D-016

無共有型 DBMS 向けデータ領域リマッピング機能の開発

Proposal of Database Area Remapping for Shared-Nothing Database Management System

伊藤 大輔† 牛嶋 一智†
Daisuke Ito† Kazutomo Ushijima†

1 緒言

近年、IT リソースを複数業務間で共有し、必要に応じてその割り当てを変更することでシステムの総保有コスト (TCO) 削減を目指す IT プラットフォームが注目されている。本プラットフォーム上で動作するミドルウエアにはクラスタサーバ上でノード数割り当て変更などに対応し、その構成を柔軟に変更できることが要求される。しかし、これらミドルウエアのうち、データを分散格納することを特長とする無共有型データベース管理システム (DBMS) は、ノード数増加に対するスケーラビリティは高いが、構成変更時にノード間でデータ移動が起こるため、構成変更を迅速に行うことができない。そこで、我々は、複数のノードでストレージを共有するストレージエリアネットワーク (SAN) 環境を前提として、構成変更時にノード間でストレージ割当て情報のやり取りを行い、データ移動を省略することで構成変更所要時間を短縮するデータ領域リマッピング機能を提案した。本報告書では、提案機能の概要と、その効果について述べる。

2 研究の背景と課題

企業間の競争激化に伴い、ビジネスプロセスの継続的な変化に追随可能で、TCO の低い IT プラットフォームが要求されている。この要求を満たすために、Policy-Based Computing Services コンセプト[1]に基づく IT コスト削減技術が各 IT システムベンダから提案されている。これらの技術の一つに、複数業務の統合運用がある。これは、従来は業務ごとに別々のシステムとして設計し、また運用されてきた複数の基幹系業務システムを、単一のシステムとして統合運用する技術である。統合運用は管理コスト / 機器コストの両面から TCO を下げる効果がある。

また、統合運用を実現するにはブレード型計算機を用いることが有効である。図 1に示すように、ブレード型計算機からなるクラスタの上で中長期的に予測可能な負荷の変動に合わせて業務ごとのリソース割り当てを変更することで、計算機の総保有数を削減できる。

図 1に示すような運用を行う場合、システムを構成するミドルウエアには高いスケーラビリティを有し、かつ割り当て計算機数変更に伴う構成変更を数時間以内に行え

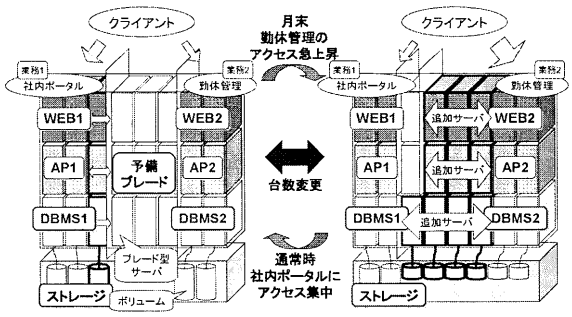
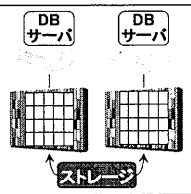
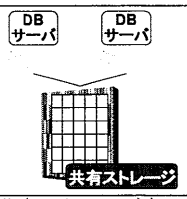


図 1：ブレード型計算機を用いた運用の例

† (株) 日立製作所 中央研究所
Hitachi, Ltd., Central Research Laboratory
{d.ito, ushijima}@crl.hitachi.co.jp

表 1：DBMS アーキテクチャの比較

アーキテクチャ	無共有	ストレージ共有
構成図		
データの配置	各ストレージに分割配置	共有ストレージ上に一括配置
スケーラビリティ	○ (16 サーバ以上)	× (8 サーバ以下)
構成変更所要時間	× (数時間～数日)	○ (数秒)

ることが要求される。しかし、現状では典型的な基幹系業務システムで用いられる WEB サーバ / アプリケーションサーバ / DBMS の 3 つのミドルウエアのうち、DBMS だけがこれら 2 つの要求を両立させることは難しい。

