異種プラットフォーム混在環境下での並列DBMS(HiRDB)の実現と評価

1R－3

八高 克志　下川 隆義　正井 一夫　亀城 嘉人
日立製作所ソフトウェア開発本部

1.はじめに

近年，ライトサイジングが行われ，従来メインフレームで処理していた大規模データベース(DB)をワークステーションで実現するようになった．一方，DB を用いた業務の拡大とともに扱うデータ量も飛躍的に増大し，その増加率の予測は困難になっている．このような状況下でスケーラブルな並列データベース管理システム(DBMS)が注目を集めている．従来の並列 DBMS は同種プラットフォーム環境下での動作を前提としていたが，異種プラットフォーム混在環境下で並列 DB の構築が可能となることにより，運用面での柔軟性が向上し，DB 活用方法に新たな可能性が生まれる．日立製作所ではスケーラブル並列関係 DBMS HiRDB を開発し，各種プラットフォームに移植している．本発表では，ニューヨークで開催された DB Expo '96 で展示した，HiRDB を用いた異種プラットフォーム混在環境下並列DB(混在環境並列DB)について評価を行う．

2.HiRDB の特徴

異種プラットフォーム混在環境下での並列 DB 構築の視点から HiRDB の特徴について説明する．

2.1.Shared-Nothing 型並列アーキテクチャ

並列 DBMS のアーキテクチャにはプロセッサ間で主記憶および DB を格納したディスク共有する Shared-Everything 方式，ディスクのみ共有する Shared-Disk 方式，どちらも共有しない Shared-Nothing 方式の 3 種類がある．共有する資源が多いほど DBMS の構造は非並列 DBMS に類似し，プロセッサ間の負荷バランスがとりやすくなるが，ハードウェアの制約や資源排他の問題からスケーラビリティが低くなる傾向がある．

HiRDB ではスケーラビリティを最重視し，Shared-Nothing 方式を採用している．この方式は，プロセッサ間での均等な負荷分散が困難であることが弱点であるが，独自の表分割法およびフロータブルサーバを導入することにより解決をはかっている．一方で，Shared-Nothing 方式は各プロセッサ間の同期，相互制御機構が最も単純化されるので，混在環境並列 DB の構築が容易になる．

2.2.HiRDB のサーバ構成

HiRDB は図 1 に示した分散マルチサーバ方式のソフトウェア構造を持っており，サーバを各プロセッサに分散配置することによって並列性を実現している．各サーバの役割について以下で説明する．

(a)SQL受付サーバ クライアントから投入されたSQL受け取り，解釈，最適化して各 DB 処理サーバに処理を要求する．また，DB 処理サーバからの処理結果を収集してクライアントに返却する．

(b)DB処理サーバ 分割表を格納したデータ格納単位ごとに存在し，そのデータに専任でアクセスを行う．また，ソートなどの演算処理を行う．

(c)フロータブルサーバ 分割表を持たない DB 処理サーバであり，負荷分散を目的としてプロセッサに配置され，ソート，ジョインなどの CPU ボトルネックとなる処理を担当する．これによって，Shared-Nothing 方式の欠点であるデータの偏在にともないサーバ負荷が偏る問題の解決をはかる．

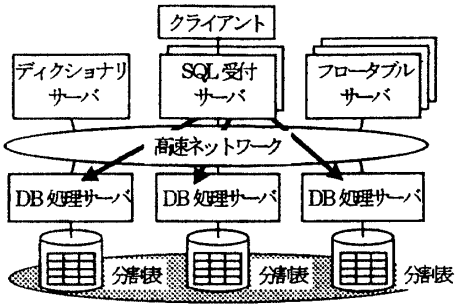

図 1 HiRDB のサーバ構成

このほかに，定義情報を管理するディクショナリサーバ，ログを出力するログサーバ，トランザクション処理用のサーバ群などがある．

2.3.非同期データ待ち合わせによるサーバ連携

これらサーバでシリアルに処理を行うのは SQL 受付サーバにおけるクライアントとの間でやり取りを行う部分のみである．DB 処理サーバ，フロータブルサーバなどは独立に動作し，非同期のデータ待ち合わせにより連携をとっている．このことは，HiRDB のアーキテクチャが真の Shared-Nothing 型であり，シェアードメモリや，POST-WAIT 機構によりプロセッサ間の制御を行う Shared-Everything 方式とは一線を隔していることを意味する．これによって，プラットフォームに依存しない機構を用いて並列 DBMS が実現できる．HiRDB は非同期データ通信機構として TCP/IP+ソケットをサポートしている．

2.4.中間コードによる処理要求

SQL 受付サーバで受け付けた SQL は内部表現にコンパイルした後，該当する DB 処理サーバに伝えられる．この内部表現は CPU のネイティブコードのようなものではなく，DB 操作の実行手順を示した，プラットフォーム非依存の中間言語である．そのため，異種プラットフォーム混在環境下でも問題無く使用することができる．

2.5.オープンプラットフォーム設計

HiRDB はもともと UNIX 系の複数のプラットフォームをターゲットとしており，特殊化された OS の固有機能を用いずに実現できるよう設計されている．ソースコードのうち，

Realization and Evaluation of Parallel DBMS—HiRDB—on the Heterogeneous Platform
Katsushi Yako, Takayoshi Shimokawa,
Kazuo Masai, Yoshito Kamegi
Hitachi Software Development Center

性能要求などでプラットフォームに依存する部分は，各プラットフォームごとに用意したコードに置き換えられるよう局所化されている．また，システムコールなどの UNIX 系 OS が提供するは機能は，ほぼ同等機能でありながら仕様がプラットフォームごとに少しずつ異なるのが現状である．HiRDB では，これらの仕様差をもつ OS 提供機能をカプセル化している．以上により，OS 差異依存部分を全体の 1%程度に押さえることができ，完全に同一仕様の DBMS を各種プラットフォームで用意することが可能となった．これにより，混在環境並列 DB の実現が可能になる．

