

ミドルウェアのトランザクション処理における同時実行制御プロトコルの開発

永田 元康 大阪教育大学教育学部情報科学講座教授

要旨

「ミドルウェアのトランザクション処理における同時実行制御プロトコルの開発」なる研究助成テーマに取り組み、オブジェクトリクエストブローカーおよび電子商取引のトランザクション処理に関する成果を獲得するとともに、ミドルウェアにおける非同期通信を可能にするメッセ - ジキューに関する成果を得た。我々の研究成果は、下記の様にまとめられる。

オブジェクトリクエストブローカーの同時実行制御プロトコルの開発

- (1) オブジェクトリクエストブローカーにおける並列トランザクション処理
- (2) XMLによるメッセージキュー

レガシーデータ情報ブローカーの同時実行制御プロトコルの開発

- (3) 電子商取引におけるクラスター化したトランザクション処理
- (4) 電子商取引における緊急トランザクション処理

オブジェクトリクエストブローカーにおける並列トランザクション処理は、分散するサーバに配置されたOSおよびプログラミング言語が異なるソフトウェアコンポーネントに渡って実行される、いわゆるグローバルトランザクションの同時実行制御のモデリングおよび、複数サーバにおける並列処理を含むトランザクション処理のプロトコルを提案するとともに、プロトコルのベンチマークテストを報告した。

XMLによるメッセージキューは、オブジェクトリクエストブローカーなどのミドルウェアにおける非同期通信のために使用されるメッセージキューに対して、異なるデータ形式を統一するとともにデータの構造化表現を可能にするXMLを使用して、メッセージおよびメッセージの内部情報をXMLで表現して、データ交換のみならず、メッセージキューに対する非FIFOアクセスを可能とするメッセージキューサービスを構築し、提案した。

電子商取引におけるクラスター化したトランザクション処理は、電子商取引のデファクトスタンダードなプロトコルであるSET等がトランザクションなる概念を採用しつつも本格的な同時実行制御や障害回復の機能が不十分であることに鑑み、一定の時間間隔に到着した複数のトランザクションをクラスターとしてグループ化して、クラスター単位で同時実行制御や障害回復をおこなうことが電子商取引にとって有効であることをタイミングチャートで示し、提案した。

電子商取引における緊急トランザクション処理は、商取引過程において緊急を要するトランザクションを他のトランザクションに先んじて実行するため、緊急トランザクション処理と割り込み先のトランザクションとの衝突、互換などの意味的情報を参照しつつ、状況に応じて割り込み先のトランザクションを処理したのち、緊急トランザクションの処理を優先させるトランザクション処理のプロトコルを提案した。

1 研究背景および目的

「ミドルウェアのトランザクション処理における同時実行制御プロトコルの開発」なる研究助成テーマ、すなわち4つの個別の研究テーマに共通する研究背景として、オブジェクトリクエストブローカーの存在が挙げられる。各々の研究テーマは、オブジェクトリクエストブローカーが抱える未解決な課題を個別に取り組んだものである。

オブジェクトリクエストブローカー(ORB)は、インターネットにおける分散オブジェクトプラットフォームであり、下記に示す主要な機能を備えつつある。

- (1) インタフェース定義言語(IDL)で表現されたインタフェースを仲介した、異種OSや異種言語間におけるメッセージパッシング
- (2) 複数のサーバをまたがって広がる階層トランザクション
- (3) 複雑なデータの管理、特にレガシーデータの管理
- (4) 複数サーバ間のソフトウェアコンポーネント同士の非同期通信
- (5) 電子商取引、ワークフローなどのビジネス情報システムに特有な問題解決

(6) 知的なミドルウェア

オブジェクトリクエストブローカーの代表的なデファクトスタンダードは、米国のOMGなるグループによるCORBA (Common Object Request Broker Architecture) である。CORBA仕様に準拠して、VisiBrokerなどに代表されるオブジェクトリクエストブローカーのソフトウェアが作成され流通している。

(1) オブジェクトリクエストブローカーにおける並列トランザクション処理

当テーマの研究背景は、オブジェクトリクエストブローカーの機能である高度なトランザクションを実現するためにトランザクション処理の Protokolを開発する必要があることである。同時に研究背景として、複数のサーバをまたがって広がる階層トランザクションを対象とするトランザクション処理として、我々が従来から研究してきたオブジェクト指向データベースのトランザクション処理がある。このような研究背景に基づいて、オブジェクトリクエストブローカーにおける並列トランザクション処理として、分散するサーバに配置されたOSおよびプログラミング言語が異なるソフトウェアコンポーネントに渡って実行される、いわゆるグローバルトランザクションの同時実行制御のモデリングおよび、複数サーバにおける並列処理を含むトランザクション処理の Protokolを提案するとともに、Protokolのベンチマークテストを報告することを研究目的としている。