そこで、「高いスケーラビリティを有する」「構成変更を数時間以内に行える」という 2 つの要求を満たす DBMS を提案することを本研究の課題とする。

3 課題解決のための提案手法

3.1 既存の並列 DBMS の分析

現在実用化されている並列 DBMS アーキテクチャには、表 1に示す無共有型とストレージ共有型の 2 つがある。表に比較したとおり、これら 2 つは、データの配置方法が本質的に異なる。本研究では課題の 1 つである高いスケーラビリティを容易に実現可能な無共有型 DBMS を元に機能拡張を行う。

3.2 無共有型 DBMS における課題

無共有型 DBMS の構成変更所要時間はデータ移動が発生するため長くなる。ここで、DBMS の扱うデータはテーブルとインデックスであるため、データ移動は「テーブルの移動」と「インデックスの再作成」に分割できる。

一般に基幹系業務システムで用いられるテーブルは、情報系業務のテーブルと比べてインデックスが少ないことが知られている。一例として、基幹系業務を模した TPC-C ベンチマーク[2]の 2004 年 4 月時点での Top 10 データではテーブル 9 つに対してインデックスは最大 10 個、また単一テーブルへのインデックスは最大 2 個であった。

表 2：単価測定環境

項目	内容
テーブルスキーマ	TPC-C の OrderLine テーブル
インデックス列	外部キー参照に対応した 2 個
実験環境	2CPU サーバ × 2 台 FC-RAID

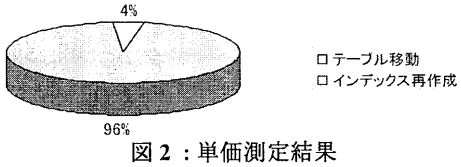


図 2：単価測定結果

そこで、TPC-C の OrderLine テーブルを用いて、表 2に示す環境でデータ移動を行った際の単価を図 2 に示す。この例ではデータ移動所要時間のうち、96%がテーブルの移動所要時間である。このことから、基幹系業務システムにおいて構成変更所要時間を短縮するためには、テーブルの移動時間削減が有効であることが分かる。

3.3 提案手法

ブレード型計算機を用いた運用ではストレージ管理コスト削減のため、SAN を用いてデータ管理を一元化する。SAN は DB サーバ毎に独立した論理ボリュームを静的に割り当てることで無共有型 DBMS でも利用可能である。

ここで、我々は、SAN を用いた無共有型 DBMS では、DB サーバは物理的には任意の DB サーバに割り当てられた論理ボリュームにアクセスできることに着目したデータ領域リマッピング機能を提案する。ここでは、共有ストレージを予め適当な定数の小領域に分割し、ハッシュ分割法を用いてデータをこれら小領域に分割して保存する。これら小領域を DB サーバに重複なく割り当てるために、マッピング管理機構を用いる。構成変更に伴う DB サーバ数の変更の際には、マッピング管理機構を用いて小領域と DB サーバの再割り当て (リマッピング) を行う。一例として、図 3に新規 DB サーバを追加する際の再割り当てを示す。このようにテーブル移動の代わりに小領域のリマッピングを行うことで、構成変更所要時間を大幅に短縮することが可能となる。

なお、無共有型 DBMS には、クライアントからのトランザクション要求に対し、DB サーバ毎のデータ割り当てを考慮してジョブを割り振るトランザクション受付サーバが存在する。このためトランザクション実行中にリマッピングを行うと DB サーバ毎のデータ割り当てが変わり、トランザクションが継続不可能になる。この問題を回避するために、リマッピング機能を用いて構成変更を行う場合には、図 4に示すとおり、トランザクション受付サーバを用いてトランザクション要求を一時保留する。

4 提案手法の検証

構成変更所要時間に関して、提案手法を用いることで図 2のテーブル移動部分を数秒に短縮可能である。再び表 2の環境を用いると、例えば従来 1 日を要していた場合でも、1 時間以内に実行可能である。

