Kappa-P のネームサーバ機能

5R−5

坂下之子[1]，合田光宏[1]，佐藤裕幸[2]，河村元夫[3]，

1: (株) アーティフィッシャル・インテリジェンス　2: 三菱電機 (株) 情報電子研究所
3: (財) 新世代コンピュータ技術開発機構

1 はじめに

Kappa-P [1] は、ICOT で開発された並列推論マシン PIM [2] 上で動作する並列データベース管理システムである。

Kappa-P は、複数のローカルデータベース管理システム (LDBMS) から構成される。LDBMS は PIM の各クラスタに配置され、それ自体が完全なデータベース管理機能を持っている。また、データモデルとして非正規関係型モデルを採用し、複雑な知識データを格納、検索する機能がある。それらを複数、並列に動作させることにより、大量の知識データを効率的に処理することができるようになる。このためには、テーブル情報の管理が重要になる。

Kappa-P では、テーブル情報を一元管理するネームサーバ機能を実装した。また、ネームサーバへのアクセス集中を避けるため、ネームサーバの複製を可能にした。本論文では、この Kappa-P のネームサーバ機能の実装について、複製とトランザクション処理を中心に報告する。

2 ネームサーバの設計方針

Kappa-P のネームサーバ機能の設計方針は、以下の 2 点である。

- テーブルの位置をユーザに意識させない。
- アクセス集中を避けるためと耐故障性の向上のため、ネームサーバの複製を作る。

Kappa-P は、複数の LDBMS と水平分割テーブルの機能を提供している。これらはデータアクセスの並列化を目指したものである。しかし、ユーザの立場から考えると、DB のアクセスは複雑にしたくない。そこで、存在位置に関係なく、DB 内ではテーブル名をユニークにつけることにし、テーブルの位置情報は、内部的にネームサーバから得ることにする。これにより、ユーザは、テーブルの位置情報を意識することなく、DB にアクセスすることができる。ネームサーバ機能のある LDBMS をサーバ DBMS (SDBMS) と呼ぶ。

テーブル情報を一元管理しているネームサーバがただ一つのときは、ネームサーバにアクセスが集中し、それが全体の並列性を妨げるボトルネックになってしまう。またその場合、SDBMS に障害が起きたときには、データベース全体がアクセス不可能になる。このような事態を避けるため、異なるクラスタにネームサーバの複製を置き、ある程度は、データベースの動作を保証することを考える。別クラスタで並列に動くのであれば、複製の増加がネームサーバへのアクセス実行に与える影響は小さいと考えられる。

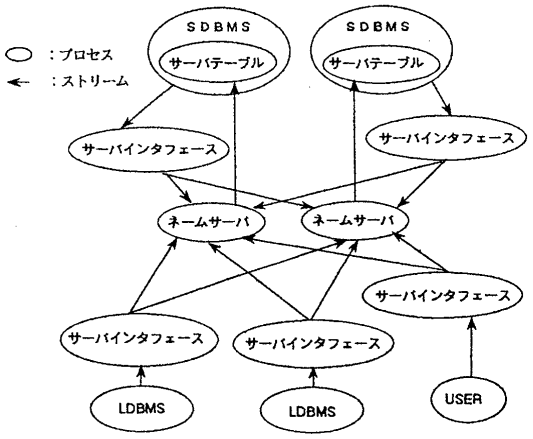

図 1: サーバの構成

そこで DB の複製方法として、Special copy (primary copy 、 available copy)、Voting などのアルゴリズムを検討したが、Special copy による複製の場合、対応できる障害が限られていることが問題となった。また primary copy は読みだしの集中が問題となり、 available copy は実装が複雑になることが問題となった。その結果、対処できる障害に制限がなく、実装が比較的簡単な Voting [3] を採用することにした。

3 サーバの構成と複製方式

Voting のため、個々のネームサーバへの通信路を持ったサーバインタフェースをつくる (図 1)。テーブル情報の登録・更新、問い合わせなどのネームサーバ機能は全て、サーバインタフェースを通して行われる。これにより、ネームサーバが幾つ存在するかなどを意識することなしに、処理ができるようになる。

ネームサーバ内では、テーブル情報はテーブル名で管理され、バージョン番号が付けられる。一度ネームサーバに登録されたあとは、そのテーブル名の情報が更新される度にバージョン番号がインクリメントされる。サーバインタフェースは、複数のネームサーバに問い合わせた結果が異なっている場合、バージョン番号の最も新しいものを正しい情報と判断する。削除されたテーブルは削除情報で識別し、バージョン番号はインクリメントされて保存される。これは、最新情報としてテーブルが削除されていることを示す。

Name service function on Kappa-P
Yukiko SAKASHITA[1], Mitsuhiro GOUDA[1], Hiroyuki SATO[2], Moto KAWAMURA[3]
1:Artificial Inteligence Co.　2:Mitsubishi Electric Corporation
3:Institute for New Generation Computer Technology.