(2) XMLによるメッセージキュー

当テーマの研究背景として、メッセージキューは、オブジェクトリクエストブローカーなどのミドルウェアにおけるソフトウェアコンポーネント間の非同期通信のために使用される機能である。分散システムにおける同期機構としての2相コミットの普及に対比して、メッセージキューはメッセージ指向ミドルウェア(MOM)として、ミドルウェアで重要な位置を占めている。メッセージキューを前面に押し出した代表的なシステムとして、IBMのMQ-Seriesがある。また、メッセージキューに対する様々なDequeueを可能にしたJPMQ(Java Persistent Message Queues)が提案されている。また、別の研究背景として、データ交換を可能にするタグ言語であるXMLがある。このような研究背景に基づいて、ミドルウェアにおける非同期通信のために使用されるメッセージキューに対して、異なるデータ形式を統一するとともにデータの構造化表現を可能にするXMLを使用して、メッセージおよびメッセージの内部情報をXMLで表現して、データ交換のみならず、メッセージキューに対する非FIFOアクセスを可能とするメッセージキューサービスを構築することを研究目的とする。

(3) 電子商取引におけるクラスター化したトランザクション処理

当テーマの研究背景として、電子商取引の Protokolの開発がある。とくに、VisaとMaster Cardの両社が提唱した電子商取引の ProtokolSETは、デファクトスタンダードとして広範に使用されている。このような研究背景のもとで、電子商取引におけるSET等の Protokolがトランザクションなる概念を採用しつつも本格的な同時実行制御や障害回復の機能が不十分であることに鑑み、一定の時間間隔に到着した複数のトランザクションをクラスターとしてグループ化して、クラスター単位で同時実行制御や障害回復をおこなうことを研究目的としている。

(4) 電子商取引における緊急トランザクション処理

当テーマの研究背景として、オブジェクト指向データベースの緊急トランザクション処理に関する我々の研究がある。このような研究背景に基づいて、電子商取引における緊急トランザクション処理は、商取引過程において緊急を要するトランザクションを他のトランザクションに先んじて実行するため、緊急トランザクション処理と割り込み先のトランザクションとの衝突、互換などの意味的情報を参照しつつ、状況に応じて割り込み先のトランザクションを処理したのち、緊急トランザクションの処理を優先させるトランザクション処理の Protokol開発を研究目的としている。

2 オブジェクトリクエストブローカーにおける並列トランザクション処理

オブジェクトリクエストブローカーの同時実行制御 Protokolの開発にあたって、我々は次の考察から出発した。

CORBAは企業情報システムの構築に利用される代表的なプラットフォームである。

CORBAなどのオブジェクトリクエストブローカー(ORB)におけるトランザクション処理の設計要件は、ORBとトランザクションモニタが結合されたものでなければならない。すなわち、オブジェクトは、トランザクションのスコープの中で実行されなければならない。さらに、ネットワーク上の異種言語環境を渡るグローバルなトランザクションを規定することが必要である。このグローバルなトランザクションは、複数のサーバオブジェクトの並列処理を含むスケジューリングを前提とする。このグローバルトランザクションのアトミシティの設定を始めとするACID(atomicity, concurrency, isolation, durability)の規定、トランザクション境界の設定、コミット、直列可能性の規定などのモデル設計は重要である。

このような考察に基づいて、まず、ORBにおけるグローバルトランザクションの定義をおこなった。グローバルトランザクションは、再帰的なメッセージパッシングを伴う、オブジェクト指向プログラムであると定義する。グローバルトランザクションは、メッセージパッシングの発進側の命令から受信側のオブジェクトに向かって、根(root)から葉(leaf)に向かうような木構造をなす。企業間情報システムに対応するORBにおいて、一般性を失うことなく、発進側と受信側はクライアント/サーバまたは異なるサーバに位置する。また、ORBはIDL(interface definition language) サポートのため、発進側と受信側は異なるオブジェクト指向言語でも差し支えない。要するに、グローバルトランザクションは、ネットワーク上の異種言語環境にまたがり、IDLにサポートされた一連のまとまった業務としてのオブジェクト指向プログラムである。従って、ORBにおけるグローバルトランザクションのスケジューリングは、複数サーバにおける並列処理を時には含むような、グローバルトランザクションのコンカレントな実行となる。これから便宜上、グローバルトランザクションをトランザクションと呼ぶことにする。

