4W－10

分散トランザクションシステム IXI における耐障害性機構

小穴泰裕　松高雄一　中村素典　大久保英嗣　大野豊

立命館大学理工学部情報学科

1　はじめに

分散環境においては，分散配置された共有資源に複数のユーザがアクセスするため，トランザクションの概念が重要となる．各々の分散アプリケーションの構築時において，その都度トランザクション処理を設計および実装するのは非効率的である．以上の問題を解決するために，我々は，分散アプリケーション構築のためのプラットホームである IXI を開発している．IXI を使用することにより，分散アプリケーションの効率的な構築が可能となり，開発コストも軽減される．本稿では，IXI の耐障害性機構について述べる．今回対象とする障害は，トランザクションのアボートおよびサイト障害である．サイト障害は，マシンの突然のダウン等が発生し，トランザクションの処理が中断してしまい，さらに，揮発メモリ中のバッファの内容が失われてしまうことをいう．ただし，この障害では二次記憶装置に格納されているファイルには影響を及ぼさないものとする．

2　分散トランザクションシステム IXI の概要

まず，IXI の特徴について述べる．IXI は弱い一貫性の概念を導入した入れ子トランザクションの記述をサポートしている．また，多重版時刻印方式に基づく複数の方式の集合である適応型時刻印方式により並行処理制御を行っている [1]．また，分散トランザクションの原子性を保証するために，2 相コミットメントプロトコルによるコミットメント制御を行っている．

次に，IXI のシステム構成について述べる (図 1 参照)．IXI は，4 つのシステムタスクおよびユーザインタフェースライブラリにより構成される．トランザクション管理部は，アプリケーションタスクの窓口であり，トランザクションに関する情報を管理する．並行処理制御部は，適応型時刻印方式による並行処理制御を行う．オブジェクト管理部は，共有資源のバージョン管理を行う．回復処理部は，サイト障害発生後にログファイルを走査し，回復処理を行う．ユーザは，トランザクションを発行するクライアント，および実際に資源にアクセスするサーバを記述する．ユーザは，システムの用意するユーザインタフェースライブラリを用いることにより，容易にアプリケーションを構築をすることが可能となっている．

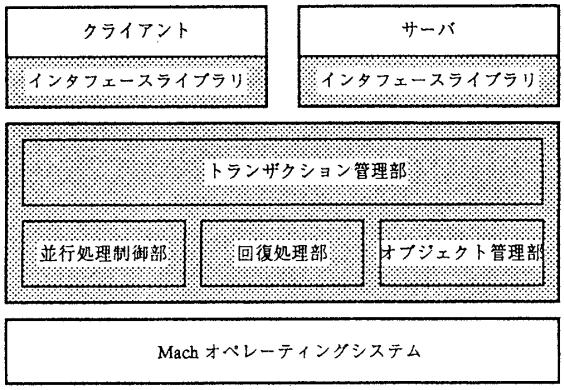

図 1　IXI のシステム構成

3　トランザクションの弱い一貫性およびアボート

IXI では，子トランザクションのモード設定を行うことにより，弱い一貫性を有したトランザクションの記述が可能である．一貫性モードには，通常モード，先行コミットモード，破棄可能モードおよび独立モードがある．表 1 に，それぞれの一貫性モードの違いを示す．ある親トランザクションがアボートした場合には，そのすべての子トランザクションが行った共有資源への操作は無効化されなければならない．先行

Recovery Manager in Distributed Transaction System IXI
Yasuhiro Oana, Yuichi Matsutaka, Motonori Nakamura, Eiji Okubo and Yutaka Ohno
Department of Computer Science, Faculty of Science and Engineering, Ritsumeikan University
1916 Noji, Kusatsu, Shiga 525, Japan

表 1　子トランザクションの一貫性モード

モード	コミット処理のタイミング	アボートする場合の親への影響
通常	親と同時	親はアボート
先行コミット	親に先行	親はアボート
破棄可能	親と同時	親はアボートせず
独立	親に先行	親はアボートせず

コミットおよび独立モードの子トランザクションが既にコミットされ，その後，これらの親トランザクションがアボートされる場合には，これらの子トランザクションの行った操作を無効化する補償トランザクションが発行される．この補償トランザクションも，アプリケーションの設計者が記述する．

4　コミットメント制御

各々のトランザクションは，複数のサイトに分散されて処理される可能性がある．分散トランザクションの原子性を保証するために，それらのサイト間でのコミットメント制御が必要である．我々は，従来から使用されている 2 相コミットメントプロトコルを，弱い一貫性の概念を導入した入れ子トランザクションのコミットメント制御が可能なように変更を加えた．

2 相コミットメントプロトコルは，調整者と従事者のメッセージ交換によって処理が進行する．第 1 相では，調整者が従事者に投票メッセージを送り，従事者はこのメッセージに対して，コミット可能かアボートかを調整者に投票する．第 2 相では，調整者は，従事者からの投票によってコミットかアボートかを決定し，従事者に結果を送信する．従事者はこの結果に従い，コミット処理またはアボート処理を行う．

IXI では，トップレベルのトランザクションが発行されたサイトは，その一族全体の調整者となる．また，各々の子トランザクションが発行されたサイトは，このトランザクションのローカルな調整者となる．弱い一貫性に対応するために，先行コミットおよび独立モードの子トランザクションが開始されたサイトは，親トランザクションの投票メッセージを待つことなく調整者として単独にコミットメント制御を開始する．

サイト障害後の回復処理を可能にするために，トランザクションには状態が付与され，コミットメント制御の進行につれ状態が遷移する．状態には，execute, prepare, prepared, commit, committed, abort, aborted がある．この状態をログファイルに記録しておくことにより，サイト障害発生後のシステム再起動時に，ログファイルを走査することによって回復処理を行うことが可能である．

5　サイト障害に対する処理

サイト障害により，トランザクションの処理が中断した時点では，トランザクションの一貫性が失われてしまっている．このため，システムの再起動時に回復処理を行い，一貫性のある状態に戻す必要がある．ログファイルは，トランザクションの処理が及んだサイトに，トランザクション毎に記録される．システムの再起動時に，回復処理部は複数のスレッドによりログファイルを走査し，主に以下のような情報を探す．

- 当該トランザクションの識別子
- 障害発生時の当該トランザクションの状態
- 当該トランザクションが発行されたサイトか否か
- 当該トランザクションの親および子の識別子

ログを走査した結果，従事者でありなおかつ状態が prepared である場合は次のことを意味する．すなわち，調整者にコミット可能である旨を送信し，コミットまたはアボートの返答を待っている時点で障害が発生しているので，調整者に対して結果の問い合わせが必要となる．

6　おわりに

2 相コミットメントプロトコルでは，従事者が prepared 状態の時に調整者に障害が発生すると，従事者がブロックされる問題がある．3 相コミットメントプロトコルでは，このブロックの問題は解消されるが，メッセージの通信オーバヘッドが増大する．これらのどちらを選択するかは，アプリケーションの設計者が判断して決定するのが適当であり，3 相コミットメントプロトコルについても設計および実装を行う必要がある．

参考文献

[1] 國枝和雄, 畑田孝幸, 大久保英嗣, 津田孝夫: 適応型時刻印方式に基づく同時実行制御, 情報処理学会論文誌, Vol. 33, No. 6, pp. 802-811 (1992).