

分散システムにおけるデータアクセス性能解析と更新アルゴリズム

宮西 洋太郎†、中村 健二‡、佐藤 文明‡、水野 忠則‡
 †三菱電機 ‡静岡大学工学部

分散システムにおけるデータベースアクセス時には、並行処理制御や一貫性維持のために、ロック操作が用いられる場合が多い。データベースが複製を有する場合の更新処理は通常、全ての複製へのロック操作の後に全複製の更新処理が行なわれる。また読み込み処理も読み込みロックの後に行われる。全複製へのロック操作および読み込み、書き込み処理は応答時間低下の原因の一つとなりうる。

本論文では、分散システムのモデルを考察し、待ち行列解析により近似的にデータアクセスの応答時間を求める方法を示す。また、応答時間性能の低下を避けるために、全複製にわたる広範囲なロックおよび更新処理を行わないで、データ更新を行なう方式を提案する。この方式はデータ内容の意味を利用するので適用分野には制限があるが、応答時間の改善とネットワーク分断時のデータ更新に対して有効である。

Analysis of Data Access Performance in Distributed Systems and A Proposal of a Data Update Algorithm

Yohtarō Miyanishi*, Kenji Nakamura+, Fumiaki Sato+, and Tadanori Mizuno+

* Mitsubishi Electric Corp.

7-10-4 Nishigotanda Shinagawa, Tokyo, 141

+ Shizuoka University

3-5-1 Johoku, Hamamatsu, 432, Japan

LOCK operations are usually used in the case of updating the distributed database. When the database has several replicas, the data may be updated after all replicas have been locked. Locking and updating of all replicas tends to degrade the performance. We analyze the response time for accessing data by using a queueing model.

Then we propose a new updating algorithm which avoids locking and updating of all replicas. As this algorithm uses the semantics of data, the applications will be limited. However, it will be effective for keeping the performance well and it will be applicable to updating in the situation of network partition.

1. はじめに

分散システムのトランザクション処理におけるデータベースアクセスは通常、読み込み時には読み込みロック、書き込み時には書き込みロックが行われてアクセスが行われる。性能上、読み込みロック中の読み込み要求はブロックされない。書き込みロック中は読み込み、書き込みともに要求はブロックされる。複製を有する場合には、全複製へのロック操作及び読み込み、書き込み処理はデータ更新応答時間性能低下の原因の一つとなりうる。本発表では、ロック動作および読み込み、書き込み処理が応答時間に与える影響について、待ち行列解析により近似的に応答時間を求める方法を示す。また、応答時間性能の低下を避けるデータ更新方法を提案する。

2. 関連する研究

分散データの更新処理において、データ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、2PL方式、2PC方式が実用化されている。

分散システムの性能解析の多くは待行列モデルを用いて行われている[1][2]が、ロックとの関連については、あまり検討がなされていない[3]。

待ち行列を主な解析手段としない研究もある。例えば文献[4]では、性能に関連するシステムの構造が複雑、不明確である場合に、統計的な観測値から性能評価式を導出しようとする方法であるが、全体のシステム構造そのものを設計する立場からは、このような方法は適用しがたい。文献[3]

は評価指数としてコストを設定し、コストを最小にするような設計方法である。しかし、コストとともに、応答時間が重要な設計着目点である。

本発表の3.では、解析の準備を行う。4.ではデータアクセス（読み込み及び更新）処理を単純化し、「全複製へのロックを行ない、全複製のアクセスを行ない、その後ロックを解除する」というモデルで、アクセス応答時間性能に焦点を絞った待ち行列モデルによる性能解析を行なう。

同様の主旨で発表した前回[5]からの進展は、①サイト間通信時間の考慮、②読み込み、更新の混在、③タンデム型の処理時間分布の考慮、④ロックとアクセスの待ち行列動作の検討、⑤サーバ能力分割方法を応答時間を基準にした、などである。

一方、性能向上を目的とした方式として、楽観的並行処理制御方式が研究されている[6]。またネットワーク分断時の複製維持方式として投票-定数方式[7]やデータの内容を利用する更新方式も研究されている[8]。楽観的制御方式は、衝突がほとんど発生しないという前提の方式である。衝突の場合には訂正操作が必要である。定数方式では、ネットワーク分断時等において、一貫性は維持されるがアベイラビリティは低下する。

