数理解析研究所講究録
第666巻 1988年 255-264

先読みスケジューラによる分散型データベースシステムの並行処理制御

京都大学工学部　武田 真人 (Makoto TAKEDA)
京都大学工学部　茨木 俊秀 (Toshihide IBARAKI)
Simon Fraser Univ.　亀田 恒彦 (Tiko KAMEDA)

あらまし

各サイトが多版方式の先読みスケジューラを用いた分散型データベースシステムにおいて、すべてのトランザクションが、自己のサイトに対しては読み書きを行なうが、他サイトに対しては読み取り操作のみを行なうという制約をみたすとき、各サイトが独立にそのステップを実行できるような並行処理制御法が可能であることを示す。このようにしても、先読みスケジューラの特徴である、デッドロックがなく、後退復帰を必要としないという性質がそのまま成立する。

1. まえがき

分散型データベースシステムの並行処理制御法(concurrency control)において、2相ロック方式や時刻印方式が広く知られているが[1]、これらでは、実行済み操作の後退復帰(rollback)やシステムのデッドロックを避けることができない。システムが複雑となる分散型データベースシステムでは、これらは並行処理の効率を下げる大きな要因となる。この点を改善するため、予め各トランザクションの読み取り集合、書き込み集合を宣言することによって、システムの通常処理の実行中では後退復帰も必要なくデッドロックも生じない先読みスケジューラ(cautious scheduler)[2],[5]を用いた方式も提案されている[3]。しかし、この方式では各トランザクションの実行の前に、他のサイトへの通信を行って、その登録を行っておかねばならず、通信量の増大と処理時刻の遅延が避けられない。そこで、本論文では、一つのデータ項目は一つのサイトにのみ存在するという制限を設け、さらに各トランザクションは自己のサイトについては読み書きの両操作を行うが、他のサイトに対しては読み取り操作のみを行うと仮定する。このとき、各サイトがMCS(MWW)と呼ばれる多版先読みスケジューラを用いるならば、トランザクションの実行は他のサイトの状態とは独立に、自己サイトの先読みスケジューラのみに基づいて実行できることを示す。これはいわゆるBakunin

データベースネットワーク[6]と目的を一にするものであるが、トランザクションに加えられる制約および制御方式自体は相当異なる。このようなシステムではサイト間の通信量も減少し、各サイトにおけるトランザクションの制御も容易であるので、実用上意味があると考えられる。

本論文の後半では、トランザクションをさらに制限することで、長時間を要すると思われる他のサイトへの読み取りステップのために自己のサイトへのステップが待つことがないような並行処理方式を提案する。

2. データベースシステム

データベースシステムはデータ項目の集合$D=\{X_1,\cdots,X_m\}$と、データベース上で実行されるトランザクションの集合$\mathbb{T}=\{T_0,T_1,\cdots,T_n,T_f\}$とから成る。ただし、各トランザクション$T_i$($1\leqq i\leqq n$)は、データベースからデータ項目$X\in S$(ただし、$S\subset D$)を読み取る読み取りステップ($R_i[S]$と記す)と、$X\in S$の内容を更新する書き込みステップ($W_i[S]$と記す)の二種類のステップの有限個からなる系列である。$X\in S$に対する$R_i[X]$および$W_i[X]$をそれぞれ読み取り操作、書き込み操作という。$T_0=W_0[D]$、$T_f=R_f[D]$はそれぞれ、初期および最終トランザクションで、すべてのデータ項目の初期値を準備し、さらに最終値を読み取るために仮想的に導入されたものである。$\mathbb{T}$上の系列sとは、$\mathbb{T}$内のトランザクションのすべてのステップを左から右に一列に並べたもので、$W_0[D]$で始まり、$R_f[D]$で終わり、かつ得られたステップの順序が個々のT_i内のステップの順序に矛盾しないものをいう。

