. 4 Behavior of system.

● 本方式は，近い将来の不均質，大規模システムを前提にしているため，各サイトに複製がすでに存在していることを前提にしている．すなわち，各サイトでのアプリケーションプログラムは，各サイトごとのデータ管理システムの下で動作していることを前提としている．この既成の構成をできる限り少ない変更で，新たな連携システムを構築したいというニーズ，すなわち既存のデータをファイルサーバに集約するということを行わずに対応したいというニーズが予想される．既存のデータは連携前は複製ではなく，独立したデータであり，連携後はデータの様式はそれぞれ異なるが，論理的にあるデータの複製になるという状況が想定される．ただし，各サイトではデータベースは個別に発展してきているので，ここでのモデルのような単純なデータ構造ではなく，各種の変換を行うことが必要となる．

● 近い将来のモーバイル環境などで，各サイトがネットワークから切り離されても，旧データなら読取り可能という機能を提供できる．書込みについても，別途何らかの手段（たとえば，本論文の後半で述べるような方法や文献12)に提案している方法）を用意することにより，一時的に書込み可能とすることができる．

● タイムスタンプなどの方法により，旧データであると判断でき，旧データでも意味のあるシステム（たとえば，時系列データ，天気図，新聞，協同作成中の文書など）では，ネットワークから切り離された場合でも，使用可能である．

● 読取り要求時，最新データがサイトまで転送されている場合は，要求受付サイトとドメインリーダの間

を制御情報（問合せ情報，状態情報）のみが往復し，大量データの転送時間を省くことができる．

3.5 性能の評価

従来方式とドメインリーダ方式（ここでは本方式と略す）の性能を比較評価する．ここでの評価は2つの方式の相対的な比較が主な目的であり，比較が可能であるという範囲で評価方法を簡略化している．具体的には，次の条件を想定し，2つの方式を比較する．

- **システムの構成**：図1および図2に示す構成で，本方式の場合には，全体は D 個のドメインからなり，各ドメインには同数 I 個のサイトからなる．従来方式のサイト数は $D \cdot I$ 個である（サイト数を同じとする）．
- **サイトに保持する複製**：両方式とも，全サイトに全アイテムの複製を保持しているものとする．
- **待ち時間の扱い**：両方式とも，待ち行列による待ち時間は無視する．
- **ネットワークの種類**：両方式とも，各サイト間には同じ性能の通信路が確保されているものとし，通信機能が一斉通報機能（放送機能）を持たない場合と持つ場合の両者を評価する．
- **通信時間**：両方式とも，サイト間の通信時間は2種類（制御情報のみ，ユーザデータ）の一定値（確率変数ではない）とする．
- **評価項目**：両方式とも，書込み時間のみを評価する．読取り時間，更新時間についても評価は可能であるが，2つの方式の比較が目的であるので，書込み時間で代表させる．

両方式とも，アクセス要求受付サイトに書き込むべきデータが準備されているものとし，複製へのデータ転送開始から書込み完了までの時間で評価する．従来方式では，アクセス要求受付サイトが調整者で，各複製に向かってデータが転送される．本方式では，受付サイトからひとまず調整者としてのドメインリーダに転送され，そこから各複製にデータが転送される．

- **書込み時間の確率密度関数**：両方式とも，1つのサイト内の書込み時間は指数分布に従い，各サイトは独立に動作し，各サイトにおける平均書込み時間は同一とする．
- **書込み動作**：両方式とも，二相ロック（2PL）動作で評価する．前述のように，従来方式，本方式ともに書込み動作は二相コミット（2PC）方式で動作することを想定しているが，両方式の比較が目的であり，図4におけるデータ書込み（暫定）とデータ書込み（正式）の2段階の動作を簡略化し，1段階の書込み動作のみで両方式を比較する．

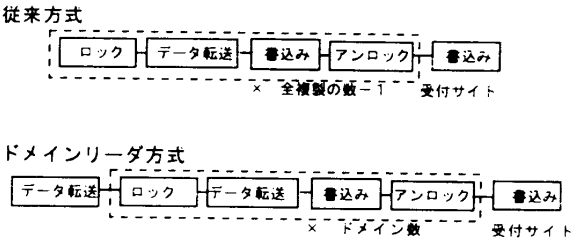


図5 評価モデル
Fig. 5 Evaluation model.