Voting のためのネームサーバの数は以下のように決定される。ネームサーバがN個あり、各ネームサーバに1票づつ割り当てる。読みに対して、R個のサーバの読みが必要であり、更新に対してW個のサーバの更新が必要である、とすると、R + W = N+1 の式を満たす数である。

サーバインタフェースからの Voting のアルゴリズムは、以下のようになる。

問い合わせ： サーバインタフェースは、各ネームサーバに対してテーブルの情報を尋ね、R個以上のネームサーバから返答が返ってきたとき、そのなかでバージョン番号の一番新しいものを最新情報として扱うことにする。

登録 / 更新： サーバインタフェースは、ネームサーバにテーブル情報を問い合わせ、その最新情報から登録 / 更新可能であると判断したら、最新のバージョン番号をインクリメントして更新情報を作り、各ネームサーバにメッセージを送る。W個以上のネームサーバから登録成功の答えが返ってきたら、その登録を成功とする。

これにより、最低R個の SDBMS が存在するかぎりは、ネームサーバからとりだした情報を信用することができる。

4 実装

実際のネームサーバは、 SDBMS のテーブルを利用して情報を保持、提供する。一つのネームサーバには、一つのサーバテーブルが対応している。DB がアクセス可能な状態の間は、ネームサーバ以外のユーザからアクセスできないように、これを排他ロックしている。サーバテーブルの1レコードには、1つのテーブル名についての情報が格納されている。

以下に、実装されたネームサーバのトランザクション管理と同時実行制御について報告し、問題になった点について述べる。

4.1 トランザクション管理

Kappa-P の LDBMS は、2相コミットによるトランザクション管理 [4] や、リカバリ処理を提供している。Votingで必要なトランザクション管理は、2相コミットと異なる。

一般の2相コミットでは、指定したテーブルのうち1つでも失敗すれば正常終了できない。しかし Voting 用のトランザクションでは、指定したテーブルのうち必要数のテーブルがロック / コミットできればよい。

このため Voting のトランザクションに2相コミットをそのまま使用すると、アルゴリズムとは異なり、本来書き込みが正常終了するものが、エラーになる場合がある。アルゴリズムでは、必要数のサーバテーブルに書き込めれば、サーバテーブルへの登録 / 更新は成功するのだが、2相コミットでは、ロックしたテーブル全ての書き込みが成功しなければ、トランザクションは失敗する。しかし、このような障害発生はまれなので、実用上問題ないため、現在の実装では2相コミットを少し修正して使っている。

一般のトランザクションからの変更は次の点である。

- 必要数のテーブルがロックできればトランザクションが開始できる。

トランザクション開始以後は、2相コミットプロトコルに従う。これによりロックできたテーブルが全部書き込めない限り、サーバテーブルへの登録 / 更新はできない。

読み出しについてはアルゴリズム通り、必要数のサーバテーブルが読めた時点で、 Voting を行なっている。

4.2 同時実行制御

現在の LDBMS ではレコード単位の同時実行制御は行っていない。上で述べたように、テーブル情報の更新の要請が来たとき、サーバインタフェースでは、 Voting に必要な数のサーバテーブルを、トランザクションで排他ロックしている。実際に排他制御が必要なのは、ユーザが指定したテーブル名の情報があるレコードだけであるが、現在の実装では、テーブル単位に排他制御されており、このことが並列処理の妨げになっている。

さらにもう一つの問題は、ロックするサーバテーブルの選び方である。実装は、全てのサーバテーブルをロックしてみて、成功したテーブルは全部使っているが、必要数より多くのテーブルに対してアクセスしていることもある。また、どのサーバテーブルを使うかという選択 (SDBMS の負荷の分散化など) も考慮する必要がある。

5 まとめ

以上のネームサーバの機能により、複数の LDBMS とそこに配置されたテーブルを、位置情報の指定なしに扱うことができるようになった。また、ネームサーバの複製により、アクセスの集中の緩和と安全性を実現することができた。が、幾つかの問題点も残っている。今後の課題としては、以下の点が挙げられる。

- Voting 向きトランザクションの検討
- ネームサーバの選択の最適化、分散化
- ロックするサーバテーブル数の最少化

これらを解決すれば、より効率的なネームサーバを作ることができる。

最後に、実装と本論文の作成にあたって、さまざまな助言、御指導をいただいた ICOT の方々、 Kappa-P の関係者の方々に深く感謝いたします。

参考文献

[1] 河村ほか: 並列データベース管理システム Kappa-P の概要, 第45回情処全国大会, 5R-3, 1992-10.
[2] 瀧 : "Parallel Inference Machine PIM" The International Conference on Fifth Generation Computer Systems '92, 1992.
[3] S.H.Son, "Replicated Data Management in Distributed - Database Systems", SIGMOD RECORD, Vol.17, No.4, December 1988.
[4] 椿野ほか: 並列データベース管理システム Kappa-P のトランザクション制御, 第45回情処全国大会,5R-6,1992-10.