3.異種プラットフォーム混在環境下並列 DB 構築

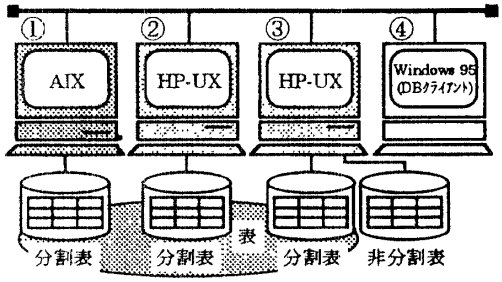

図 2プラットフォーム構成

ここでは，ニューヨークで開催された DB Expo '96 で展示した混在環境並列 DB について説明する．

3.1.プラットフォーム，サーバ構成

DB Expo '96 で展示したプラットフォーム構成を図 2 に示す．①②③に HiRDB/Parallel Server をインストールし並列 DB を構築した．各プラットフォーム用の HiRDB は製品ベースのものであり，特に変更を加えていない．①に SQL 受け付けサーバ，ディクショナリサーバ，DB 処理サーバ，②③には DB 処理サーバが配置してある．また，④にクライアントマシンとして PC が接続されている．

各プラットフォームのスペックを下表に示す．

	OS	CPU	主記憶
①	AIX*	PowerPC† 67MH	128MB
②	HP-UX‡	PA-RISC§ 100MH	64MB
③	HP-UX	PA-RISC 100MH	128MB
④	Windows 95**	Pentium†† 100MH	32MB

①②③における CPU スペックは，ほぼ同程度であるといえる．主記憶サイズにばらつきがあるが，いずれも各サーバがオンメモリでで動作するには十分な量であり，性能には影響しないと思われる．

3.2.表分割

デモ用データとして，実際の大規模 DB 運用を想定した 100 万件のデパートの売り上げ情報を格納した．売り上げ情報は主キーである売り上げ ID を用いたキーレンジ分割法によって均等に分割し，①②③上の分割表に格納した．また，並列性の検証のために同じデータを分割せずに③上の非分割表に格納した．

4.結果

IBM AIX4 および HP HP-UX10 を用いた混在環境並列 DB の構築は可能であった．また，分割表および非分割表に対して全件検索を行った結果，それぞれ，約 6 秒と約 18 秒というプロセッサ数に反比例したレスポンスタイムを得た．

5.検討

HiRDB がプラットフォーム非依存に設計されていおり，並列効果も十分に発揮されることを確認した．

今回，混在環境並列 DB が構築可能となったことにより，次のような並列 DB 運用の可能性が考えられる．

i. DB構築時点で入手可能なプロセッサを用いて混在環境並列 DB を構築する．
ii. DB のスケールアップの際に，その時点で最もコストパフォーマンスの高いプロセッサを増設する．
iii. ローカルエリアネットワーク上のすべてのプロセッサにフロータブルサーバを配置し，DB のデータ処理に CPU 使用率の低いプロセッサを動的に割り当てる．
iv. インターネット上に DB 処理サーバを分散配置し，並列動作可能で高速な分散 DB 的運用をする．

iii および iv は将来機能であり，HiRDB の改良が必要であるが，実現性は高いと考える．これらの運用を考える場合，プロセッサを同種に統一することは事実上不可能であるため，混在環境 DB 構築が可能であることが重要となる．

異種プラットフォームを混在させた場合，プロセッサの能力差が問題となる．しかし，HiRDB は各サーバが独立に動作するので，各 DB 処理サーバが担当する分割表のサイズを調整することで容易に問題を解決できる．

一方，現時点では，混在環境下並列 DB を構成するプロセッサに，主記憶内の int 型のサイズ，境界制約およびバイトオーダーなどが同一であるという制限がある．これらオブジェクト配置の異なるプラットフォームを混在させる場合には，通信電文などのプロセッサ間で共有する外部データをオブジェクト配置非依存に改良する必要があると考えている．

6.おわりに

今回の展示により，HiRDB で採用した Shared-Nothing 方式およびプラットフォーム独立性の高い構造をもつ DBMS は混在環境並列 DB の構築に適していることが実証された．また，混在環境並列 DB が可能になることにより，柔軟な並列 DBMS 運用の可能性が示せた．

参考文献

[1] 正井他：「更新処理を並列実行する UNIX 向け DBMS を開発」，日経エレクトロニクス，1995.2.27(No.630)
[2] 根岸他：「並列リレーショナルデータベースシステム HiRDB の概要と基本技術」，電子情報通信学会 信学技報 DE95-79, 1995
[3] 藤原他：「並列 RDB システムにおける通信機能の実現方式」，情報処理学会台 49 回全国大会講演論文集,7W-3 1995
[4] 金丸他：「並列 DB サーバ HiRDB における SQL 最適化処理方式」,情報処理学会第 52 回全国大会講演論文集,3Q-5 1996

* AIX は，米国における米国 International Business Machines Corp. の登録商標です。
† PowerPC は，米国における米国 International Business Machines Corp.の商標です。
‡ HP-UX は，米国 Hewlett-Packard Company のｵﾍﾟﾚｰﾃｨﾝｸﾞｼｽﾃﾑの名称です。
§ PA-RISC は，米国 Hewlett-Packard Company の商標です。
** Windows は，米国およびその他の国における米国 Microsoft Corp. の登録商標です。
†† Pentium は，米国 Intel Corp.の登録商標です。