(1) 評価モデル

図5に示すモデルで性能の概略評価を行う．

従来方式の動作時間は図に示す破線内の処理が自己サイト（アクセス要求受付サイト）以外の全複製数，すなわち $(D \cdot I - 1)$ 回繰り返される時間であり，破線外は自己サイトの複製を書き込む時間である．図のロックおよびデータ転送は複製を保持するサイトへのロック要求の時間および書込みデータの転送時間である．また書込みの後には，書込み完了を返信する通信時間も存在する．図のアンロックは複製を保持するサイトへのアンロック要求の通信時間である．破線内および外の書込みは各複製を保持するサイトでの書込み時間である．実際の動作はアンロックが最後に行われるというように，図に示す順序では行われないが，全体の時間算定には図示した要素の時間で近似的に計算できる．

同様に本方式では，自己サイトから調整者としてのドメインリーダへのデータ転送後，破線内の処理がドメインリーダの数 D 回分繰り返され（ただし当該ドメインリーダの分はロック，データ転送，アンロックの通信部分は不要），それから自己サイトの書込みが行われる．

(2) 計算式

(a) 放送機能なしの場合

放送機能がない場合は，各複製への書込み動作は直列に行われる．したがって，従来方式の全複製への書込み時間 T_a および本方式のドメインリーダ複製への書込み時間 T_b は，次式で表される．

$$T_a = (T_1 + T_2 + T_{wo})(D \cdot I - 1) + T_{wo}, \quad (1)$$

$$T_b = T_2 + (T_1 + T_2)(D - 1) + T_{wo} \cdot D + T_{wo} \quad (2)$$

ただし，

T_1 はロック（往復），アンロック（往復），書込み完了，を通知する通信時間の合計であり，

T_2 はユーザデータ転送の通信時間であり，

T_{wo} は1つのサイトでの平均書込み時間である．

(b) 放送機能ありの場合

放送機能がある場合には，動作は並列して行われる．上記式 (1)，(2) のカッコ内の T_1 および T_2 は放送機能を使用し，1回の動作で行われ，一定値（確率変数ではなく）としているので，並列動作するのは自己サイトの複製も含めて各複製への書込み動作である．自己サイトの場合には並列動作開始時点が他のサイトとは異なるが同時開始であると近似する．書込み時間は指数分布であるとしているので，書込みすべてが完了する時間を T_w とすると，次のようにして求めることができる．

1つのサイトでの書込みが完了する時間を τ（確率変数）とすると，τ が時間 t 以下 ($\tau \leq t$) で完了する確率分布 $F_1(t)$ は次式で表される．

$$F_1(t) = 1 - \exp(-\mu t), \tag{3}$$

ただし，μ は処理率であり，$\mu = 1/T_{wo}$ である．

このようなサイトが並列に n 個ある場合，時間 t で，n 個すべてが完了する確率分布 $F(t)$ は，各サイトが独立して動作するので，各確率の積の次式で表される．

$$F(t) = \prod_{i=1}^{n} F_1(t) = (F_1(t))^n, \tag{4}$$

上式を2項定理で展開し，次式が得られる．

$$F(t) = \sum_{r=0}^{n} (-1)^r {}_nC_r \exp(-r\mu t), \tag{5}$$

上式を t で微分して確率密度関数 $f(t)$ を求める．$f(t)$ は指数分布の式（指数関数）の線形結合となっている．これから t の平均値 T_w を求める．

