# ASPを用いた異機種DBMSアクセスの同期方式*

6K－5

菊地 宏　三摩 竜治　井上 幸美**
立命館大学大学院理工学研究科 ***

## 1　はじめに

ネットワーク化の進展に伴って、イントラネットからエクストラネットへと情報共有の枠は広がっている。イントラネットでの情報共有の場合、データベースは同一ベンダーが提供するDBMSを使用して分散データベース環境を構築している場合が多いのでシステム開発者は特に意識することなくデータベース間のデータ操作を行うことができる。

しかしエクストラネットを考えた場合、各企業同士が同じベンダーの提供するDBMSを使用しているとは必ずしも限らない。従って分散データベース環境で重要な概念となる「位置」、「分割」、「移動」に対する透過性は保障されないわけである。また更新系の処理（commit、rollback）も保障されない。この問題を解消するために、本論文では、異機種データベース間の同期を図ることを目的としてASPを使用した異機種DBMSの透過的なアクセス法について述べる。

## 2　システムの概要

図1に本研究で構築したシステム構成を示す。

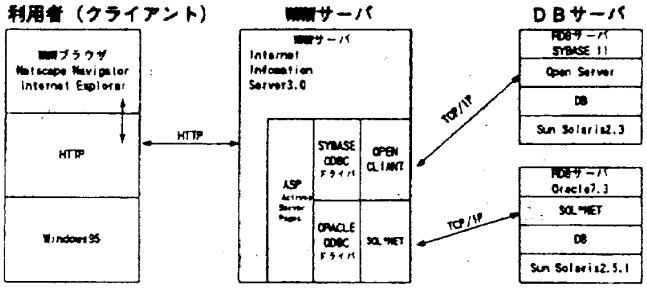


図1　システム構成図

図1よりASP（ActiveServerPages）はODBCを経由してデータベースとの連携をとることができるため、サーバ側に各社ベンダーの提供するODBCドライバと通信ミドルウェア（本システムではTCP/IPプロトコルアダプタ）がインストールされていれば、異機種DBMS間のアクセスを1つのプログラムで実装することが可能である。この特性を活かすことによって複数のデータベースにまたがる検索や、特定の更新処理までが可能になる。しかし異機種間データベースの同期を考えたときデッドロックやダーティ読み出しなどに代表される同時実行制御に関する問題が生じる。

複数のトランザクションが矛盾なく実行されるには、同じデータに対するトランザクションの結果が互いの結果に影響を与えてはいけない。

上述の機能を保証するために本研究では以下の機能をASPファイル上で実装した。

1）　コミットメント制御機能
2）　同時実行制御機能

それぞれの機能について以下に述べる。なお本システムでは作成されたASPファイルは図1のWWWサーバ上に保存されるのでデータベースに対するトランザクションはWWWサーバを介して発生することになる。

## 3　システムの機能

### 3．1　コミットメント制御機能

トランザクションの原子性、およびデータベースの一貫性を守るため、本システムではもっとも一般的に用いられている原子コミットプロトコルである、二層コミットプロトコルを用いてプログラムを作成した。この機能はASPのサーバオブジェクトADO（ActiveX Data Object）のメソッドである、CommitTransとRollbackTransを使用して実装し

---

* A Design of a Synchronized System for Different DBMS Using ActiveServerPages

** Hiroshi Kikuchi. Ryuji Samma. Yukiyoshi Inoue.
*** Ritsumeikan University 1-1-1Nojihigashi,Kusatsu,Shiga 5258857 ,Japan

た。

### 3．2　同時実行制御機能

同時に実行されるトランザクションが生み出す矛盾やデッドロックを回避するために次の2つの処理をプログラムに組み込んだ。

1　同データへのアクセスはロックする

2　トランザクションの実行順序は同じにする

以上の機能を実現するにはロックのタイミングが問題になる。排他ロックを用いる場合アクセスされるデータの性質によってロックのタイミングを変えればパフォーマンスの向上が期待できる。以下に実装した2つのロック方式を示す。

1）共有的ロック

更新をかける段階でロックをかけ、衝突があった場合にトランザクションはロールバックされる。ASP のレコードソースのロックタイプである adLockOptimistic を使用して実装した。

2）排他的ロック

編集の開始と同時にロックをかける。データの整合性は保証されるがロックの競合がおこり応答時間、スループットになどに影響を与えるためパフォーマンスは前者より低下する。ASP のレコードソースの adLockPessimistic を用いて実装した。

## 4　実行例

前述の機能を現在我が研究室で構築しているインターネットショッピングシステムの顧客データ変更処理に実装して動作確認を行った。

なお使用した DBMS は Oracle7 Server（Oracle 社）と Sybase SQL Server11（SYBASE 社）である。

## 5　システム評価

二層コミットを用いた場合に同時に実行されるトランザクション数と処理にかかる時間との関係を排他ロックと共有ロックを用いた場合にわけて図2と図3に示す。実行時間とは WWW サーバが ASP ファイルの処理に要する時間を表わす。

図2、図3より共有ロックを用いた処理のほうが排他ロックを用いた処理よりロックの競合が少ないぶん実行時間が速いことがわかる。また更新成功時は更新確定の処理を DB サーバに送らなければならないため DB サーバとの接続が生じる。このため更新失敗時より時間がかかることが図2、図3よりわかる。ロックの競合による待ちトランザクションの発生の他にオーバヘッドとして考えられる要因はパケット交換網を使用することによって生じるネットワークの遅延やクライアント側での HTML 処理などがあげられる。

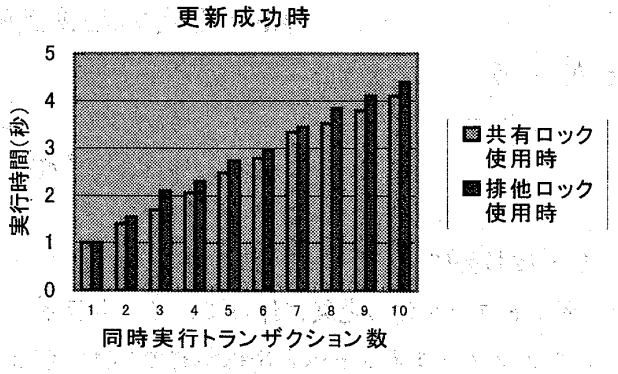


図2　トランザクション数と実行時間との比較（更新成功時）

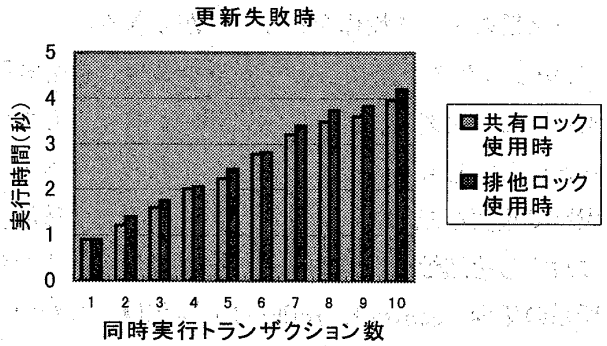


図3　トランザクション数と実行時間との比較（更新失敗時）

## 6　終わりに

本稿では ASP を用いた異機種 DBMS の透過的なアクセス方法について説明した。今回作成したシステムではロックを用いた同時実行制御機能を実装したが、ロックを用いずにタイムスタンプを用いた機能についても考察中である。

### 参考文献

[1]Michael Corning,Steve Elfanbaum:
"ActiveServerPages 開発テクニック",
アスキー出版局,1998