本論文で扱う多版方式では、系列sにおいて、各$R_j[X]$はそれに先行する(すなわち、$R_j[X]$の左に位置する)任意の$W_i[X]$の一つによって書かれたデータXを読む。従って、どの書き込み操作から読むかを定める必要があり、これを版割当て(version assignment)と呼び、Iと記す。すなわち、$R_j[X]$が$W_i[X]$から読むとき、$I(R_j[X])=W_i[X]$と書く。とくに、$W_i[X]$が$R_j[X]$に先行する最も新しい書き込み操作であるとき、その版割当てを標準版割当てと呼び、I^*と表わす。系列sとその版割当てIの組$<s, I>$を$\mathbb{T}$のスケジュールと呼ぶ。スケジュール$<s, I>$が直列(serial)であるとは、$I=I^*$であり、かつsにおいて各トランザクション内のすべてのステップが他のトランザクションのステップに割り込まれることなく連続して置かれているときをいう。同じステップ集合に対するスケジュール$<s, I>$と$<s', I'>$において、$I=I'$ならば、両者は等価であるといい、$<s, I>\equiv<s', I'>$と記す。スケジュール$<s, I>$が直列可能(serializable)とは、$<s, I>\equiv<s', I'>$を満たす直列スケジュール$<s', I'>$が存在することである。すべての直列可能なスケジュールの集合をISRと書く。また、$<s, I>\in$ISRなるIをもつ系列sの集合をSRと書く。直列可能性はデータの一貫性を保証するもの

であって、スケジューラの役目は、各トランザクションから発せられたステップの実行要求を直列可能スケジュールに整列することにある。

[定義 2.1][4]あるスケジュール＜ｓ，Ｉ＞に対してTIOグラフ、TIO(ｓ，Ｉ)＝(Ｖ，Ｅ)、を次のように定義する。

Ｖ：節点の集合で、ｓのトランザクションに対応する節点、および後述のダミー節点からなる。

Ｅ：枝の集合で、T_jがT_iの書いたＸの値を読むとき、有向枝ｅ＝(T_i，T_j)∈Ｅを設け、ラベルＸを付す。これらを読み取り枝と呼ぶ。あるT_iが書いたＸの値がどのトランザクションによっても読まれなかったとき、$W_i[X]$は不用(useless)であるといい、ダミー節点T'_iを作り、ラベルＸの有向枝ｅ'＝(T_i，T'_i)∈Ｅを付す。トランザクションと節点は一対一に対応するので、以下ではTIOグラフの節点をトランザクションと呼ぶことがある。□

[定義 2.2][4]TIO(ｓ，Ｉ)がDITS(Disjoint Interval Topological Sort)をもつとは、Ｖの要素間に次の二つの条件を満たすように全順序≪を定め得ることをいう。

１.TIO(ｓ，Ｉ)でT_iからT_jへ道があるとき、$T_i \ll T_j$。

２．排除規則：同じラベルＸをもつ枝ｅ＝(T_h，T_i)とｅ'＝(T_j，T_k)が$T_h \neq T_j$を満たせば、$T_i \ll T_j$または$T_k \ll T_h$のいずれかが成り立つ。□

一般に与えられたTIOグラフがDITSをもつかどうかを判定する問題はＮＰ完全である[7]，[4]。そこでその判定を多項式時間で可能とするためDITSに下記の制限を付し、その計算の複雑さを軽減することが試みられている[4]。

ｗｗ制約：ｓ内で、あるＸに対し$W_i[X]$が$W_j[X]$に先行するならば、DITSにおいてT_iをT_jのまえに置く。

制約ｗｗの下で直列可能なスケジュールの集合をIWW、さらに、＜ｓ，Ｉ＞∈IWWなる版割当てＩをもつような系列ｓの集合をMWWと記す。TIO_{ww}(ｓ，Ｉ)とは、TIO(ｓ，Ｉ)にｗｗにもとづく順序制約の枝を加えたグラフである。すなわち、ｗｗ制約を満たすｉとｊに対し、有向枝(T_i，T_j)を加え、ｗｗ枝と呼ぶ。□