$$T_w = \int_{t=0}^{\infty} t f(t) dt, \tag{6}$$

であり，その結果次式が得られる．

$$T_w = \sum_{r=1}^{n} (-1)^{r+1} {}_nC_r (1/r) T_{wo}. \tag{7}$$

従来方式の場合は，$n = D \cdot I$（全サイト数）であり，この場合の T_w を T_{wa} とする．

本方式の場合は，$n = D + 1$（ドメイン数+自己サイト）であり，この場合の T_w を T_{wb} とする．

したがって，全体の書込み時間 T は次のようになる．

$$T_a = T_1 + T_2 + T_{wa}, \tag{8}$$

$$T_b = T_2 + T_1 + T_2 + T_{wb} = T_1 + 2 \cdot T_2 + T_{wb}. \tag{9}$$

(3) 数値計算例

上記の，従来方式と本方式の計算例を放送機能のない場合およびある場合を次の**図6**，**図7**のグラフに示す．1つのドメインに4つのサイトが含まれている場合で，ドメイン数をパラメータとして計算している．また簡単のため，T_{wo} を1（秒）とし，T_1，T_2 をそれぞれ0（秒）としている．

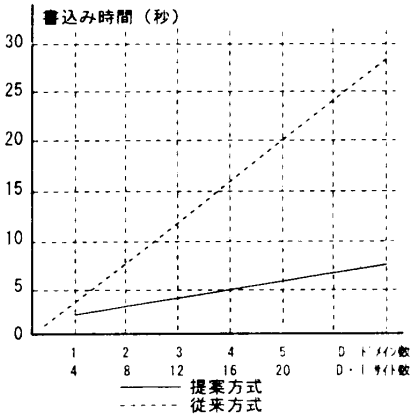


図6 書込み性能のグラフ（放送機能なし）
Fig. 6 Performance without broadcast function.

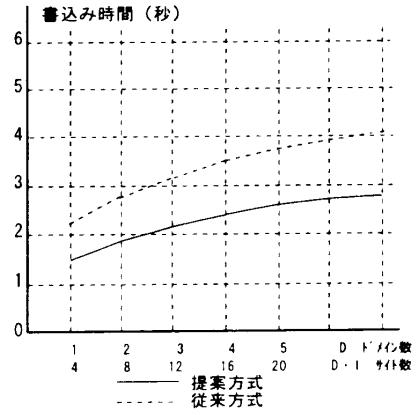


図7 書込み性能のグラフ（放送機能あり）
Fig. 7 Performance with broadcast function.

4. 制限付き楽観的並行処理制御方式

4.1 制限付き楽観的制御（MOCC）方式の概要

この方式は，データ更新の応答時間を改善するとともに，近い将来のモーバイル環境のように，ネットワークとサイトとの間の通信の接続が不安定な場合でも，サイトにおける処理が継続できることを目標とする方式である．その実現のためには，全複製への書込みを回避し，各サイトごとおよびデータアイテムごとにデータ更新の限界値を設定し，その範囲内ならば各サイト独自に更新することを許容する．サイト独自の更新により一貫性は維持されていないので，夜間等のアクセスの閑散時に全体の一貫性を回復する．またネットワーク分断時にも上限値範囲内での更新が可能

である．

更新内容にかかわらず楽観的に更新を行う通常の楽観的並行処理制御方式に比べて，更新動作は制限付きであり，modest である（慎み深い）ので，Modestly Optimistic Concurrency Control (MOCC: 制限付き楽観的並行処理制御）方式（ここでは本方式と略す）と名付けた．

ただし，本方式はデータ内容の意味情報を利用しているので，適用対象に制限がある．ここでの意味情報とは，あるデータアイテムの属性の値を操作（演算）する場合に，その演算に何らかの制約がある場合を想定している．たとえば，預金口座残高，在庫量，座席予約用座席数等のように共通の量を増減し，一定の範囲内（たとえば非負）に管理するような適用分野である．数学的には，演算されるデータの集合が加算による群，すなわち加群となる場合に適用できると言える．具体的には，データ間の演算が加算（+），減算（逆元の存在）であり，単位元（0）が存在し，交換則（$a+b=b+a$）が成立する場合，具体的にはデータの内容が加減算の演算を行え，大小の判断（順序関係）ができる整数（正数，負数，ゼロの値をとる．最小単位を考えると上記例の応用分野のデータは整数と見なせる）の場合に適用できる．