本発表の5.では応答性の向上、ネットワーク分断時のアベイラビリティ向上上の目的でデータ更新アルゴリズム案を紹介する。

3. 対象データのモデル

データアクセスは、読み込みロック及び書き込みロックによって排他制御される場合が多い(表1)。

表1 読み込みロックと書き込みロック

	読み込みロック	書き込みロック
読み込みロック中	前進	ウエイト
書き込みロック中	ウエイト	ウエイト

単純なデータアクセスの場合には、上記の読み込みロックの論理的なロックと物理的な動作とは、ほとんど同時であるので、読み込みが動作をブロックされるのは、読み込みの物理的操作自体と書き込みロックである。従って、排他制御は書き込みロック一種類のみとして近似する。

3. 1 ロックとデータアクセスとの関係

ここではロックとデータアクセスを論理的サーバと物理的サーバという関係で整理を行う。ここでの、サーバは待ち行列理論での用語である。

(1) 二つのアクセスグループの場合

まず、二つの論理的なアクセスのグループがあり、物理的には一つのサーバ（ディスク、CPU）、これを物理的サーバとここでは称する、を共有しているモデルを考える。

グループごとにロックを行い、ロックが確保された物理的サーバに処理要求を行う。物理的サーバはFIFOにて処理を行い、処理が終われば、該当するグループのロックを解除する。これを図1に表す。それぞれの到着率、サービス時間を、 $\lambda_1, s_1, \lambda_2, s_2$ とする。

論理的アクセスの処理部分を論理的サーバとここでは称する。論理的サーバの前段には論理的待ち行列、物理的サーバの前段には物理的待ち行列をもつものとする。

物理的待ち行列には、ロックが確保された要求だけが入るので、同一グループの要求は高々一つ存在するのみである。

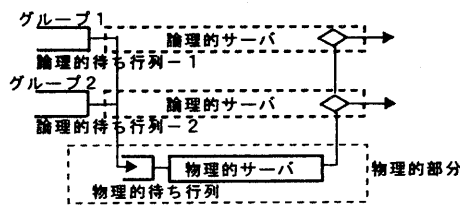


図1 ロックモデル

論理的待ち行列に待ち中の要求の数を含めたモデル全体の状態を考え、この状態遷移をマルコフ過程として、定常状態での各状態の確率を計算することが考えられるが、待ち中の要求数に制限はなく、状態が複雑となる。

そこで、近似として物理的部分（待ち行列及びサーバ）のみに着目し、5つの状態の近似的な確率を求め、応答時間 $T1, T2$ を計算する。

これを、二つの数値計算例とシミュレーション結果を表2に示す。（ $s_1 = s_2 = 1.0$ ）

表2 数値計算例とシミュレーション結果

	数値計算	シミュレーション
$\lambda_1 = 0.7273$	$T1 = 5.264$	$T1 = 5.280$
$\lambda_2 = 0.0727$	$T2 = 2.033$	$T2 = 2.043$
$\lambda_1 = 0.4545$	$T1 = 1.999$	$T1 = 1.980$
$\lambda_2 = 0.0455$	$T2 = 1.569$	$T2 = 1.618$

これにより、近似方法がほぼ妥当とみなせる。

到着頻度の少ない方が物理待ち行列の中に同種の要求が存在する確率は小さく、現在処理中の次に処理順序が確保されやすい。従って、結果的に到着頻度の少ない方が処理順序が優先されて、応答時間が速くなる。

(2) 複数のアクセスグループの場合

上記の近似方法でもグループの数が多くなると従って物理的部分（サーバ、待ち行列）の状態の数が階乗で増加し、計算が困難となる。そこで、限定された数の状態で近似することを試行したが、良い結果は得られなかった。

そこで、図2に示す近似的モデルを考える。

グループgの到着率、平均処理時間、処理時間の分散を $\lambda_g, s_g, \sigma_g^2$ とする。

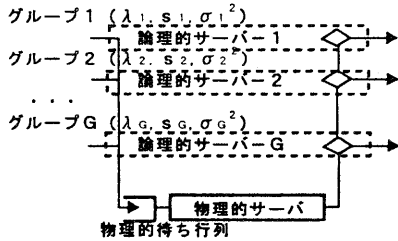


図2 近似的モデル

アクセスのグループ数をGとすると、以下の式により、物理的待ち行列における待ち時間 T_w 及びグループgの応答時間 T_g が求められる。

$$\lambda = \sum_{g=1}^G \lambda_g, \quad \rho = \sum_{g=1}^G \lambda_g \cdot s_g$$

$$S = \sum_{g=1}^G \frac{\lambda_g}{\lambda} \cdot s_g, \quad \sigma_s^2 = \sum_{g=1}^G \frac{\lambda_g}{\lambda} \cdot \sigma_g^2$$