[定義 2.3][4]定義 2.2の２の条件をみたす枝ｅおよびｅ'において(T_iとT_kはダミー節点であってもよい)T_hからT_kへの道が存在するとき、枝ｅ"＝(T_i，T_j)を付し排除枝と呼ぶ。TIO_{ww}(ｓ，Ｉ)にすべての排除枝を加えたグラフを排除閉包(exclusion closure)と呼び、TIO^*_{ww}(ｓ，Ｉ)で表わす。□

[定理 2.1][5]系列ｓ∈MWWの必要十分条件はTIO_{ww}(ｓ，Ｉ)がDITSをもつことである。□

[定理 2.2][4]TIO_{ww}(ｓ，Ｉ)がDITSをもつための必要十分条件はTIO^*_{ww}(ｓ，Ｉ)が閉路をもたないことである。□

3．先読みスケジューラMCS(MWW)

ここでは、本並行処理制御方式の基本となる多版方式の先読みスケジューラMCS(MWW)について説明する。

各トランザクションTは、実行に先立ち、読み取り集合(Tが読み取るデータ項目の集合)と書き込み集合(Tが書き込むデータ項目の集合)をスケジューラに宣言し登録を行う。登録が済むとTはステップの実行要求をスケジューラに発する。MCS(MWW)は最終的に＜s，I＞∈IWWなるスケジュールが得られるようにそれらの順序と版割当てを定めて出力する(実行を許可する)。各ステップの実行可能性の判定を完成テストと呼ぶ。

MWWの完成テストは以下のように行う。

＜P，I＞をある時点ですでに出力された部分スケジュール、qを現在判定すべきステップ、PENDを

{既にスケジューラに最初のステップを送っているトランザクションが宣言したステップ}−{Pq中のステップ}

とする。

[定義 3.1][5] ATIOグラフ、ATIO(＜P，I＞，q，PEND)、はTIOグラフと同様にして以下のように定義される。すでにPqあるいはPENDにその最初のステップが含まれているようなトランザクションT_iを節点とし、＜P，I＞部分においてT_jがX をT_iから読み取っているならば、ラベルXの有向枝(T_i，T_j)を付す。＜P，I＞中のダミー節点についても同様に定める。また、各$W_i[X]$∈PENDはダミー節点T'_iへのラベルXの有向枝(T_i，T'_i)で示され、$R_i[X]$∈PENDは始点をもたないラベルXの枝(−，T_i)によって示される。これらを処理待ち枝と呼ぶ。□

ATIO(＜P，I＞，q，PEND)に以下の有向枝を加え、さらにその排除閉包をとったものを$ATIO^*_{ww}$(＜P，I＞，q，PEND)と記す。

(i) Pqにおいて$W_i[X]$が$W_j[X]$に先行するときww枝(T_i，T_j)を付す。

(ii) $W_k[X]$をPqにおけるXへの最後の書き込み操作とする。このとき、すべての$W_j[X]$∈PENDに対しww枝(T_k，T_j)を付す。

(iii) q＝$R_j[S]$ のとき、すべてのX∈Sへの処理待ち書き込みステップをもつトランザクションT_kへrw枝 (T_j，T_k) を付す。

[定理 3.1][5] MCS(MWW)の完成テストに合格する必要十分条件は$ATIO^*_{ww}$(＜P，I＞，q，PEND)が有向閉路をもたないこと。□

さらに、MCS(MWW)によって出力されるスケジュールについて次の定理が成り立つ。

[定理 3.2][5] MCS(MWW)は入力されたｓ'に対し(必要ならばステップの順序を変更し)、必ずIWWに属すスケジュール＜ｓ，Ｉ＞を出力する。このとき、ｓにおけるステップの順序は各トランザクション内のステップの順序に矛盾しない。□