4.2 MOCC アルゴリズム

検討対象のデータの構成は，前記の図 2 において，LAN ネットワークまたは WAN ネットワークにサイト $(1,2,\cdots,I)$ が接続されていて，データは図 3 のようにアイテム $(1,2,\cdots,J)$ からなるものとする．

データアイテムのロック操作および更新処理は次の種類を定義する．

- 広域ロック：アイテム j の全複製にロックを行う．
- 狭域ロック：自己サイトにアイテム j の複製が存在していれば，自己サイト複製のみにロックを行う．
- 狭域更新処理：狭域ロックを行い，自己サイトの複製の現在値を更新する．
- 広域更新処理：広域ロックを行い，全複製の現在値を更新する．

(1) 準備状態

- データベースの維持を担当するサイト（ホストと称する）を定めておく．
- データアイテムごと，またはアイテムの群ごとに，ある 1 つのデータタイプ属性を持たせ，当該アイテムまたは当該アイテム群に対して本アルゴリズムを適用可能か否かを表現する．

データタイプ 0：本アルゴリズム適用不可

データタイプ 1：本アルゴリズム適用可能

- アイテム j の現在値を x_j とする．
- サイト i ごと，データアイテム j ごとに，現在の値に対してどの程度の範囲まで更新を許可するかという指標として，更新の限界比率（limit rate）r_{ij} を持つ（$r_{ij}=0$ ならば読取りのみ許可）．

$$\sum_{i=1}^{I} r_{ij} = 1. \tag{10}$$

- サイト i ごと，データアイテム j ごとに，限界比率と現在値によって定まる更新の限界値（limit value）u_{ij} を持つ．

$$u_{ij} = r_{ij} \cdot x_j. \tag{11}$$

- 限界比率は，適切なアルゴリズムによって定める．たとえば，限界比率決定アルゴリズムとして，
  - アクセス頻度に応じて重み付けを行う．
  - 特定サイトに重みを持たせる．

(2) アクセス要求到着時

サイト i に，アイテム j の更新要求が到着する場合を考える，更新は現在値からある値を減少させる要求で次式で表されるとする．

$$\text{更新要求：}\ x_j := x_j - \Delta x_j. \tag{12}$$

Δx_j は x_j の変化分

次のとき，本アルゴリズムの処理を行う．

- アクセス対象のデータアイテムが本アルゴリズム適用対象である（データタイプで判断）．

データ更新処理は次のように行う．

- ネットワークが正常の場合
  - 更新要求が

$$\Delta x_j \le u_{ij}. \tag{13}$$

ならば，狭域更新処理を行う．

  - 更新要求が

$$\Delta x_j > u_{ij}. \tag{14}$$

ならば，広域更新処理を行う．

- ネットワークが異常の場合（要求受付サイトとホストの間でのコミュニケーションが途絶えている場合）
  - 更新要求が

$$\Delta x_j \le u_{ij}. \tag{15}$$

の範囲内ならば狭域更新処理を行う．

  - 更新要求が

$$\Delta x_j > u_{ij}. \tag{16}$$

ならば，更新要求を棄却する．

(3) 回復処理（同期処理）

一貫性の回復処理を行うタイミングとして：

1) 同期時刻

2) ネットワーク復旧時

がある．前述のように，データベースのホストの役割をあるサイトに定めておき，一定時刻にデータベース整合のための処理を起動する．この処理は広域更新処理とする．

- ホスト：「回復要求」の発行
- 各サイト：「回復要求」を受け取ると各サイトは現在の限界値 u_{ij} $(j=1,\cdots,J)$ をホストに回答する．
- ホスト：回答された u_{ij} を集計する．回答がなかったサイト i については，以前の限界値が保持されているものと見なす．回答があったサイトのみ限界値を再計算する．