M/G/1のポラチェックヒンチンの公式 [9] から $T_w = \rho S (1 + \sigma_s^2 / S^2) / (2(1 - \rho))$

$$T_{qg} = T_w + s_g$$

このモデルでは上式のように、グループによる相違は処理時間 s_g の相違のみとなる。

この状況は、意味がないわけではなく「処理の優先順序は、頻度の相違に依存するものではなく、明示的な優先度の指定によって行うべきである。」という一つの考え方を実現することでもある。これは物理的待ち行列のメカニズムをつぎのように変更することにより近似的に実現できる。

「各グループの要求にタイムスタンプを含ませ、物理的待ち行列に要求が到着するとき、そのタイムスタンプにより要求発生順に待ち行列の中を並び替える。」

システム全体としての性能解析には、上記のような論理的サーバの待ち行列ネットワークにより、応答時間を算定することが必要であるので、物理的サーバの能力(1とする)がどの程度、論理的サーバに配分されるかを考える。

それぞれの論理的サーバはM/M/1であると考え、その応答時間が同一の値になるようにサーバ能力が配分されていると考えると、グループgに配分されたサーバの能力 C_g は、次の応答時間の式から求められる。(左辺は前述、右辺はM/M/1)

$$T_{qg} = (s_g / C_g) / (1 - \lambda_g (s_g / C_g))$$

から、 $C_g = \lambda_g \cdot s_g + s_g / T_{qg}$

この配分されたサーバ能力で、論理的グループそれぞれの処理時間(仮想処理時間と称す)

$s_g' = s_g / C_g$ を求める。これにより物理的サ

ーバを共有しているという事実から、近似的に解放され、図3のように、各グループに仮想的に専属のサーバが接続されているものとみなせる。

このサーバ能力分割計算を行う部分を「計算パッケージ1」と称する。

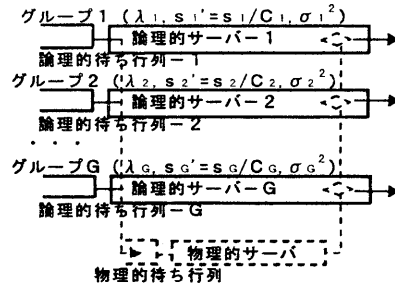


図3 論理的サーバによるモデル

3. 2 直列、並列合成サーバの処理時間

システムの中の随所に現れる直列、並列合成処理の処理時間、分散について検討する。

図4で表されるモデルを考える。並列数はデータ複製へのアクセスを想定しているので、高々3とする。また直列数はアクセス要求の通信、アクセス処理、返答の通信を想定しているので、これも高々3とする。また並列処理の完了は、AND型とOR型があるものとする。各要素の平均処理時間を μ_{ij} とし、指数分布に従うものとする。

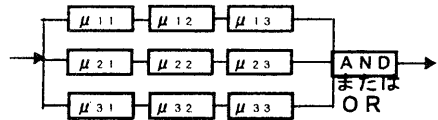


図4 直列、並列合成サーバ

このモデルの平均処理時間 \bar{t} と処理時間の分散 σ_s^2 を求める。直列部分は確率密度関数の畳み込み積分で、並列部分は確率分布関数の積で全体の確率分布関数を求め、時間で微分して、密度関数に戻すことにより算出できる。その結果は、

$$\bar{t} = \sum_{i=1}^3 \sum_{j=1}^3 \sum_{k=1}^3 C_{1i} C_{2j} C_{3k} (1/\mu_{1i} + 1/\mu_{2j} + 1/\mu_{3k} - 1/(\mu_{1i} + \mu_{2j}) - 1/(\mu_{2j} + \mu_{3k}) - 1/(\mu_{3k} + \mu_{1i}) + 1/(\mu_{1i} + \mu_{2j} + \mu_{3k}))$$

