分散ＤＢＭＳ・Ｄｒｅａｍの トランザクション管理とユーザ管理

3H－6

小寺誠、荻野正樹、坂本明史、疋田定幸

（沖電気工業株式会社）

1　はじめに

本稿では、分散データベース管理システム Dream [坂斎88]のトランザクション管理、および利用者管理の機能概要と実現方式を報告する。

2　ﾄﾗﾝｻﾞｸｼｮﾝと利用者管理の課題

分散データベースシステムでの複数サイトの更新機能は、利用者のアプリケーションがデータの配置を意識することなくトランザクションを記述できるために必須である。一方で、複数サイト更新のメカニズムは局所的に動くトランザクションの性能を損ねるものであってはならない。なぜなら、分散システムの目的を考えた場合、各サイトのデータの大半は局所的に処理されるであろうからである。さらに、システム障害に十分に対処するあまり、大部分の正常なトランザクションの性能を損ねることは避けなければならない。

また、複数サイトがシステムの構成要素となるため、分散データベースでの利用者管理には、注意深い設計が要求される。図１のように、あるサイトの利用者 sakamoto と別のサイトの同一名の利用者 sakamoto が同一人物でない場合もある。また sakamoto と saka のように別の名前であっても同一の人物である場合もある。

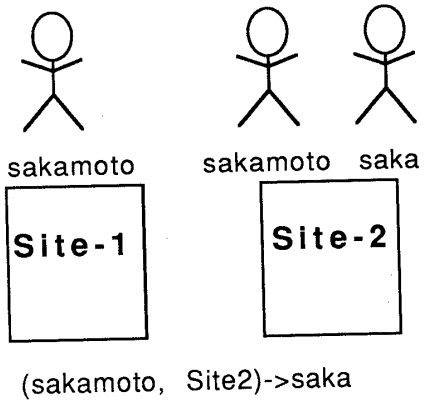

図１　分散データベース環境での利用者

Transaction and User Management in DREAM.
by Makoto KOTERA, Masaki OGINO, Akifumi SAKAMOTO
and
Sadayuki HIKITA

Oki Electric Industry Co., Ldt.

3　機能概要

3．１　トランザクション管理

Dream では２相コミットメカニズムを用いてトランザクションの原子性を保証する。２相コミットは障害発生時にコミット／ロールバックできないサイトが発生する可能性があるという欠点がある。しかし、これを回避するために相を増やすアプローチ [SKEE81] は、正常動作する大部分のトランザクションにとって不利益となるため得策ではない。 Dream では、コミット／ロールバックの判断が不可能になるような障害は、その発生の頻度が極めて低いとの前提に立って２相コミットを採用している。

トランザクションは、実際には各サイトで代理実行されるサブトランザクションの集まりによって実現される。各サブトランザクションはデータの検索、更新を行う。また、トランザクション発生サイトにはトランザクションのコミット時にこれらサブトランザクションを２相コミットで制御するためのコーディネータがある。分散システムにおいても、実際のトランザクションの大きな部分が１サイトに閉じて行われる。 Dream のトランザクション管理においては、トランザクションの処理状況によって、以下のようなコミットメントの最適化を行う。不必要な通信による、各サイトでのトランザクションの実行の遅延を、できる限り小さなものとするためである。

コーディネータは、実際に更新を行ったサブトランザクションを管理している。コミット時には、サブトランザクションの更新の状態に応じてコミットメントプロトコルの選択を行う。図２に示すように、実際に２相コミットプロトコルに従って通信が行われるのは、２サイト以上のサブトランザクションで更新が行われた場合に限られる。

もしどのサブトランザクションも更新を行っていなければトランザクションは read only transactionである。この場合には全サブトランザクションをただちに終了させてよい。

また１サイトだけが更新されている場合、他サイトとの矛盾は発生しないので２相コミットの必要はない。トランザクションのコミットの正否は、更新を行ったサブトランザクションのコミットの正否である。

２サイトが更新されている場合でも、自サイトと他サイト１サイトの更新は例外的に扱うことができる。まず、自サイトを２相コミットの第１相の終わった状態、いわゆるセキュアな状態にする。これが成功したら他サイトをコミットする。他サイトのコミットのせいひにより、