$$SUM=\sum_{i=1}^{I}\delta_i\cdot u_{ij}. \tag{17}$$

ただしサイト i から回答があれば $\delta_i=1$，なければ $\delta_i=0$ とする．

回答があったサイト i に対して，次の値を新しい限界値 u_{ij} とする．

$$u_{ij}=SUM\cdot r_{ij}/(\sum_{i=1}^{I}\delta_i\cdot r_{ij}). \tag{18}$$

これを回答があったそれぞれのサイトに送信する．

- 各サイト：送られてきた u_{ij} をサイト i での新しい限界値 u_{ij} とする．

4.3 データの一貫性

本方式では，回復処理により全体の一貫性を回復するが，ある回復処理時点から次の回復処理時点まで，それぞれの複製におけるデータの値は，互いに異なる値となっている．したがって従来の一貫性の考えで判断すると一貫性が保たれていないということになる．しかし，対象および更新処理が4.1節および4.2節に従っていれば，コンピュータシステムの外の実世界の値（たとえば，預金残高，在庫量，座席数）に対して不都合な事態（たとえば，預金残高，在庫量，座席数が負になる）は防止できるし，回復処理により全体の一貫性もある期間内にはとれる．したがってシステムの運用ができることになる．

4.4 本方式適用の検討

本方式を在庫管理に適用することを想定し，具体例を述べる．在庫管理とは企業において，倉庫に部品などの物品がどれだけ保管されているかを管理する業務で，倉庫に物品が入ってくることを入庫，物品を倉庫から出して要求元に手渡すことを払い出しまたは出庫，また払い出しを予約することを引き当てと称する．在庫管理は，在庫切れの発生を少なくおさえ，払い出しが円滑に行われるように適正に在庫量を維持することである．

表1 在庫管理への適用例
Table 1 An application to inventory management.

拠点		A	B	C
部品1	限界率	0.4	0.2	0.4
在庫量 200	引当可能量	80	40	80
部品2	限界率	0.5	0.1	0.4
在庫量 400	引当可能量	200	40	160
部品3	限界率	0.3	0.2	0.5
在庫量 100	引当可能量	30	20	50
部品4	限界率	0.6	0.1	0.3
在庫量 500	引当可能量	300	50	150

物品の種類すなわち品目が J 種類の在庫管理を考える．ある1カ所に倉庫があり，品目 j の在庫量が x_j である．この物品を I カ所の拠点（サイト）から引き当てすることを想定する．拠点 i ごとに限界率 r_{ij} が設定されているとする．また1日1回営業時間外に前述の回復処理を行うとする．

この場合，拠点 i では $u_{ij}=r_{ij}\cdot x_j$ までの引き当ては，他の拠点とは独立して自由に引き当てすることができる．全体としての在庫量 x_j は回復処理の時点で整合させる．

数値例を**表1**に示す．倉庫に部品1，2，3，4が，表に示す在庫量をもっているとする．この在庫を拠点A，B，Cという3カ所から引き当てるものとする．各拠点で通常使用する頻度などを考慮して，表に示す各限界率が設定されているものとする．表の中の引当可能量は各拠点での各部品の独自に引き当て可能な量を示す．

この表の場合には，たとえば拠点Aでは部品2は200まで，独自に引き当てることができる．

4.5 シミュレーション結果

従来方式と比較し，本方式の効果を確認するために，シミュレーションを行った．

データアクセスの種別は書込み処理のみとした．比較対象とする従来方式は，アクセス要求を受け付けたサイトが調整者の役割をはたし，複製を持つ全サイトにロック要求の送信を行い，全複製のロック完了後，調整者から書き込むべきデータを複製を持つ全サイトに送信し，複製を持つ各サイトで個別に複製への書込みを行い，調整者は全複製の書込み完了を待ち，アンロック要求の送信を行い，全複製のアンロック完了を待ち，その後にアクセスを完了する方式である．