$$\bar{t}^2 = \sum_{i=1}^3 \sum_{j=1}^3 \sum_{k=1}^3 C_{1i} C_{2j} C_{3k} (2/\mu_{1i}^2 + 2/\mu_{2j}^2 + 2/\mu_{3k}^2 - 2/(\mu_{1i} + \mu_{2j})^2 - 2/(\mu_{2j} + \mu_{3k})^2 - 2/(\mu_{3k} + \mu_{1i})^2 + 2/(\mu_{1i} + \mu_{2j} + \mu_{3k})^2)$$

$$\sigma_s^2 = \bar{t}^2 - (\bar{t})^2$$

ただし、

$$C_{11} = \mu_{12} \mu_{13} / (\mu_{11} - \mu_{12})(\mu_{11} - \mu_{13})$$

$$C_{12} = \mu_{13} \mu_{11} / (\mu_{12} - \mu_{11})(\mu_{12} - \mu_{13})$$

$$C_{13} = \mu_{11} \mu_{12} / (\mu_{13} - \mu_{12})(\mu_{13} - \mu_{11})$$

$$C_{21} = \mu_{22} \mu_{23} / (\mu_{21} - \mu_{22})(\mu_{21} - \mu_{23})$$

$$C_{22} = \mu_{23}\mu_{21}/(\mu_{22}-\mu_{21})(\mu_{22}-\mu_{23})$$

$$C_{23} = \mu_{21}\mu_{22}/(\mu_{23}-\mu_{22})(\mu_{23}-\mu_{21})$$

$$C_{31} = \mu_{32}\mu_{33}/(\mu_{31}-\mu_{32})(\mu_{31}-\mu_{33})$$

$$C_{32} = \mu_{33}\mu_{31}/(\mu_{32}-\mu_{31})(\mu_{32}-\mu_{33})$$

$$C_{33} = \mu_{31}\mu_{32}/(\mu_{33}-\mu_{32})(\mu_{33}-\mu_{31})$$

ほぼ同様にOR型の平均処理時間と分散も求められる。(ここでは省略する)

平均処理時間と分散が求めれば、M/G/1のポラチェックヒンチンの公式から待ち時間も計算できる。上記を1×1から3×3まで対応できるようにして「計算パッケージ2」と称する。

4. システムのモデル

(1) 対象データとアクセスパラメータ

検討の対象となる分散システム、データベースをそれぞれ図5、図6に示す。LAN型のネットワークにI個のサイトが接続している。通信時間は、どのサイトからどのサイトへの時間も同じとする。

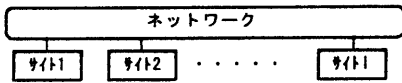
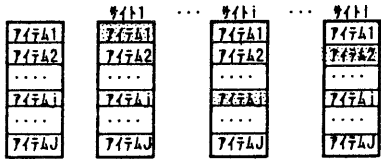


図5 対象分散システム



例えば、ハッチング部分には複製が存在する

図6 対象データベース

データのアクセスには読み込み操作、書き込み操作及び更新操作があるが、本発表では読み込み操作及び更新操作を検討する。また一般的には書き込みデータを生成するために、事前に読み込みを行うことが必要である。

対象となるデータは、J個のうちのデータアイテム(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サイトの内の一つに到着し、そのサイト(i)でアクセスの管理(協調者の役割)がなされるものとする。システム全体の動作を定める次の変数を定義する。

- λr_{ij} : サイトiにデータアイテムjの読み込み要求が到着する到着率(1/Sec)
- λu_{ij} : サイトiにデータアイテムjの更新要求が到着する到着率(1/Sec)
- d1: サイトiからサイトi'までの通信時間 (タイプ1: 制御情報のみ)
- d2: サイトiからサイトi'までの通信時間 (タイプ2: データを含む)
- X_{ij} : 複製存在変数(2値変数)

$X_{ij}=1$ ならばサイトiにアイテムjの複製あり
 $X_{ij}=0$ ならばサイトiにアイテムjの複製なし
 sr_{ij} : サイトiに複製がある場合の、この複製単独の平均読み込み時間(Sec)

$$\mu r_{ij}: \text{読み込み処理率}, \mu r_{ij} = 1/sr_{ij}$$

sw_{ij} : サイトiに複製がある場合の、この複製単独の平均書き込み時間(Sec)

$$\mu w_{ij}: \text{書き込み処理率}, \mu w_{ij} = 1/sw_{ij}$$

(2) データのアクセス動作

1) 読み込み要求

サイトiにデータアイテムjの読み込み要求が発生すると、次の手順で読み込みを行うものとする。

- ・ロック取得: 複製保持の全サイトへロック要求を行い、すべての取得を確認する。(AND型)
- ・読み込み: 複製保持の全サイトへ読み込み要求を行い、複製保持サイトは読み込み処理を行い、要求サイトにデータを返送する。要求サイトは最初に到着したデータを使用する。(OR型)
- ・ロック解放: 複製保持の全サイトへロック解放を要求し、すべての解放を確認する。(AND型)