また、新機能追加の性能への影響を調査するため、プロトタイプを作成し実験を行った。特に DB サーバ内で小領域を跨ぐ処理のオーバーヘッドが大きいことが予想されるため、インデックス無しテーブルの全件検索を想定したクエリを投入した。実験条件および従来手法に対する提案手法の相対値をそれぞれ表 3、図 5に示す。4 台以上の場合に従来比 95%の性能を有することが分かる。

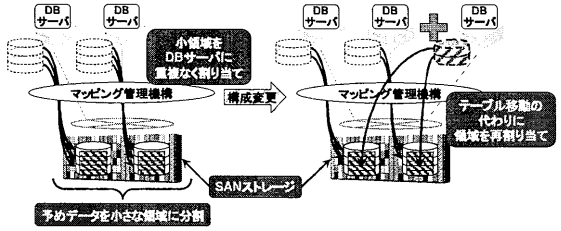


図 3 : データ領域リマッピング機能の動作例

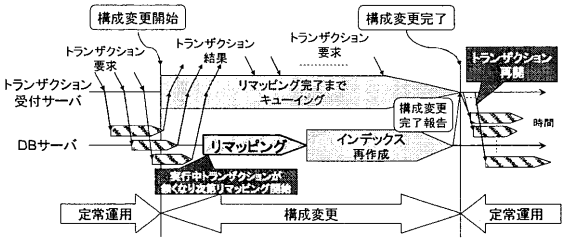


図 4 : トランザクション管理方式

表 3 : 実験条件

項目	内容
テーブルスキーマ	TPC-H[3]の Orders テーブル
データサイズ	150 万件
実験環境	2CPU サーバ × 8 台 FC-RAID
測定方法	全件検索クエリのレスポンス時間を 40 回計測した平均

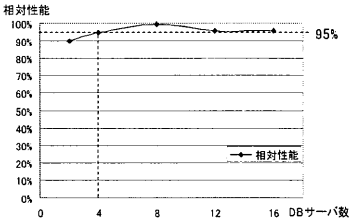


図 5 : 実験結果

5 関連研究との比較

従来の並列 DBMS は、SAN 登場以前のアーキテクチャをそのまま採用し続けていた。一方、提案手法では定常動作中は無共有型 DBMS に準じ、構成変更の際にはストレージ共有型 DBMS に習い SAN 上の共有ストレージの存在を前提とすることで、構成変更の大幅な高速化を達成した。これは従来の無共有型 / ストレージ共有型 DBMS とは一線を隔するものである。

また、提案手法のように、共有ストレージを論理的に分割し、それらを複数の計算機に重複なく割り当てる機能は論理ユニット (LU) として広く用いられている[4]。しかし、従来の LU は DB のコンテンツを理解できないため、LU 単独で提案手法のように小領域単位での割り当て変更を行うことはできない。また、DB サーバから指示を与えて LU を操作し割り当て変更を行うことも考えられるが、これはソフトウエアのモジュラリティを損なう。以上のことから提案手法は従来手法より優れることが分かる。

6 まとめと今後の課題

本研究では、SAN 環境でデータ割り当て変更を行うことで無共有型 DBMS の構成変更所要時間を大幅に短縮するデータ領域リマッピング機能の提案を行った。また、プロトタイプを用いた評価実験を行い、提案機能の性能への影響も許容範囲であることを確認した。以上のことから、提案機能を用いることで、複数業務を統合運用した際に、予測可能な負荷変動に合わせたリソース割り当てを柔軟に行えることが確認できた。

今後の課題として、インデックス再構成部分の高速化手法を提案し、データ領域リマッピング機能の適用範囲を拡大することが挙げられる。

参考文献

[1] Gartner, "Policy-Based Computing Services: The Vision, The Reality", White Paper, 2002
[2] TPC, "TPC Benchmark C", Standard Specification
[3] TPC, "TPC Benchmark H", Standard Specification
[4] SNIA, "Shared Storage Model", White Paper, 2001