要求の到着はポアソン過程に従い，書込み時間は指数分布に従うものとする．

4.2節で述べたように，図2のネットワークに接続されたサイト数が3で，データアイテム数が4であ

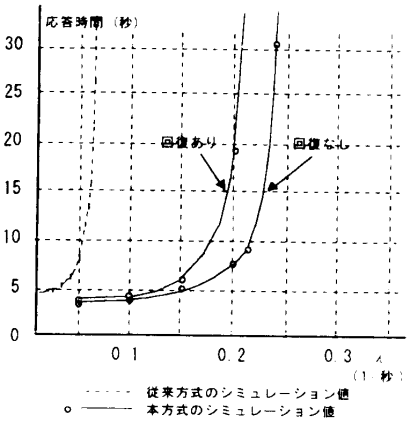


図8 従来方式と本方式
Fig. 8 Results of new algorithm.

り，全サイトに全データアイテムの複製を持っているものとする．

各サイト個別の平均書込み時間が1（秒），書込み要求がある頻度 λ_{ij}（各サイト i に到着する各データアイテム j の書込み要求の到着する頻度）で到着する場合の応答時間（書込み要求が到着してから書込みが完了するまでの時間）を計測した．

複数の複製データのロックおよび書込みを行う従来方式に対して，単一の複製のロックおよび書込みを行う本方式の性能比較が目的であるので，この場合の性能の主要要因ではない通信時間は近似として無視できるものとした．また回復時以外は，すべてが狭域更新処理で対応できるものとした．

λ_{ij} は i および j について同一とし，λ_{ij} をパラメータとして変動させた．

シミュレーションは20000秒を0.1秒刻みに離散化して行った．

このシミュレーション結果を図8に示す．書込み要求頻度の1/10の頻度で回復処理を行う場合と回復処理を行わない場合の結果を図示している．

図により，従来方式に比べて，本方式は応答時間および処理能力の点で性能が改善されていることが示されている．

5. おわりに

以上述べたことをまとめると，

(1) 近い将来，不均質データベースシステムを相互運用する際の複製管理方式として，ドメインリーダ方式について述べた．この方式は従来の必要最小限数の複製という考えとは異なり，随所に複製が存在するという前提での考えである．一貫性維持のための処理時間の削減および，ネットワークから分離時の可用性向上が目標であるが，前者を実現していることを数値計算により示した．

(2) 制限付き楽観的並行処理制御方式（MOCC）について述べた．この方式も，一貫性維持のための処理時間の削減およびネットワークから分離時の可用性向上を目標にしている．前者を実現していることをシミュレーションにより示した．

