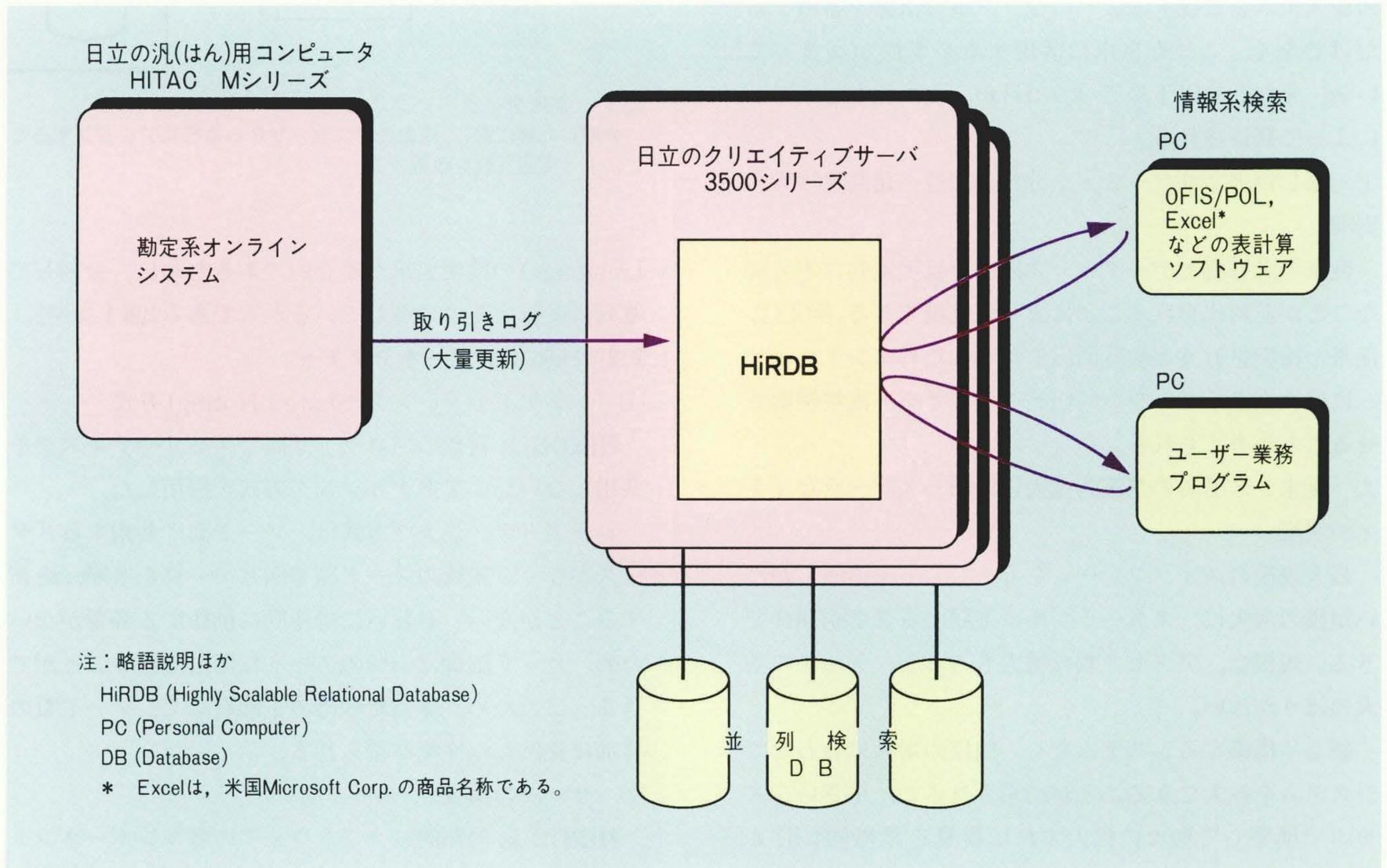


並列データベース適用による新金融情報系システム

—スケーラブルデータベースサーバ“HiRDB”—

New Financial Information Analysis System Using Parallel Database

正井一夫* Kazuo Masai
古賀旨茂** Muneshige Koga
高山 浩* Hiroshi Takayama



HiRDBの金融情報系への適用

金融情報系では並列データベースを適用することにより、信頼性、生産性の高いシステムの構築を目指す。HiRDBは、SQL並列機能、トランザクション並列機能をサポートし、高スループットを実現する。