次に、ORBにおけるトランザクションスケジューリングについて提案する。

ORBにおけるトランザクションスケジューリングは、次のルールに従って実行される。

スケジューリングのルールにおいて使用するプライオリティシーリングについて予め定義する。

定義 トランザクションには、その起動順序に従ってプライオリティが付けられる。オブジェクトのプライオリティシーリングは、そのオブジェクトをアクセスするすべてのトランザクションのなかで最も高いプライオリティを意味する。プライオリティシーリングの値は、トランザクションがアクセスするオブジェクト毎に決められる。

また、トランザクション処理の実行ルールにおいて使用する概念であるサブトランザクションについても定義を与える。

定義 トランザクションに対応する木構造の部分グラフを、サブトランザクションという。サブトランザクションも、トランザクションと同様に、複数のサーバに渡って位置する場合がある。

プライオリティシーリングおよびサブトランザクションの定義に基づいて、トランザクション処理に関する実行ルールを次のように提案する。

定義 トランザクション処理の実行ルール

(1) プライオリティは全てのトランザクションに指定される。トランザクションの起動順序に従って、全てのトランザクションのプライオリティの値が決められる。

(2) 衝突するサブトランザクション間の実行順序のスケジューリングは、プライオリティによって決定される。

(3) 互換であるサブトランザクション間の実行順序のスケジューリングは、プライオリティシーリングによって決定される。

我々は、トランザクション処理の実行ルールに基づいてトランザクション処理のプロトコルを提案するとともに、ORBにおけるトランザクションスケジューリングのベンチマークテストをおこない、報告した。詳細は発表資料を見られたい。なお、ORBにおけるトランザクションスケジューリングのタイミングチャートを図1に示す。

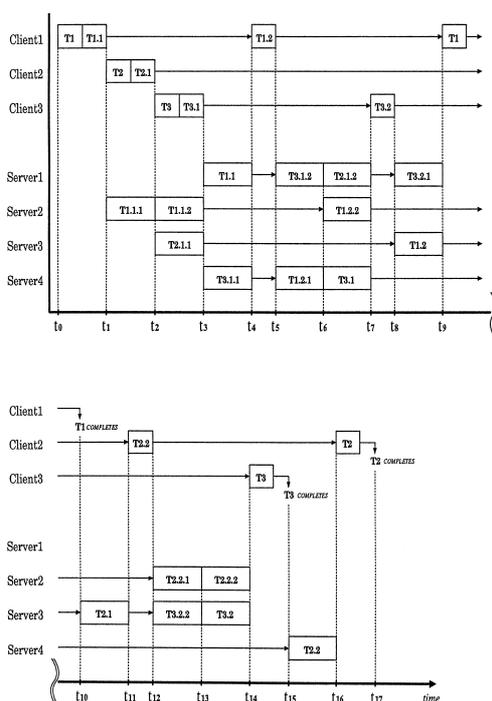


図1 ORBにおけるトランザクションスケジューリングのタイミングチャート

3 XMLによるメッセージキュー

メッセージキューは、オブジェクトリクエストブローカーなどのミドルウェアにおけるソフトウェアコンポーネント間の非同期通信のために使用される機能である。分散システムにおける同期機構としての2相コミットの普及に対比して、メッセージキューはメッセージ指向ミドルウェア(MOM)として、クライアント/サーバ通信またはサーバ間通信が重要な位置を占めるミドルウェアで大きな役割を果たしている。メッセージキューを前面に押し出した代表的なシステムとして、IBMのMQ-Seriesがある。また、メッセージキューに対する様々なDequeueを可能にしたJPMQ(Java Persistent Message Queues)が提案されている。また、データ交換を可能にするタグ言語であるXMLが注目を浴びている。このような研究背景に基づいて、ミドルウェアにおける非同期通信のために使用されるメッセージキューに対して、異なるデータ形式を統一するとともにデータの構造化表現を可能にするXMLを使用して、メッセージおよびメッセージの内部情報をXMLで表現して、データ交換のみならず、メッセージキューに対する非FIFOアクセスを可能とするメッセージキューサービスを構築する。

