分散データベースシステムＲＤＢ／ＤＶにおけるリカバリ方式

7H-1

岡崎　卓，武　理一郎，山根　康男，手塚　正義

（富士通研究所）

1. はじめに

著者らは，均質型分散データベースシステムＲＤＢ／ＤＶ［１］のリカバリ方式の方式検討を行った。本稿ではそのうち，リカバリプロトコルとロックログについて述べる。

2. 目的

分散データベースシステムの障害発生時にデータベースの一貫性を保つために実行されるプロトコルをコミットプロトコルも含め，リカバリプロトコルと呼ぶ。

我々はＲＤＢ／ＤＶのリカバリプロトコルの方式検討に際して，

(1) ブロッキングの発生率・持続時間の減少

(2) 信頼性の低いネットワークへの適用性

を狙い，ＢＳＴ（Broadcast on every State Transition）プロトコルを考案した。

また，障害復旧後のデータベースの可用性を高めるためにロックログを導入し，未了トランザクションの存在による資源占有を最小化した。

3. ＢＳＴ　プロトコル

ＢＳＴプロトコルは　(1)正常時のコミットプロトコル，(2)障害発生時に正常なサイトの実行する終了プロトコル，(3)障害サイトが復旧時に行うリスタートプロトコルの３つに大別できる。以下にプロトコルの概要を示す。

(1) コミットプロトコル

メッセージ数が少なくてすむ２フェーズコミットプロトコルとブロッキング発生率の少ない３フェーズコミットプロトコル［２］をユーザがオプションにより選択する。本稿では以下，２フェーズコミットを選択したものとして述べる。

(2) 終了プロトコル

マスタサイトは常にコミット権をもっているため，スレーブサイトの状態に関係無くマスタサイトはサブトランザクションを終了できる。

一方，スレーブサイトにはコミットプロトコル実行中にサブトランザクションを単独では終了できない状態（インダウト状態）が存在する。この状態のときにマスタサイトのダウンやネットワーク障害によりスレーブサイトがマスタサイトと切り離された場合にはインダウト状態が持続（ブロッキング）する。従って，サブトランザクションで使用している資源が解放されないため，データベースの可用性が低下する。

ＢＳＴプロトコルではブロッキングを可能な限り回避するために以下のようなプロトコルを実行する。

通信エラー，タイムアウト等によりスレーブサイトのいずれかがマスタサイトの異常を検出する。

異常を検出したスレーブサイトはａｌｅｒｔ状態に遷移し，ａｌｅｒｔメッセージを他の全てのスレーブサイトに送信（放送）する。

インダウト状態にあるスレーブサイトがａｌｅｒｔメ

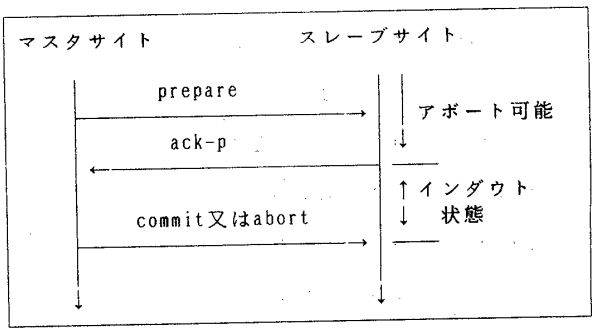

図１　２フェーズコミットプロトコル

Recovery Methods in Distributed Database System RDB/DV

Taku Okazaki, Riichiro Take, Yasuo Yamane, Masayoshi Tezuka

Fujitsu Lab.

ッセージを受信した場合にはａｌｅｒｔ状態に遷移し，再度ａｌｅｒｔメッセージを放送（多段階放送）する。多段階放送によって，連続的な障害に対してもａｌｅｒｔメッセージの伝播を確実にし，ブロッキングの発生率を低下させる。

トランザクションの結果の判明しているサイトはａｌｅｒｔメッセージに対してトランザクションの結果を返送する。また，インダウト状態に達していないサイトはａｌｅｒｔメッセージに対して無条件にａｂｏｒｔｅｄメッセージを返送し，自身もアボートする。

ａｌｅｒｔ状態に遷移したサイトはトランザクション結果が判明した時点で自サイトのサブトランザクションを結果に合せて終了させ，他の全てのスレーブサイトに結果を放送する。このことにより，ブロッキング状態の解決を促進し，また連続的なサイト障害に対する耐性を強化する。

(3)　リスタートプロトコル

マスタサイトが障害を起した場合には障害から復旧した時点で障害前の決定（コミット／アボート）に従い，全サイトに決定を再送信する。スレーブサイトが障害を起した場合には，障害復旧時点でスレーブサイトはａｌｅｒｔ状態に遷移し，ａｌｅｒｔメッセージを発行することにより終了プロトコルを起動する。リスタートプロトコルが終了プロトコルを起動することにより，リカバリプロトコル全体が単純化される。

表１　メッセージの種類

メッセージ	意　味	略型
prepare	コミット準備要求	p
ack-p	コミット準備完了	a-p
commit	コミット要求	c
abort	アボート要求	ab
alert	alert 状態遷移要求	al
committed	コミット済	cd
aborted	アボート済	abd

ＢＳＴプロトコルのメッセージの意味を表１に，メッセージの略型を用いた状態遷移図を図２に示す。

4.　ロックログ

障害復旧時にサイトはダウン時点で終了していなかったトランザクション（未了トランザクション）を終了させねばならない。未了トランザクションのうち障害発生時点でインダウト状態にあったものは他のサイトに結果の問い合せが必要なため，長期にわたって終了しないことがある。障害復旧直後の可用性を高めるためには，これらのトランザクションの使用していた資源のみをロックし，他の資源は解放する必要がある。

このためにロックログを導入した。トランザクション実行中にかけたロックは全てログにとられ，障害復旧時に未了トランザクションでインダウト状態にあったものに関するロックログから，これらのトランザクションに関するロックのみを再現し，他の資源を解放して運用に入る。これにより，未了トランザクションによるデータベース可用性の低下は最小限に抑えられる。

参考文献

［１］安達　他

「分散データベースシステムＲＤＢ／ＤＶの試作」

情報処理学会第２９回全国

［２］S.Ceri, G.Pellagtti

"Distributed Databases", McGraw-Hill, 1984

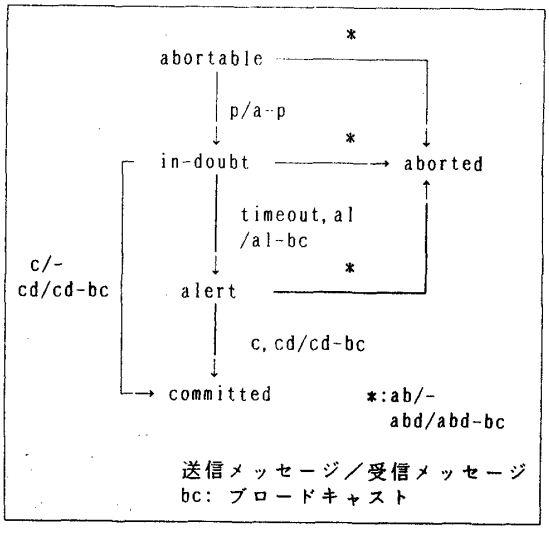

図２　ＢＳＴプロトコルの状態遷移図（スレーブサイト）