すなわち、これらの定理よりMCS(MWW)は

（１）システム中でデッドロックを生じない

（２）ステップの後退復帰の必要がない

（３）各ステップは必ず有限時間内で実行できる

ことが保証されている[5]。

ところで、各トランザクションはその書き込み集合と読み取り集合を先読みスケジューラへの登録時に予め宣言するが、後になって宣言されていない操作が到着したとする。そのとき、その操作のためにシステムが停止してしまう恐れがある。このような異常が起こることを追加異常(augmentation anomaly)[5]と呼ぶ。MCS(MWW)は、追加異常について次の定理が成り立つ。

[定理 3.3][5] MCS(MWW)においては、予め宣言されていない読み取り操作が到着しても追加異常は起こらない。（書き込み操作についてはこの限りではない。）□

４．分散型データベースシステムモデル

分散型データベースシステム(DDBMS)とは互いに通信回線によって接続されたｎ個のデータベースシステム(以下では個々のデータベースシステムをサイトと呼ぶ。)から成り立っている。本論文ではDDBMSに以下の仮定を置く。

（１）サイトａに存在するデータ項目をD_aとするとき、任意の異なるサイトａ，ｂに対して$D_a \cap D_b = \phi$。すなわち、複数のサイトに存在するようなデータ項目はない。

（２）各トランザクションＴが書き込むデータ項目の集合を$D_w(T)$とすれば、$D_w(T) \subset D_a$なるデータ項目をもつサイトａが必ず一つだけ存在する。すなわち、各トランザクションの書き込み集合はただ一つのサイトのデータ項目にのみ許される。そのようなサイトａをＴのホームサイトと呼ぶ。

（３）各トランザクションは任意のサイトのデータ項目を読み取ることができる。

（４）各サイトは時計をもち、時刻印を発することができる。また、システムが複数のトランザクションに同一の時刻印を与えることはないとする。

（５）各サイトは、他のサイトにどのようなデータ項目があるかを知っている。

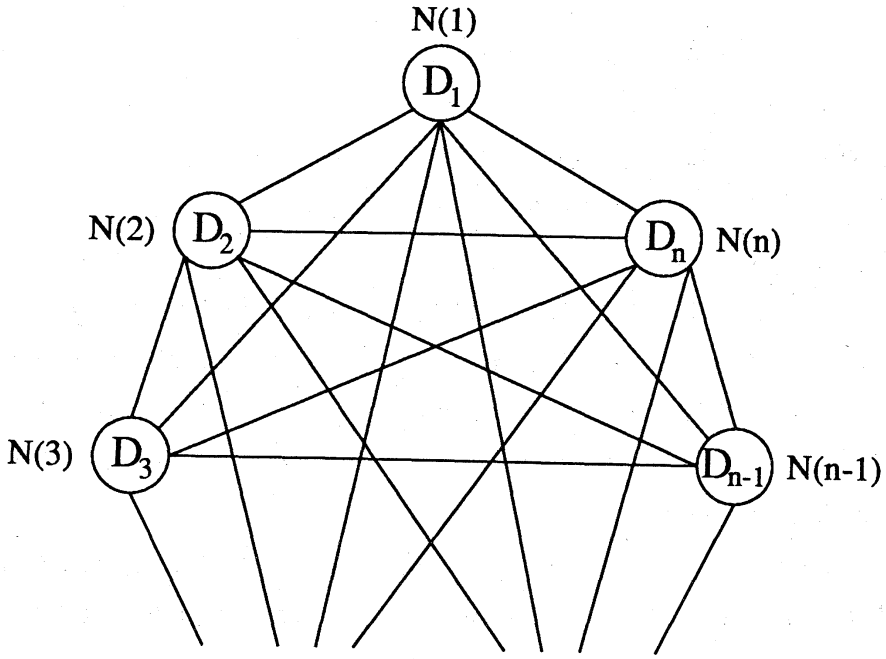
（６）各サイトは、それぞれトランザクション制御装置および先読みスケジューラMCS(MWW)によって独立に制御されている。