メッセージのXMLによる格納表現は、メッセージタイプが単数または複数のノード要素から構成され、ノード要素はインスタンスを表す属性の集合であり、葉としてのノード要素はデータを表し、相互参照が可能であるという特徴を利用して実現される。メッセージのXMLによる格納表現の例を示す。

```
<?xml version=" 1.0 " encoding=" UTS-8 " ?>
<!DOCTYPE ex-message SYSTEM
  " http://myhost/ex-message.dtd " >
<ex-message>
  <header>
    <sender id=" 001 " >client</sender>
    <date>2000/01/01</date>
  </header>
  <body>
    <info>Good Computer!</info>
    <item>
      <name>ABC</name>
      <typeNO>XYZ111</typeNO>
      <spec>
        <CPU>PenIII</CPU>
        <MEMORY>64MB</MEMORY>
        <HDD>10GB</HDD>
      </spec>
    </item>
  </body>
</ex-message>
```

メッセージキューの操作は、FIFO方式の基本的なdequeueおよび非FIFO方式のアドバンストなdequeueの双方を提案した。アドバンストなdequeueは、プライオリティが付けられたメッセージに対するdequeue操作である。

メッセージキューの内部構造をXMLで表現することによって、アドバンストなdequeueのみならず、メッセージキューに対して複数のトランザクションからdequeueが要求される場合にも、コンカレントな処理を実現できる。このXMLによるメッセージキューの内部構造の表現を記述する。メッセージキューはrootから格納される。各々のwriterは、書き順にrootの子孫となる。writerがenqueueしたメッセージは、writerノードの子孫となる。writerノードは次のように表現される。

```
<writer writer-id=" wri-a ", bl=" false " >
```

記法bl="false"は、writerがenqueueしたメッセージを誰も読んでないことを表す。一旦reader-1がwri-aのメッセージを読んだならば、bl>trueが代入されて他の誰も読めなくなるような仕掛けになっている。

我々は、JavaプログラミングおよびIBMのXML Parser for Javaを使用することにより、XMLベースのメッセージキューサービスを構築することが出来た。このシステムは、様々なisolationレベルに対応しているため、dequeueにおける非FIFOアクセスが可能となる。

4 電子商取引におけるクラスター化したトランザクション処理

当テーマの研究背景として、電子商取引のプロトコルの開発がある。とくに、VisaとMaster Cardの両社が提唱した電子商取引のプロトコルSETは、デファクトスタンダードとして広範に使用されている。このような研究背景のもとで、電子商取引におけるSET等のプロトコルがトランザクションなる概念を採用しつつも本格的な同時実行制御や障害回復の機能が不十分であることに鑑み、一定の時間間隔に到着した複数のトランザクションをクラスターとしてグループ化して、クラスター単位で同時実行制御や障害回復をおこなうことを研究目的としている。

電子商取引のプロトコルにおいてトランザクション処理の厳密な直列可能性は物理的な意味を持たないという考察が、当研究の出発の動機である。我々はこの考察に基づいて、クラスターなる概念を提案した。

