

6H-8

分散データベース・システムにおける 同時実行制御方式の評価

加藤 宣弘 柿元 満

(株) 東芝 総合研究所

1. はじめに

最近、分散データベース・システム（以下分散DBS）が注目を集めている。分散DBSにおいて重要な問題の1つが同時実行制御である。これまで、いくつかの同時実行制御方式が提案されている¹⁾が、その性能予測は難しい。このような理由から、これまでシミュレーションによる性能評価について、いくつか報告されている²⁾。今回、同時実行制御方式の中で最も一般的な2相ロック方式について、単版の場合と2版の場合のシミュレーションによる評価を行ったので、本文で報告する。

2. 同時実行制御方式

評価した同時実行制御方式は、単版2相ロック方式（以下SV2PL）と2版2相ロック方式（以下2V2PL）である。ここでは、SV2PL、2V2PLについて簡単に説明する。

[SV2PL]

SV2PLでは、readロックとwriteロックが存在する。ある2つの操作があるデータ項目に同時にロックをかける場合に、それら2つのロックが、ともにreadロックである場合には共有できるが、そうでない場合には共有できない。後者の場合に、ここでは次のように処理する。

(1)それらの2つの操作が、異なるトランザクションに含まれる場合には、後からロックをかけようとした操作を、ロックをかけることができるまで待たす。(2)それらが、同じトランザクションに含まれる場合（rewrite操作の場合）には、データ項目にかかっているreadロックをwriteロックにコンバージョンする。

デッドロックは待ち時間付きWD（Wait-Die）方式により防止する。通常のWD方式では、ある操作がロックをかけようとして衝突を起こした場合（上記の(1)の場合）には、操作をそのまま待たせるか、拒否するかをすぐに決定するが、待ち時間付きWD方式では、一定の待ち時間の後にその決定を行う。

[2V2PL]¹⁾

2V2PLでは、readロック、writeロックとcertifyロックが存在する。新しくwriteされたデータ項目をコミットするためには、writeロックをcertifyロックにコンバージョンしなければならない。ある2

つの操作があるデータ項目に同時にロックをかけるとき、それら2つのロックが、ともにreadロックである場合、一方がreadロックで他方がwriteロックの場合には共有できる。しかし、そうでない場合には共有できない。

ロックの衝突時の処理、デッドロック防止方式は、SV2PLと同じである。

3. 分散DBSのモデル

分散DBSのモデルは、複数の計算機サイトとそれらを接続するネットワークにより構成される。各サイトの計算機には複数の端末が接続される。

このモデルでは、各サイトにそれぞれ4つの処理モジュールがあり、それぞれ全く独立に処理を行う。モジュールは、お互いにメッセージ通信により情報を交換する。図1に、4つのモジュールとそのメッセージ通信を待行列網として示す。TMはそのサイトで受け付けたトランザクションの処理を管理する。SCはトランザクションの同時実行制御を行う。DMはディスクへのアクセスを行う。CMはサイト間の通信を処理する。

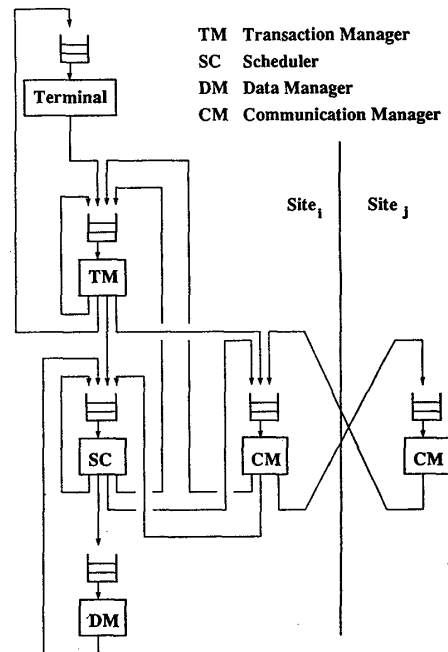


Fig 1 Queuing Network

4. パラメータの設定

シミュレーションのパラメータを表1に示す。表1の数値は、今回のシミュレーションにおいて固定されたパラメータである。ロックの衝突がよく起こるようにパラメータを設定している。

記号	意味	数値
stime	シミュレーション時間	10sec
nterm	サイトあたりの端末数	
nitem	サイトあたりのデータ項目数	10
nopr	トランザクションあたりの操作数	6
prw	rewrite操作の割合	50%
prd	read操作の割合	25%
ttime	端末の思考時間	1sec
rtime	アボート時の再開始時間	1sec
wtime	データ競合時の待ち時間	
tmcost	TMの処理あたりの実行時間	5msec
sccost	SCの処理あたりの実行時間	5msec
ddcost	DMのread、コミット時の実行時間	20msec
dwcost	DMのwrite、アボート時の実行時間	5msec
cmcost	CMの送信、受信実行時間	2.5msec