（７）各ユーザは次のようにしてトランザクションＴを実行する。

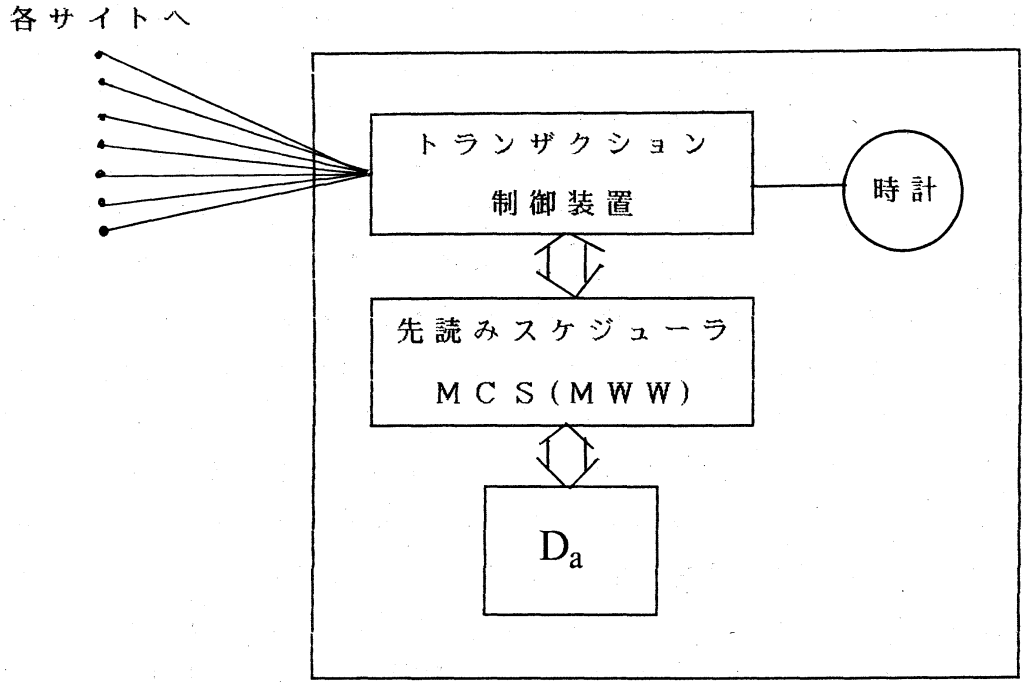
（ｉ）まず、Ｔのホームサイトへその読み取り、書き込み集合とともに登録したの

ち、そのステップの実行要求を出す。

(ii)Tは一つのステップの実行が終了してはじめて次のステップの実行要求を出すことができる。□



分散型データベースシステム



サイトaの構成

あるトランザクションがホームサイト以外のデータ項目を読み取るステップをもつとき、そのトランザクションを全域的(global)トランザクション、また、その読み取りステップを遠方(remote)読み取りステップと呼ぶ。全域的トランザクション以外のトランザクションを局所的(local)トランザクションと呼ぶ。なお、全域的トランザクションのステップにおいて、遠方読み取りステップ以外のステップを局所的(local)ステップという。

5．MCS(MWW)を用いた分散型アルゴリズムとその正当性

各トランザクションTはそのホームサイトaにおいて、トランザクション制御装置および先読みスケジューラMCS(MWW)によって以下のように制御される。

（1）Tが全域的トランザクションであれば、トランザクション制御装置は、Tの到着時に時刻印を付す。(局所的トランザクションには付さない。)その後、Tの宣言した読み取り集合および書き込み集合にもとづいて、TをサイトaのMCS(MWW)に登録する。なお、サイトaのMCS(MWW)の完成テストにおいては、ATIOグラフに上記の時刻印の制約枝を付加して行う。(つまり、ATIOグラフ内のT_iとT_jが時刻印をもち、T_iの時刻印がT_jの時刻印より小さければ、枝(T_i, T_j)を加える。)