定義 クラスターは、電子商取引において予め設定する一定時間に到着したトランザクションの集合である。クラスターは、1つのトランザクションとみなされて処理される。

トランザクションを生成する様子を図2に示す。ここで、一定時間に到着したトランザクションが1つのクラスターにまとめられる様子が見て取れる。

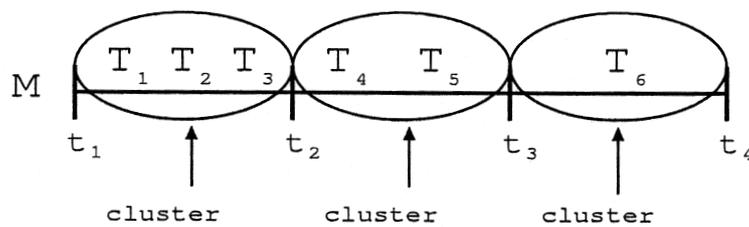


図2 クラスター

プロトコルSETにおけるクラスター単位のトランザクション処理の準備として、SETのトレードフローを図3に示す。

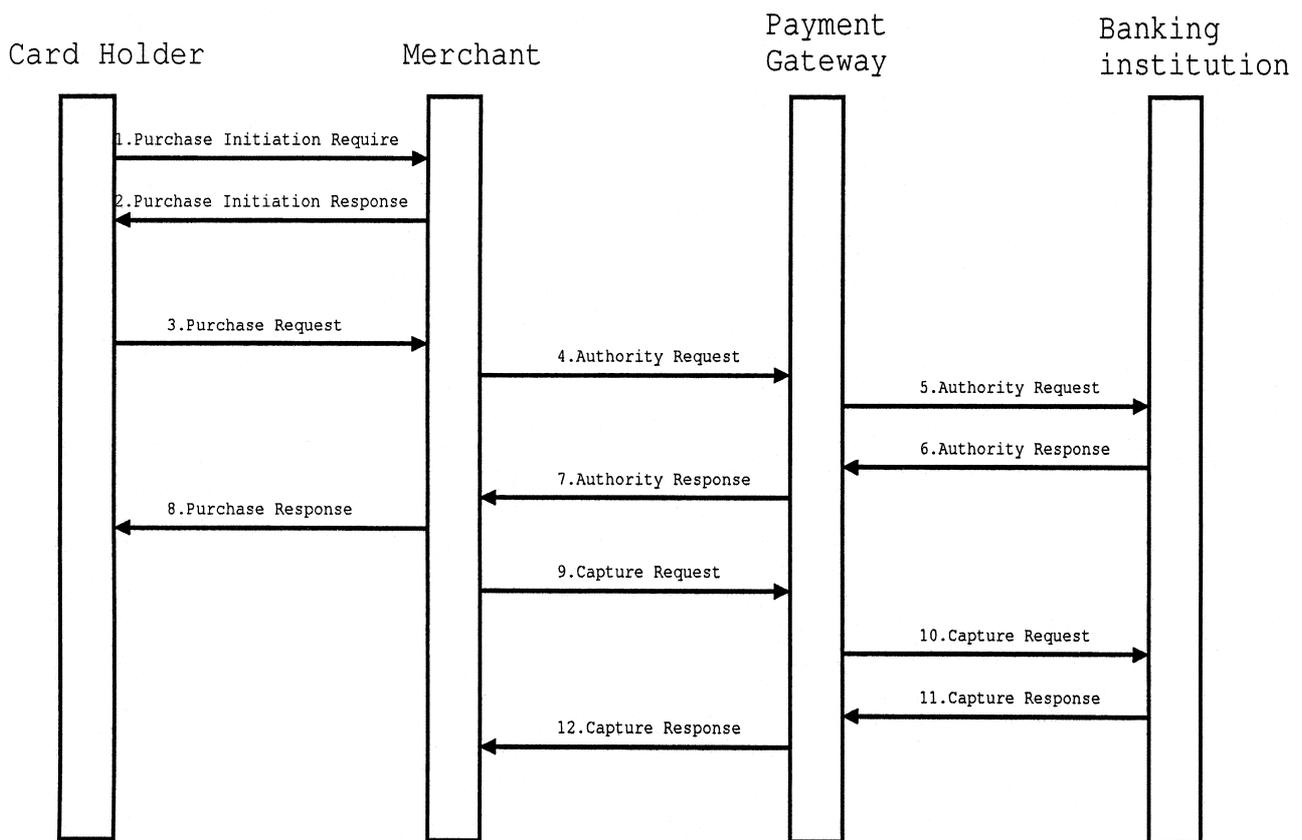


図3 SETのトレードフロー

プロトコルSETにおけるクラスター単位のトランザクション処理は、下記のアルゴリズムに従う。

ステップ1 ベンダー (Merchant) はカード会社(Payment Gateway) から一定時間に到着したトランザクションを1つのクラスターにまとめる。

ステップ2 ベンダーはカード会社にクラスターを送信する。

ステップ3 カード会社は、ベンダーから送られたクラスターを受け取り、クラスター内の各々のトランザクションを処理する。その後、カード会社はクラスターを銀行に送信する。

ステップ4 銀行は、カード会社から送られたクラスターを受け取り、クラスター内の各々のトランザクションを処理する。その後、銀行はクラスターをカード会社にする。

ステップ5 カード会社は、銀行から送られたクラスターをベンダーに送る。

ステップ6 ベンダーは、カード会社から送られたクラスターを受け取り、クラスター内の各々のトランザクションを処理する。

我々は、クラスター化したトランザクション処理における障害回復についても提案した。提案した手法によるクラスター化したトランザクション処理における障害回復を図4に示す。