自サイトをコミットするかの判断を行うのである

更新サブトランザクション	コミットメントプロトコル
なし	1相
1サイト	1相
自サイト＋1他サイト	2相*
2他サイト以上	2相

* 通信は1相（本文参照）

図2 Dreamのコミットメントプロトコル

3．2 利用者の管理

Dream では利用者管理は完全にローカルな処理として行なわれる。すなわちあるサイトに関する利用者の登録や認証は必ずそのサイトで行なわれ他サイトの管理者や他サイトのDream によって行なわれることはない。これにより基本的に各サイトの利用者はたとえ同一名であっても異なる利用者を表わすことになる。

加えて、同一利用者が他サイトに利用者登録を持つ場合を管理する。各サイトでは他のサイトへアクセスするとき各個人がどの利用者名によってアクセスすべきかをデータディレクトリとしてとして管理する。例えばサイト2にも利用者登録 saka をもつサイト1の利用者 sakamoto がサイト2にもアクセスする場合、サイト1のデータディレクトリにサイト2では利用者 saka としてアクセスできることを登録しておく。利用者 sakamoto の利用中にサイト1からサイト2へのアクセスが発生した場合、Dream は起動のパラメタとして利用者名sakaおよびパスワードを送り、そのサイトでのアクセスの権限をsakaに設定する。

4 処理構成

[坂荻90]で報告したDream システム全体のアーキテクチャの内、トランザクションマネージャとサイトマネージャは、図3に示すような部分に位置づけられる。トランザクションマネージャは、上記のようなコミットメントの最適化、サイトマネージャは分散環境での利用者管理を、それぞれ実現している。

4．1 トランザクションマネージャ

クライアント側のトランザクションマネージャは start_transaction, commit_transaction, abort_transaction の利用者トランザクションを制御する制御プリミティブを提供する。また実際に更新の行われたサイトを記録するためのプリミティブも持つ。利用者からの commit_transacionが発行されると、コミットメントプロトコルの選択を行いコミットメント制御を行う。

サーバ側のトランザクションマネージャはサブトランザクションに関する制御を行う。start, commit, abort のプリミティブに加えて、2相コミットに対応する secure プリミティブを持つ。これらの機能は REAM の機能として予め実現されており、実際の処理はREAMの内部で行われる。

コミット時のロギングはクライアント側およびサーバ側双方のトランザクションマネージャで行われる。

4．2 サイトマネージャ

各マネージャは他のサイトへアクセスする必要が起きると、このマネージャを呼び出す。ここでは要求に従ってサーバの起動／停止を行う。サーバの起動時には、起動に必要な利用者名、パスワードをデータディレクトリより検索しこれを用いる。

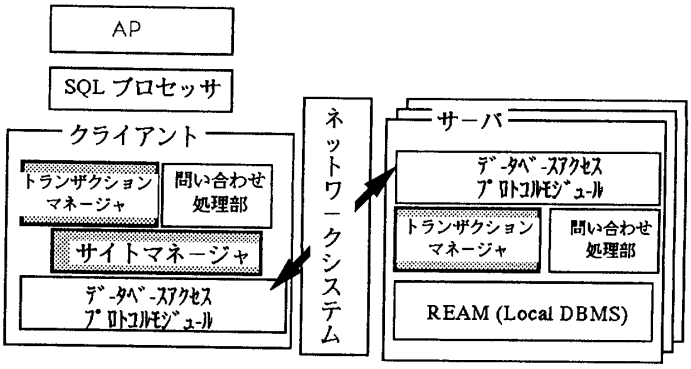

図3 トランザクションマネージャとサイトマネージャの位置づけ

5 おわりに

本稿では、分散データベースシステム Dream に関して、分散処理を考慮したトランザクション管理と利用者管理に関する報告を行った。

参考文献

[坂斎88] 坂本、斎藤、正田、“ＤＲＥＡＭシステムについて”、情報処理学会オペレーティングシステム・データベースシステム合同研究会、９月、1989年

[SKEE81] Skeen, D., "Nonblocking Commit Control", In Proc. of ACM-SIGMOD81, pp133-142, 1981

[坂荻90] 坂本、荻野、小寺、正田、“分散ＤＢＭＳＤｒｅａｍの概要”、情報処理学会第４０回全国大会、３月、1990年