最近、並列処理の技術が科学技術計算だけでなく、ビジネスユースでも注目されている。その代表的なものが並列データベースである。並列データベースを導入することにより、ビジネスアプリケーションに大きな変化が生じている。金融系のアプリケーションはその典型的な例であり、特に金融情報系では大量のデータを扱っており、並列データベースの効

果が大きい。

そのため、日立のスケーラブルデータベースサーバ“HiRDB”(Highly Scalable Relational Database)を並列データベースの例としてあげ、その特長と金融情報系への適用の可能性、および並列データベースを適用することによって発生する業務の変化の可能性について提案する。

* 日立製作所 ソフトウェア開発本部 ** 日立ソフトウェアエンジニアリング株式会社

1 はじめに

近年、データベースの大規模化に従って並列処理技術が注目を浴びている。それは、計算機のハードウェアの進歩に比べてデータ量の増加のほうが速く、並列化でそのギャップを埋めることが必要となったためである。特に金融系では、金利自由化の流れの中にあって、情報系のシステムを重視するようになり、単に情報を蓄積するだけでなく、これを自由に活用する必要性が高まっている。大別して以下の二つの項目が、並列化を行うことによって期待される。

(1) 新しいアプリケーション分野の創造—超高速検索の実現

従来の計算機とデータベース技術では現実的に不可能なことが並列化を行うことによって実現できる。例えば、従来、検索を行うと数時間以上かかるためエンドユーザーには検索させていない明細データなどを、直接検索させることが考えられる。

(2) 従来システムの飛躍的拡大に追随—スケラビリティの実現

最大規模のメインフレームをもってしても追随しがたい規模の増大に、スケラブルに追随することが期待できる。規模は、アクセス数の増大もあるが、データの増大のほうが速い。

新しく構築するシステムでも、規模の増大に合わせてシステムを拡大できることが約束されると、小さいシステムで構築して順々に拡大すれば投資を段階的に行える。また、一度開発したシステムを移行せずに維持できる適用効果大きい。

ここでは、新しい並列データベースとして日立製作所が開発したスケラブル データベース サーバ“HiRDB”，および金融情報系のシステムでHiRDBを用いた新しいシステムについて述べる。

2 並列データベース “HiRDB”

2.1 並列データベースの概念

並列データベースでは、複数のプロセッサ上にデータベースを構築し、同時に並行して検索を行うことによって高速な応答を得ることができる。

こうした並列検索ができるのは、一つのデータベースを各プロセッサに分割して格納(分割表と言う)することにより、各プロセッサから並行にアクセスできるようにしてあるからである。さらに、SQL(Structured Query

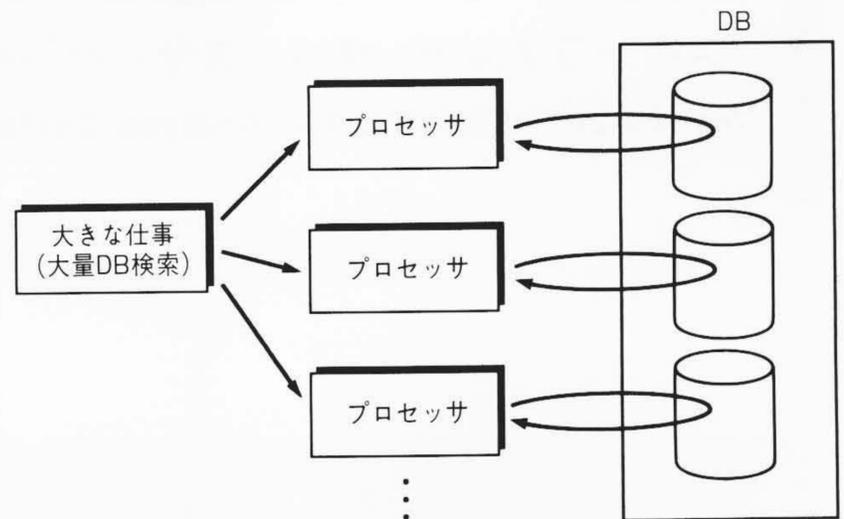


図1 並列データベースの概念

分割した表に対し、複数のプロセッサから並行にアクセスすることによって応答性を改善する。

Language)の検索要求が集合的であるために、分割して並列に実行するのに適しているからである(図1参照)。

2.2 HiRDBのアーキテクチャ

(1) シェアドナッシング(Shared Noting)方式

HiRDBは、複数のプロセッサ間でメモリやディスクを共用しないシェアドナッシング方式を採用した。

シェアドナッシング方式は、ノード間で共用するリソースがなく、複数のノード間で同じデータを参照、更新することがない。お互いに排他的に動作する必要がないため、ロック制御は一つのノードに閉じて行うことができる。このため、並列処理時の干渉がなく、ノード数の増加に比例した性能が得られる。