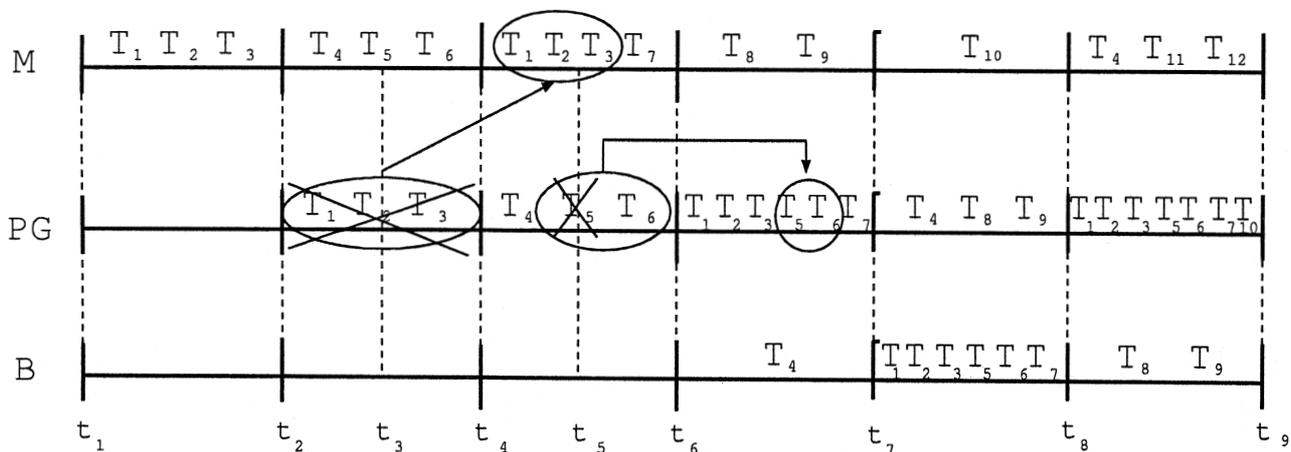


図4 クラスター化したトランザクション処理における障害回復

カード会社(PG)のサーバで処理中、時刻 t_3 においてクラスターC1内のトランザクション T_1, T_2, T_3 にクラッシュが発生したとする。カード会社はクラスターC1の処理をアボートして、時刻 t_4 においてクラスターC1をベンダー(M) に送り返す。時刻 t_4 から時刻 t_6 の間、ベンダーはクラスターC1に新たに到着したトランザクション t_7 を加えてクラスターC3として処理をおこなう。一方、カード会社のサーバで処理中、時刻 t_5 においてクラスターC2内のトランザクション T_5 にクラッシュが発生したとする。カード会社は、トランザクション T_4 のみを銀行(B) に送り、トランザクション T_5, T_6 を同じカード会社の次のクラスターC4に廻す。

5 電子商取引における緊急トランザクション処理

すでに、オブジェクト指向データベースの緊急トランザクション処理に関する我々の研究がある。これは、我々のオブジェクト指向データベースのトランザクション処理に関する研究の副産物である。電子商取引における緊急トランザクション処理は、商取引過程において緊急を要するトランザクションを他のトランザクションに先んじて実行するため、緊急トランザクション処理と割り込み先のトランザクションとの衝突、互換などの意味的情報を参照しつつ、状況に応じて割り込み先のトランザクションを処理したのち、緊急トランザクションの処理を優先させるトランザクション処理のプロトコル開発を目的としている。当研究は、トランザクション群をクラスター化することなく、個々のトランザクションとして処理する場合の緊急トランザクション処理を考察したものである。

4つのロッキングプロトコル、すなわち、2相ロック(2PL)、プライオリティ継承(PI)、プライオリティシーリング(PC)、データプライオリティロック(DP)を比較、考察した。これらの4つのロッキングプロトコルは、下記のように定義できる。定義 2相ロック(2PL:two phase locking) は、直列可能性を実現する代表的な手法であり、次の手順を採る。

