

広域分散環境における複数データベース間での レプリカ制御方式

6 X - 6

浅井 健一[†] 川越 恭二[‡]立命館大学大学院理工学研究科[†]立命館大学理工学部[‡]

1 はじめに

最近のデータベースの大規模化と広域分散化により、データベースへの要求も急増しデータベースサーバの負荷やネットワークの負荷の増加をもたらしている。そこで、データベースの一部の複製（レプリカ）を他のデータベースサーバに置きユーザは近接のサーバのレプリカにアクセスする負荷分散方法（以下レプリカ方式と呼ぶ）が通常、使われる。本稿では、広域ネットワーク環境でデータベースサーバ間での情報交換も含む環境を対象に、このレプリカにおける問題点とその解決方法を提案する。

2 広域分散環境とレプリカ方式

レプリカ方式では、通常、レプリカにアクセスした際に目的とするデータが存在しないときには、データの存在するサーバに問合せ、必要なデータを全て取り出し、レプリカを更新する。しかし、この方式では、データレベルのレプリカ制御は行われずレプリカに存在するデータも再度複製のために取り出される。このため、広域ネットワーク環境ではネットワーク負荷が問題となる。また、レプリカを置くサーバでレプリカのためのデータ容量に制限がある場合に、新たなデータをレプリカに加えるときには、存在するレプリカから削除すべき適切なレプリカを選択しそのレプリカを予め削除しなければならない。このため、レプリカのあるサーバの処理負荷が増大する。さらに、今後の広域ネットワーク環境では、特定のデータベースサーバが他のデータベースサーバと連携しデータベースサーバが他のデータベースサーバのレプリカを保持しさらに別のデータベースサーバからの要求に対応することも想定しなければならない。そこで、このような問題を解決するために、広域分散環境でのレプリカ制御方式を提案する。

3 レプリカ制御方式

図 1 に示すような構成のデータベース環境を対象とする。このとき、提案するレプリカ制御方式の基本的考え方を以下に示す。

- 1) レプリカ内のデータごとに転送日時等の情報を持ち制御に用いる。これにより無駄な複製処理を防ぐ。
- 2) どのデータがいつレプリカとして作られたかを、元データのあるサーバで管理を行う。これによって、効率的にレプリカを最新に維持できる。
- 3) レプリカが作成されているサーバをレプリカサーバ、元のデータベースがあるサーバをデータベースサーバという 2 つの概念でレプリカ制御処理を行う。したがって、複数のデータベースサーバが連携している場合でも、統一して扱う。すなわち、広域分散環境に拡大したとき、データベースサーバは、他のデータベースサーバに対するレプリカサーバとなる。

3.1 サーバ管理

レプリカがどのデータベースサーバによるものなのかを把握するために、レプリカサーバでは、各テーブルがどのサーバからのデータであるかを保存する。また、現在のレプリカの状況を把握するため、データベースサーバでは、どのサーバにどのようなレプリカを作成したかを保存する。

3.2 転送

転送では、レプリカサーバには無いデータと、前回レプリカを作成したときよりも後に更新されており、更新しなければならないデータを転送する。

3.3 削除

転送しようとしたデータがレプリカの許容量を超えたときには、削除すべきデータを作成し、レプリカサーバに転送する。レプリカサーバでは、その削除候補を元に削除を行う。また、前回レプリカを作成したときよりも後に更新されており、レプリカを更新しなければならないデータもこの削除データに格納し削除する。削除・転送の必要、不必要を論理条件とした表を表 1 に示す。

A Replica Control Method For Distributed Database Environment

Kenichi Asai, Kyoji Kawagoe.

Ritsumeikan University.

1-1-1, Noji-Higashi, Kusatsu, Shiga,
525-8577, Japan.

表 1. 論理条件と必要処理

条件 \ 項目	レプリカから削除	レプリカに転送
$S_1 \cap R$	○	○
$S_2 \cap R$	×	×
$(S_1 \cup S_2) \cap R$	○	×
$(S_1 \cap S_2) \cap \bar{R}$	×	○

- 発行した SQL によって抽出され、前回レプリカを作成した時より後に更新されたデータ : S_1
- 発行した SQL によって抽出され、前回レプリカを作成したときから更新されていないデータ: S_2
- レプリカにあるデータ : R

3.4 処理の流れ

処理の流れを以下に示す(図 1).

- ① ユーザは近接のレプリカサーバに接続, SQL を送信.
- ② レプリカサーバは SQL を受信, サーバインデックスからデータベースサーバの場所を検索しデータベースサーバに SQL を送信.
- ③ データベースサーバは SQL を受信し, データを抽出. 抽出されたレコードと, レプリカの状態によって, 転送すべきデータを選ぶ. さらに, データベースサーバに他のデータベースのレプリカが存在する場合にはサーバインデックスからデータベースサーバの場所を検索しデータベースサーバに部分 SQL を送信し繰り返す.
- ④ データベースサーバは, 転送すべきデータを転送するときに, レプリカサーバの許容量を超えてしまう場合, データベースサーバは, レプリカサーバから削除するデータを選ぶ.
- ⑤ データベースサーバは, ②の転送すべきデータと③の削除するデータを転送する.
- ⑥ レプリカサーバは, 削除するデータを元にレプリカからデータを削除. 削除完了後, レプリカにデータを受け入れる.
- ⑦ レプリカサーバは, ユーザからの SQL の結果を抽出し, ユーザに転送.

4 データベースサーバの実現

提案方式を実現するために必要なデータベースサーバ内レプリカ制御用のデータ構造を以下に示す.

a. サーバ管理テーブル

レプリカサーバ名, レプリカサーバアドレス, レプリカの許容量を登録するテーブル

[レプリカサーバ名, レプリカサーバアドレス, レプリカ許容量]

b. 転送日時テーブル

テーブル名はレプリカサーバ名とする. 列名はデータベースのテーブル名であり, その値はレプリカへの転送時間を示す.

[ID, テーブル名 1(日付時間型), テーブル名 2(日付時間型).....]

c. データベーステーブル

通常のデータが入っているテーブルの先頭列には, インデックスとして ID, 最終列にそのデータを更新した日付を追加する.

[ID,, 更新時間(日付時間型)]

5 おわりに

現在, Java と JDBC を使用して実装中である. また, ユーザによる出力件数, レプリカサーバの許容量, データベースサーバから転送する件数を変化させた評価を予定している. 今後の課題として, 本方式によるデータベースサーバの負荷の評価, 負荷の軽減方法を検討する必要がある.

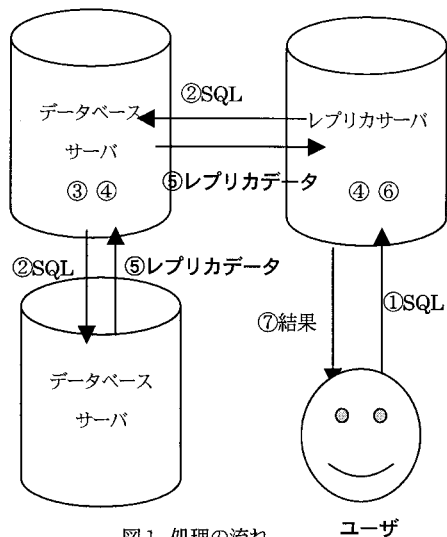


図 1 処理の流れ