2) 更新要求

サイトiにデータアイテムjの更新要求が発生すると、次の手順で更新を行うものとする。

- ・ロック取得: 複製保持の全サイトへロック要求を行い、すべての取得を確認する。(AND型)
- ・読み込み: 複製保持の全サイトへ読み込み要求を行い、複製保持サイトは読み込み処理を行い、要求サイトにデータを返送する。要求サイトは最初に到着したデータを使用する。(OR型)
- ・アプリケーション: 新しいデータの内容を作成。
- ・書き込み: 複製保持の全サイトへ書き込み要求を行い、複製保持サイトは書き込み処理を行い、要求サイトに完了を報告する。要求サイトではすべてが完了するまで待つ。(AND型)
- ・ロック解放: 複製保持の全サイトへロック解除を要求し、すべての解放を確認する。(AND型)

(3) 物理的サーバへの負荷の集計

サイトi₀にデータアイテムjのアクセス要求が1回発生すると、アイテムjを保有する複製保持サイトi₁、すなわち物理的サーバに読み込み、更新要求それぞれにつき、次の負荷がかかる。

$$sr_{i_1j}, sr_{i_1j} + sw_{i_1j}$$

従って、物理的サーバとしてのサイトiから見て、各アイテムjごとの負荷はつぎようになる。

- ・読み込み負荷
jの到着率: $X_{ij}\lambda r_{ij}$, 処理時間: sr_{ij}
 - ・更新負荷(読み込み、書き込み)
jの到着率: $X_{ij}\lambda u_{ij}$, 処理時間: $sr_{ij} + sw_{ij}$
- ただし、 $\lambda r_{ij} = \sum_{i=1}^I \lambda r_{ij}$, $\lambda u_{ij} = \sum_{i=1}^I \lambda u_{ij}$

これを、「計算パッケージ1」を用いて、アクセスグループ（データアイテムjの読み込み、データアイテムjの更新）ごとに能力の分割を行う。

その結果を Cr_{ij} 、 Cu_{ij} とする。

これを応答時間計算のときの論理的サーバの能力として、処理時間を再設定する。

$$srr_{ij}' = sr_{ij} / Cr_{ij}$$

$$sru_{ij}' = sr_{ij} / Cu_{ij}$$

$$swu_{ij}' = sw_{ij} / Cu_{ij}$$

(4) 通信システムの負荷の集計

データアイテムjについてネットワークの負荷を求める。jについての複製の数を X_j とする。

ただし、 $X_j = \sum_i X_{ij}$ 。今回は要求サイトに複製の有無を考えない。

・読み込み負荷

ネットへの到着率 $\lambda nr_j = \lambda r_j X_j$

ネットの処理時間 $s nr_j = (5 d1 + d2) X_j$

処理時間の分散は $(5 d1^2 + d2^2) X_j$

・更新負荷

ネットへの到着率 $\lambda nu_j = \lambda u_j X_j$

ネットの処理時間 $s nu_j = (6 d1 + 2 d2) X_j$

処理時間の分散は $(6 d1^2 + 2 d2^2) X_j$

上記の負荷を、「計算パッケージ1」を用いて、アクセスグループ（データアイテムjの読み込み、データアイテムjの更新）ごとにネットワークの能力を分割する。

(5) 待ち行列モデル

図7から図10に待ち行列モデルを示す。

次の手順によって、データアイテムjのサイトiからの読み込み、更新の応答時間を求める。

(1) 共有リソースのアクセスグループへの能力分割

前述の方法でアクセスグループ（アイテムjの読み込み、アイテムjの書き込み）ごとに共有リソース（ファイルサーバとしてのサイト、ネットワーク）の能力を分割する。

(2) アクセスグループごとに平均処理時間、分散を求める

1) ロック要求

図8により、直列数は2、並列数は複製数（最大3）のAND型並列処理であり、この合成されたサーバの処理時間と分散を「計算パッケージ2」により求め、この処理時間と分散からM/G/1のポラチェックヒンチンの公式から、この部分の応答時間 Tqr_j 、 Tqu_j が求められる（読み込み、更新）。