今後の課題として下記を検討する予定である．

(1) 意味情報の利用について，適用分野の分類，適用分野の拡張，および形式的記述の可能性を追求する．

(2) ドメインリーダ方式をシミュレーションにより有効性を評価する．

(3) システムの利用形態（アクセス頻度）と複製配置の関連を検討する．

(4) 単純データのアクセスから，トランザクション処理のような，複数データアイテムについての取り扱い方法を検討する．

参考文献

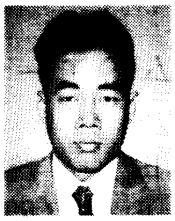
1) Chu, W.W.: Optimal File Allocation in a Multiple Computer System, *IEEE Trans. Comput.*, Vol.C-18, No.10, pp.885–889 (1969).
2) Mahmoud, S. and Riordon, J.S.: Optimal Allocation of Resources in Distributed Information Networks, *ACM Trans. Database Syst.*, Vol.1, No.1, pp.66–78 (1976).
3) Muro, S., Ibaraki, T., Miyajima, H. and Hasegawa, T.: Evaluation of the File Redundancy in Distributed Database Systems, *IEEE Trans. Softw. Eng.*, Vol.SE-11, No.2, pp.199–205 (1985).
4) Yoshida, M., Mizumachi, K., Wakino, A., Oyake, I. and Matsushita, Y.: Time and Cost Evaluation Schemes of Multiple Copies of Data in Distributed Database Systems, *IEEE Trans. Softw. Eng.*, Vol.SE-11, No.9, pp.954–959 (1985).
5) Jain, H.: A Comprehensive Model for the Design of Distributed Computer Systems, *IEEE Trans. Softw. Eng.*, Vol.SE-13, No.10, pp.1092–1104 (1987).
6) Ceri, S., Pernici, B. and Wiederhold, G.: Distributed Database Design Methodologies, *Proc. IEEE*, Vol.75, No.5, pp.533–546 (1987).
7) Coulouris, G.F. and Dollimore, J.: *Distributed Systems – Concepts and Design*, Addison-Wesley, p.366 (1988). 水野（監訳）：分散システム――コンセプトとデザイン，電気書院，京都，p.554 (1991).

8) Kung, H.T. and Robinson, J.T.: On Optimistic Methods for Concurrency Control, *ACM Trans. Database Syst.*, Vol.6, No.2, pp.213-226 (1981).
9) Thomas, R.H.: A Majority Consensus Approach to Concurrency Control for Multiple Copy Databases, *ACM Trans. Database Syst.*, Vol.4, No.2, pp.180-209 (1979).
10) Triantafillou, P. and Taylor, D.J.: The Location-based Paradigm for Replication: Achieving Efficiency and Availability in Distributed Systems, *IEEE Trans. Softw. Eng.*, Vol.21, No.1, pp.1-18 (1995).
11) 後藤：CALS: 21世紀における企業情報システムの国際標準確立と企業統合に向けて，情報処理，Vol.36, No.1, pp.1-7 (1995).
12) 宮西，中村，佐藤，水野：分散システム複製管理機能を有する共同文書作成支援についての提案，情報処理学会 DPS 研究会，95-DPS-71, pp.157-162 (1995).

(平成7年9月22日受付)
(平成8年3月12日採録)

宮西洋太郎（正会員）

昭和18年生．昭和43年神戸大学大学院工学研究科電気工学専攻修了．平成6年静岡大学大学院博士課程に社会人学生として入学，現在在学中．昭和43年三菱電機（株）入社以来，主に工業分野におけるオンラインリアルタイムシステムの構築にシステム技術者および管理者として従事した．現在同社情報システム製作所に勤務．主な研究テーマは分散システムの性能評価，複製の配置など．分散システムの設計，応用分野への適用に興味があり，特に航空分野のネットワークおよび企業間情報共有方式に興味がある．計測自動制御学会，システム制御情報学会各会員．



中村 健二（学生会員）

1970年生．1994年ブラジル国立パラー総合大学情報技術学科卒業．1994年静岡大学工学部研究生．1995年同大学大学院工学研究科情報知識工学専攻入学，現在に至る．主な研究分野は，分散データベース，モーバイルコンピューティング．

佐藤 文明（正会員）

昭和37年生．昭和59年岩手大学工学部電気工学科卒業．昭和61年東北大学大学院修士課程修了．同年三菱電機（株）入社．平成7年静岡大学工学部助教授．現在，静岡大学情報学部助教授．工学博士．分散処理システム，通信ソフトウェア開発環境，形式記述技法の研究に従事．電子情報通信学会，IEEE 各会員．

水野 忠則（正会員）

昭和20年生．昭和43年名古屋工業大学経営工学科卒業．同年三菱電機（株）入社．平成5年静岡大学工学部知識情報工学科教授，現情報学部情報科学科教授．工学博士．情報ネットワーク，プロトコル工学，モーバイルコンピューティングに関する研究に従事．著書としては，「プロトコル言語」(カットシステム)，「MAP/TOP と生産システム」(オーム社)，「分散システム入門」(近代科学社)，「分散システム—コンセプトとデザイン」(電気書院) などがある．電子情報通信学会，IEEE 各会員．