（2）全域的トランザクションの局所的ステップおよび局所的トランザクションのステップについてはサイトaのMCS(MWW)によって処理される。

（3）全域的トランザクションの遠方読み取りステップについてはそのトランザクションの時刻印とともにそれが読み取るデータ項目をもつサイトへ送る。

（4）他のサイトbから時刻印tをもった全域的トランザクションT_mの遠方読み取りステップ$R_m[X]$が到着した時、T_mをサイトaのMCS(MWW)に時刻印の制約枝を付加して加え、完成テストにかける。このとき、サイトaの時刻t'がt＞t'をみたすならば、サイトaの時計をtまで進める。□

MCS(MWW)の動作は、そのATIOグラフに全域的トランザクション間の時刻印の制約枝が付加されている点を除けば、集中型の場合[5]と同様である。

[定理 5.1] 各サイトaにおいては、そのサイトでの処理を要求されたすべてのステップを有限時間内で実行できる。 □

[定理 5.2] すべてのトランザクションの処理が終了したとき、各サイトにおいて得られているTIOグラフを次のように重ね合せて、全体のTIOグラフを作る。即ち、同じ全域的トランザクションを表わす節点は一つに重ね合せ、また、全域的トランザクション間の時刻印の制約枝は、異なるホームサイト間であっても加える。(なお、システムの仮定(1)より、新たなｗｗ枝や読み取り枝は加わらない。)このようにして得られるTIOグラフはDITSをもつ。

(証明) 各サイトはMCS(MWW)によって制御されているから、すべてのステップが終了したとき、そのサイトのTIOグラフは定理 3.2よりDITSをもつ。サイトａ(１≦ａ≦ｎ：ｎはサイト数)のDITSをDITSaとする。すべてのDITSaから次にようにしてシステム全体のトランザクションの系列σを作る。まず、すべての全域的トランザクションを時刻印の順に左から右へ一列に並べその系列をτとする。各DITSa中の全域的トランザクションは時刻印の順序に並べられているので、DITSa中の二つの全域的トランザクションT_i、T_j($i \neq j$)が$T_i < T_j$(T_iはT_jよりも左)ならばτ中でも$T_i < T_j$である。サイトａの局所的トランザクションT_mについては、DITSa中でそれがはさまれている全域的トランザクションT_iとT_j(すなわち、DITSa中で$T_i < T_m < T_j$である)の間にτにおいても同様にはめこむ。このとき、他のサイトの局所的トランザクションおよび他サイトをホームサイトとする全域的トランザクションとの位置関係は任意でよい。

こうして得られた、システムのすべてのトランザクションを一列に並べた系列がσである。次にこのσがシステム全体のTIOグラフのDITSになっていることを示す。

（１）各サイトのDITSaの順序はσでも保存されているから、システム全体のTIOグラフの節点をσに従って整列したときすべての枝は左から右に向いている。(システム全体のTIOグラフを作成するとき、新たに時刻印の制約枝がひかれているが、σは全域的トランザクションを時刻印の順に並べているのでこの枝も左から右へ向いている。)

（２）排除規則：各DITSaにおいては排除規則が満たされている。また、各データ項目Ｘは一つのサイトにしか存在しないから、同じラベルＸをもつ枝は一つのサイトのTIOグラフにしか現れていない。これは、システム全体のTIOグラフで排除規則が成り立っていることを意味する。

以上により、σは定義 2.2のDITSの条件(1)、(2)を満たしている。□

[定理 5.3] システム全体のデッドロックは起きない。 □

このように、本システムでは、すべてのトランザクションがデッドロックを起こすことなく処理されるが、全域的トランザクションの遠方読み取り操作の実行を待つ

間、ホームサイトで、局所的トランザクションの実行が遅延されることがある。これは、実用上好ましくない場合も考えられるので、次節では、トランザクションも2ステップモデルに限ることで、局所的トランザクションに高い優先度を与え得ることを述べる。