2) 読み込み処理

ロック以降の処理時間と分散を求める。図9により、3直列、複製数OR型並列部分（平均処理時間と分散を $Sr1_j$ 、 $\sigma r1_j^2$ とする）及び2直列、複製数AND型並列部分（平均処理時間と分散を

$Sr2_j$ 、 $\sigma r2_j^2$ とする）の直列であり、それぞれ「計算パッケージ2」にて求める。読み込み処理全体の平均処理時間及び分散は二つの部分の和である。また到着率は $\lambda r_j = \sum \lambda r_{ij}$ である。

$$Sr_j = Sr1_j + Sr2_j$$

$$\sigma r_j^2 = \sigma r1_j^2 + \sigma r2_j^2$$

3) 更新処理

ロック以降の処理時間と分散を求める。図10により、3直列、複製数OR型並列部分（平均処理時間と分散を $Su1_j$ 、 $\sigma u1_j^2$ とする）、3直列、複製数AND型並列部分（平均処理時間と分散を $Su2_j$ 、 $\sigma u2_j^2$ とする）及び2直列、複製数AND型並列部分（平均処理時間と分散を $Su3_j$ 、 $\sigma u3_j^2$ とする）の直列であり、それぞれ「計算パッケージ2」により求める。更新処理全体の平均処理時間及び分散は三つの部分の和である。また到着率は $\lambda u_j = \sum \lambda u_{ij}$ である。

$$Su_j = Su1_j + Su2_j + Su3_j$$

$$\sigma u_j^2 = \sigma u1_j^2 + \sigma u2_j^2 + \sigma u3_j^2$$

(3) アクセスグループごとに応答時間を求める

同一データアイテムのアクセスがポアソン過程で到着することにより図7に示すロック待ち行列が生じる。読み込みと更新が同一の論理的リソースすなわちロックを使用するので、待ち時間 T_w は共通で両者の合成された、平均処理時間、分散を $M/G/1$ のポラチェックヒンチンの公式に適用して求めることができる。

$$T_w = \rho_j S_j (1 + \sigma_j^2) / (2(1 - \rho_j S_j))$$

ただし、

$$\rho_j = \lambda r_j Sr_j + \lambda u_j Su_j$$

$$S_j = (\lambda r_j Sr_j + \lambda u_j Su_j) / (\lambda r_j + \lambda u_j)$$

$$\sigma_j^2 = (\lambda r_j \sigma r_j^2 + \lambda u_j \sigma u_j^2) / (\lambda r_j + \lambda u_j)$$