2PL スケジューラがトランザクション T_1 の操作 $p_1(x)$ (ここで $p_1(x)=read(x)$ or $write(x)$ by T_1 , ただしdata item x)を受け取る。
 if ($p_1(x)$ (ここで $p_1(x)=lock(x)$ by T_1) がすでにロックを掛けている $q_2(x)$ と衝突)

then トランザクションT1がデータ項目x をロックできるまで、トランザクションT1を待たせる
else pl1(x)にロックを掛ける

定義 プライオリティ継承(priority inheritance)は、あるトランザクションが単数または複数のよりプライオリティの高いトランザクションをブロックするとき、あるトランザクションは、ブロックされているトランザクションが持つ最も高いプライオリティを使って実行する。

定義 プライオリティシーリング(priority ceiling)は、プライオリティ継承の拡張である。トランザクションT があるデータ項目のロックを獲得するためには、T 以外のトランザクションによってロックされるデータ項目のプライオリティシーリングよりも高いプライオリティをトランザクションT は持たねばならない。

定義 データプライオリティロック(data-priority-based locking) は次の手順を採る。各々のデータ項目は、そのデータ項目をアクセスする複数のトランザクションのプライオリティのうち最も高いプライオリティを携行する。新しく到着したトランザクションのプライオリティがそのデータ項目のプライオリティより高いとき、そのデータ項目のプライオリティは更新される。

比較した結果、より高いプライオリティのトランザクションが先行される程度の順序は、 $2PL < PC < PI < DP$ であった。また、総処理時間の長さは、 $DP < PC < PI < 2PL$ であった。

我々は、一般的な緊急トランザクション処理を提案して、これをロックに適用した。緊急トランザクション処理のロックバージョンは、つぎのような条件付きのメソッドで表現される。

メソッド1 緊急トランザクションを直ちに実行する。緊急トランザクションは他のトランザクションよりもプライオリティが高いため、緊急トランザクションは互換である(commutative)他のトランザクションを横取りし(preempt)、2相ロックに従って実行、コミットする。

メソッド2 先行するトランザクションの衝突する(conflicting)部分をアボートした後、緊急トランザクションを実行する。すなわち、この場合はデータプライオリティロックを適用する。

メソッド3 先行するトランザクションの衝突する(conflicting)部分を実行した後、緊急トランザクションを実行する。

6 結論

我々は、ミドルウェアのトランザクション処理における同時実行制御プロトコルの開発という研究テーマの下に、(1) オブジェクトリクエストブローカーにおける並列トランザクション処理、(2)XMLによるメッセージキュー、(3)電子商取引におけるクラスター化したトランザクション処理、(4) 電子商取引における緊急トランザクション処理なるテーマに取り組み研究成果を挙げることが出来た。また、関連する研究成果として、(5) オブジェクト指向データベースのトランザクション処理、(6) WWWデータベース、(7) オブジェクトリクエストブローカーにおけるIDLのキャッシュにおいても一定の研究成果を挙げることが出来た。

参考文献

- (1) 永田元康 分散形データベースシステム、総文館 (1983)
- (2) 永田元康 並列処理・分散処理、コロナ社 (1994)
- (3) 永田元康 オブジェクト指向データベースシステム、森北出版 (1995)
- (4) M. Nagata, Parallel Processing for Indexed Structure of Object, Proc. International Symposium on Next Generation Database Systems and Their Applications, pp. 56-61 (1993)
- (5) Y. Murakami, M. Nishikaku, T. Okada, M. Sakaguchi, M. Nagata, " Real-Time Scheduling for Semantic Concurrency Control of Object-Oriented Database Systems " , Proceedings on Seventh International Workshop on Database and Expert Systems Applications, DEXA96, pp. 214-221 (1996)
- (6) K. Asai, J. Nishibayashi, K. Yoshihara, M. Nagata, " Object-Oriented Serializability in Real-Time Concurrency Control " , Proceedings on Cooperative Database Systems for Advanced Applications, CODAS96, pp. 48-51 (1996)
- (7) K. Asai, J. Nishibayashi, K. Yoshihara, M. Nagata, " Priority-Ceiling Based Concurrency Control of Object-Oriented Database " , Proceedings on IASTED Conference on Artificial Intelligence and Soft Computing, pp. 254-260 (1998)
- (8) M. Sugimori, K. Mizumachi, Y. Oda, S. Miyake, K. Morikawa, M. Nagata, " Contravariance-Based Class Identification for Integration of Class Hierarchies " , Proceedings on IASTED Conference on Artificial Intelligence and Soft Computing, pp. 225-231 (1998)
- (9) M. Nagata, T. Inagaki, M. Kobayashi, M. Tamura, M. Sugimori, Y. Nakamori, S. Furutani, " Object-Oriented WWW Database of Global Environment Model " , Proceedings on Japan-USA Flexible Automation Symposium 98, pp. 209-214(1998)