6．2ステップモデルを用いたシステム

トランザクションT_iが一つの読み取りステップ$R_i[S]$とその後に続く書き込みステップの$W_i[S']$から成るとき2ステップモデルであるといわれる。以下では、すべてのトランザクションが2ステップモデルであるとする。なお、一般的に読み取りステップ$R_i[S]$のSはホームサイトのデータ項目の集合S_1と他のサイトのデータ項目の集合S_2に分割されるが、T_iの処理にあったてはまず遠方読み取りステップ$R_i[S_2]$の実行をまず要求し、それが完全に済んだ後、局所的ステップである自己サイトの読み取りステップ$R_i[S_1]$および書き込みステップ$W_i[S']$の実行を要求するものとする。全域的トランザクションのうちで遠方読み取りステップが既に実行を終えその値をホームサイトにもちかえっているトランザクションを局所化(された)トランザクションと呼ぶ。

このようなトランザクションに対し、各サイトのトランザクション制御装置は次のように制御する。

（1）局所的トランザクションはそのままそのサイトの先読みスケジューラMCS(MWW)に送り読み取り、書き込みステップの実行を進める。

（2）全域的トランザクションTはホームサイト到着時に時刻印を付し、遠方読み取りステップは直ちに関係サイトへ時刻印とともに送る。Tおよびそのホームサイトに対して宣言されたステップはそのままMCS(MWW)に登録し、以後局所的ステップについては、局所的トランザクションと同様に扱う。ただし、ある全域的トランザクションの局所的ステップの実行要求が到着しても、そのトランザクションの時刻印以下の時刻印をもつすべての全域的トランザクションが局所化されるまで先読みスケジューラへは渡さない。

[定理 6.1] 上記の規則を採用したシステムでは、局所的トランザクションのどのステップも、まだ局所化されていない全域的トランザクションによって遅らされることはない。 □

7．むすび

分散型データベースシステムでは、システムが複雑となるために各サイトが独立に並行処理を行い、その結果システム全体の並行処理がなされているようにするには容

易ではない。そこで、本論文ではトランザクションに制限を設けることでこれらを可能にする並行処理制御法を提案した。これは、多版方式の先読みスケジューラMCS(MWW)が追加異常を生じないことを利用したものであって、各サイトのみならず、システム全体としてもデッドロックを生じる可能性はなく、従って、後退復帰も必要ない。

さらに、トランザクションを２ステップモデルに限定すれば、局所的トランザクションがその実行に比較的時間のかかる遠方読み取りステップのために遅らされることはないようにできることも示した。

謝辞　いろいろと熱心にご討論いただいた増山繁助手(京大)に感謝いたします。なお、本論文は一部文部省科学技術研究費によるものである。

文　　献

(1) P.A.Bernstein and N.Goodman, "Concurrency control in distributed database systems", ACM Computing Surveys, 13, 2, pp.185-224(1981).

(2) M. A. Casanova and P. A. Bernstein, "General purpose schedulers for database systems", Acta Informatica, 14, pp.195-220 (1980).

(3) 原嶋、茨木 "先読みスケジューラによる分散型データベースシステムの並行処理制御", 電子情報通信学会論文誌, J70-D, 6, pp.1140-1148(1987).

(4) T. Ibaraki, T. Kameda and T. Minoura,"Serializability with Constraints", ACM Trans.on Database Systems, 12, 3, pp.429-452(1987).

(5) T.Ibaraki, T.Kameda and N.Katoh,"Cautious transaction schedulers for multiversion database systems", LCCR TR86-2, School of Computing Science, Simon Fraser Univ.(1986).

(6) B.Kogan and H.G.Molina,"Update propagation in Bakunin database networks", Proceedings of the Sixth Annual ACM Symposium on Principles of Distributed Computing, Vancouver, B.C., Canada, Aug.10-12, pp.13-26(1987).

(7) C.H.Papadimitriou, "The serializability of concurrent database updates", J.ACM, 26,4, pp.631-653(Oct.1979).