表1 パラメータ

5. 結果と考察

図2に、wtimeが0、100msecの場合について、単位時間あたりにコミットされるトランザクション数のntermに対する変化を示す。また、wtimeが100msecの場合について、図3に応答時間のntermに対する変化を、図4に単位時間あたりのロックの衝突回数のntermに対する変化を示す。図3では、2V2PLにおけるcertifyロック時の衝突回数も示す。

(1)衝突回数 図4より、2V2PLにおけるロックの衝突回数は、SV2PLにおけるそれより少し多い。しかし、2V2PLの場合には、certifyロック時の衝突回数がかなりあるので、read、writeロック時に限って言えば、2V2PLの場合のロックの衝突回数は、SV2PLの場合より少ない。これは、2V2PLの場合には、readロックとwriteロックの共有を許すからである。

(2)応答時間 図3より、SV2PLにおける応答時間は、2V2PLにおけるそれより小さい。2V2PLの場合にはロックの衝突が多いので、ロック待ちの時間が多くなる。また、2V2PLの場合には、デッドロックの原因の1つであるロック・コンバージョンが多いので、トランザクションのアボート、再開始が多くなる。さらに、2V2PLの場合には、certifyロックのためにサイト間通信を行うので、SV2PLに比べて通信回数が多くなる。

(3)スループット 図2より、SV2PLにおけるコミットされるトランザクション数は、2V2PLにおけるそれより多い。また、図2より、wtimeが100msecであるの場合には、0msecである場合に比べて、コミットされるトランザクション数が多くなる。ロックの衝突が多い状況でも、デッドロックをできるだけ早く防止するより、デッドロックの防止を少し遅らせて、無駄なアボート、再開始を行わない方がよいことを示している。

6. おわりに

分散DBSにおける同時実行制御方式のシミュレーションによる評価を行った。今回のモデルでは、2V2PLは、SV2PLに比べて、性能が劣ることがわかった。

参考文献

- 1) P.A. Bernstein, et al. "Concurrency Control and Recovery in Database Systems" ADDISON-WESLEY PUBLISHING COMPANY, 1987
- 2) M.J. Carey, et al. "Distributed Concurrency Control Performance: A Study of Algorithms, Distribution, and Replication" Proc. of the 14th VLDB Conf., 1988

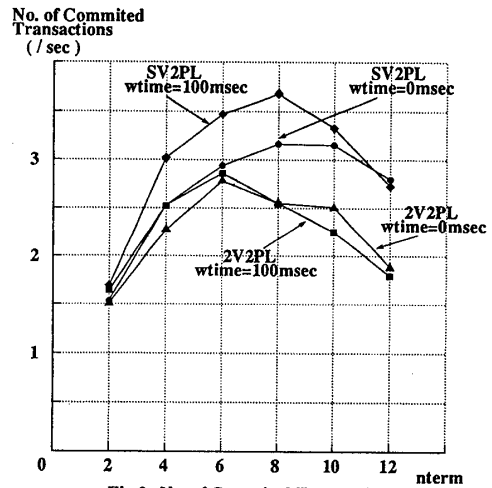


Fig 2 No. of Committed Transactions

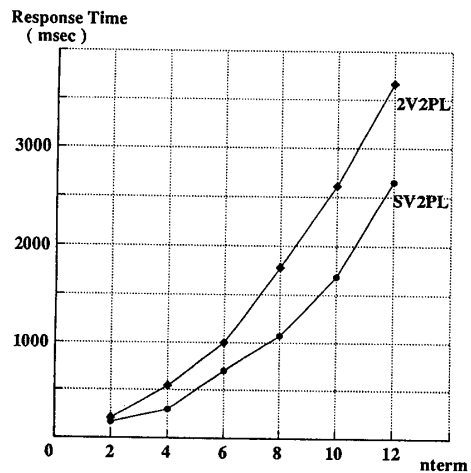


Fig 3 Response Time

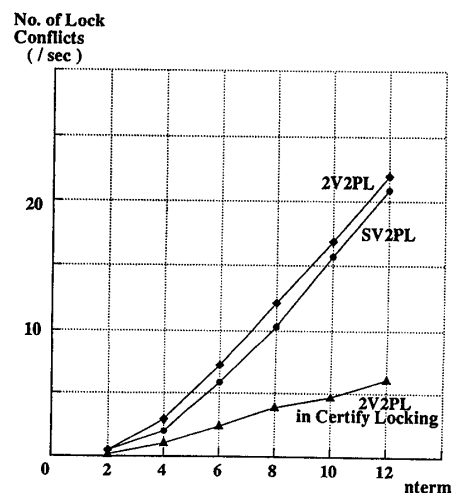


Fig 4 No. of Lock Conflicts