- (10)Y. Nakamori, S. Furutani, M. Nagata, M. Tamura, " Architecture and Implementation of WWW Database for Global Environment Models ", Proceedings on The Third International Conference on Systems Science and Systems Engineering, pp. 529-534 (1998)
- (11)A. K. Elmargamid (Ed.), Database Transaction Models for Advanced Applications, Morgan Kaufmann (1991)
- (12)J. Gray, A. Reuter, Transaction Processings: Concepts and Techniques, Morgan Kaufmann (1993)
- (13)J. Gray, The Benchmark Handbook for Database and Transaction Processing Systems, Morgan Kaufmann (1991)
- (14)M. S. Merkow, J. Breithaupt, K. L. Wheeler, Building SET Applications for Secure Transactions, John Wiley & Sons(1998)
- (15)A. Ceponkus, F. Hoodbhoy, Applied XML: A Toolkit for Programmers, John Wiley & Sons (1999)
- (16)F. Leymann, D. Roller, Production Workflow, Prentice Hall (2000)
- (17)T. C. Rakow, J. Gu, E. J. Neuhold, " Serializability in Object-Oriented Database Systems ", Proceedings on IEEE 6th International Conference on Data Engineering, pp. 112-120 (1990)
- (18)P. Muth, T. C. Rakow, G. Weikum, P. Brossler, C. Hasse, " Semantic Concurrency Control in Object-Oriented Database Systems ", Proceedings on IEEE 9th International Conference on Data Engineering, pp. 23-32 (1993)
- (19)L. Sha, R. Rajkumar, J. P. Lehoczky, " Priority Inheritance Protocols: An Approach to Real Time Synchronization ", IEEE Transactions on Computers, Vol. 39, No. 9, pp. 1175-1185 (1990)
- (20)O. Ullusoy, G. Belford, " Realtime Transaction Scheduling in Database Systems ", Information Systems, Vol. 18, No. 8, pp. 559-580 (1993)
- (21)H. P. Steiert, J. Zimmermann, " JPMQ-An Advanced Persistent Message Queuing Service ", Proceedings of Advances in Databases, 16th British National Conference on Databases, pp. 182-199, Springer Verlag (1998)
- (22)Fujitsu Limited, Hitachi Ltd., NEC Corp., " Report of Credit Division in SECE Society ", crpp9809.pdf-www.ecom.or.jp(1997)
- (23)J. D. Tyger, " Atomicity in Electronic Commerce ", Proceedings of ACM/IEEE Conference on Principles of Distributed Computing, pp. 8-26 (1995)
- (24)J. Su, D. Manchala, " Building Trust for Distributed Commerce Transactions ", Proceedings of the 17th International Conference on Distributed Computing Systems, pp. 322-329 (1997)

< 発 表 資 料 >

題 名	掲載誌・学会名等	発表年月
Performance Evaluation of Priority Ceiling Based Transaction Scheduling in Object-Oriented Database	Proc. ICCPOL'99 18th International Conference on Computer Processing of Oriental Language	1999年3月
Urgent Transaction Processing for Electronic Commerce	Proc. 1999 IEEE International Conference on Systems, Man and Cybernetics	1999年10月
Remote Method Invocation Based Web Database System for Global Environment Models	Proc. 1999 IEEE International Conference on Systems, Man and Cybernetics	1999年10月
Clustered Transaction Processing for Electronic Commerce	Proc. 2000 IEEE International Conference on Systems, Man and Cybernetics	2000年10月
XML Based Message Queuing	Proc. 2000 IEEE International Conference on Systems, Man and Cybernetics	2000年10月
JAVAのRMIによるウェブデータベースシステムの構築	第43回システム制御情報学会研究発表講演会	1999年5月
電子商取引における緊急トランザクション処理	第43回システム制御情報学会研究発表講演会	1999年5月
オブジェクトリクエストブローカーにおける並列トランザクション処理	第44回システム制御情報学会研究発表講演会	2000年5月
JavaIDLにおけるキャッシュを用いたインターフェースアクセスの高速化	第44回システム制御情報学会研究発表講演会	2000年5月
電子商取引におけるクラスター化したトランザクション処理	第44回システム制御情報学会研究発表講演会	2000年5月