従って、データアイテムjの読み込みの応答時間 Tqr_j 、更新の応答時間 Tqu_j はそれぞれ、

$$Tqr_j = Tqr1 + T_w + Sr_j$$

$$Tqu_j = Tqu1 + T_w + Su_j$$

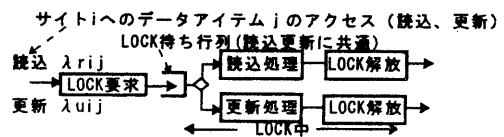


図7 全体の処理

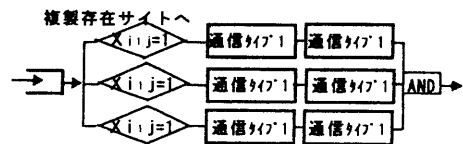


図8 ロック要求の処理

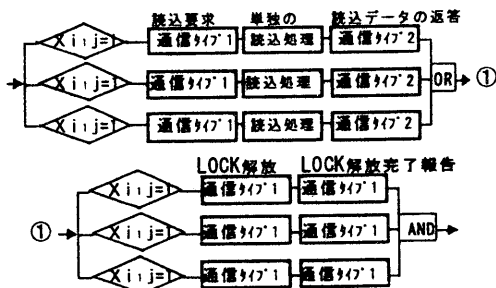


図9 読込処理、ロック解放

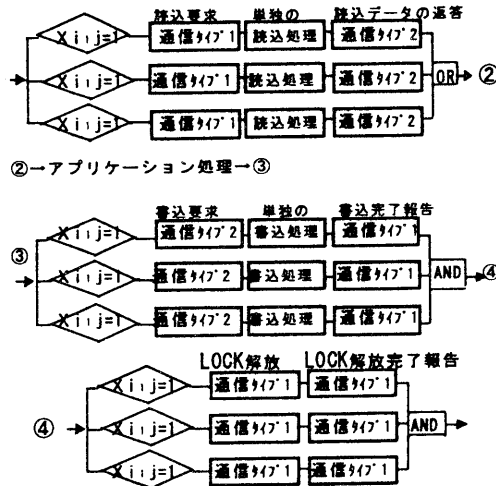


図10 更新処理、ロック解放

(4) 計算例

上記の計算方式をプログラムに作成した。表3に計算例を示す。サイト数=3、データアイテム数=6、 $d_1=0.01$ 、 $d_2=0.1$ 、 $X_{ij}=1$ (すべての複製を持つ)、 $s_{r_{ij}}=1$ 、 $s_{w_{ij}}=1$ において、 $\lambda=\lambda_{r_j}=\lambda_{u_j}$ を変化させて計算した。

表3 計算例

λ	0.001	0.01	0.02	0.04	0.045
T_{qr_j}	2.32	2.79	3.61	9.95	21.14
T_{qu_j}	5.68	6.21	7.13	14.01	25.58

5. データ更新アルゴリズム案

以上検討してきたようにロックを行わず、また全複製へのアクセスを行わなければ、当然ながら応答性能は向上する。筆者らは、先に制限付楽観制御方式 (MOCC) と呼ぶ方式を提案した5)。この方式は、各サイト毎にデータ更新の限界値を設定し、その範囲内で更新を許容する方式である。内容に関わらず楽観的に更新を行う通常の楽観制御方式に比べて、更新動作は制限付きであり、

modest (慎み深い) であるので、

Modestly Optimistic Concurrency Control(MOCC) (制限付楽観的制御方式)と仮称した。

通常のアクセス時は、上限値の範囲内で更新を行ない、夜間等のアクセスの閑散時に全体の一貫性回復するという方法である。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。

ただし、この方式はデータの意味を利用しているので、適用範囲に制限がある。この方式が有効である適用分野は預金口座残高、在庫量、座席予約用座席数等のように共通の量を増減するような分野である。

また、筆者らは仕様書作成のようにデータの内容が階層構造に従うような場合の、複製間の一貫性についても検討を行った[10]。

6. おわりに

(1) ロックを用いて複数の複製をアクセスする場合の応答時間性能についての近似的な解析的評価方法を示した。今回は、読込み、更新の両者を含む解析を行った。また、通信時間も考慮した。

(2) 全複製ロックおよびアクセスに起因する性能低下を回避する更新方式を検討してきた。

今後の課題として下記を検討する予定である。

- ・通信時間、読込みを含めたシミュレーションを行い、解析結果との比較を行う。
- ・性能評価と複製配置の関連の検討。
- ・複製管理方式の検討。

7. 参考文献

[1] Kleinrock L., "On the Modeling and Analysis of Computer Network," Proceedings of IEEE, V.81, No.8, '93
 [2] Singhal M., "Update Transport: A New Technique for Update Synchronization in Replicated Database Systems" IEEE Transactions on Soft.Eng., V.16, No.12 '90
 [3] Hellerstein J.L., "Constructin Quantitative Models Using Monotone Relationships," IEEE Trans. on Knowledge & D.E. Vol.7, No.2, 1995
 [4] Salvatore T.M. et al., "Allocating Data and Operations to Nodes in Distributed Database Design," IEEE Trans. on Knowledge and Data Eng., Vol.7, No.2, '95
 [5] 宮西, 中村, 渡辺, 水野, "分散システムにおけるデータ更新性能解析と更新アルゴリズムの提案", 情報処理学会DPS研, 95-DPS-69(1995)
 [6] Kung H.T. et al., "On Optimistic Methods for Concurrency Control" ACM Trans. on DB Sys. V.6, No.2 '81
 [7] Thomas R.H., "A Majority Consensus Approach to Concurrency Control for Multiple Copy Databases" ACM Trans. on DB Sys. V.4, No.2 '79
 [8] Reading in DISTRIBUTED COMPUTING SYSTEMS Chapter 9, 10, 11 IEEE Comp. Soc. Press, 1994
 [9] 森村, 大前 "応用待ち行列理論" 日科技連 '75
 [10] 宮西, 中村, 佐藤, 水野, "分散システムにおける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, 95-DPS-71(1995)