(2) サーバの構成

HiRDBは、内部的にソフトウェアの各コンポーネントをサーバ化し、各サーバを複数のノード上に分散配置した分散マルチサーバ方式で実現されている。これにより、ノードの物理位置に対して透過的なソフトウェア構造を実現し、さらに障害発生時の可用性を高め、サーバ間の負荷バランスを容易にしている(図2参照)。

クライアントから与えられるSQL文は、外部ネットワークに接続しているノードのプロセッサが受け付ける。このプロセッサで、受け取ったSQL文を解釈してデータの存在するノードを調べ、そのノードのデータベース処理サーバに処理を通知する。さらに、データベース処理サーバが処理を終えたとき、その結果をクライアントに通知する。

(3) フロータブルサーバの導入

さらに特徴的なことは、HiRDBには「フロータブルサーバ」と呼ぶ、CPU処理の負荷をバランスするためのサーバを定義することができることである。このサーバは分割表を持たないデータベース処理サーバであり、例え

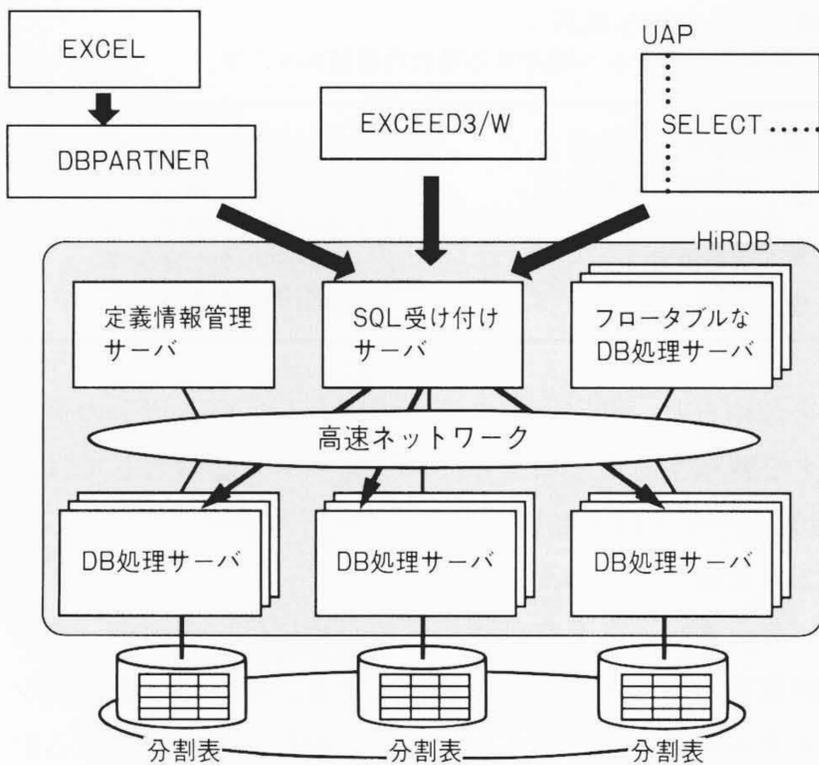


図2 HiRDBのサーバ構造
各サーバを複数のプロセッサ(ノード)へ配置する。SQL受け付けサーバはSQLを分割し、各サーバへの実行指示を行う。

ば、大量検索時のソート処理やマージ処理など、CPU負荷の高い仕事を分担して特定ノードに集中した負荷を分散することができる。

2.3 HiRDBの特徴

HiRDBの特徴について次に述べる(図3参照)。

(1) スケーラビリティ

HiRDBは、複数のプロセッサ上で動作できる。各プロセッサ間は、ネットワークで接続されている以外は共有しているリソースを持たない。したがって、非常に高いスケーラビリティが実現できる。処理量が増えてもプロセッサを増加することによって対応できる。データベースのデータ量が増えても、プロセッサ追加で一定の応答時間を保つこともできる。

(2) 広範囲に並列処理技術を適用

(a) HiRDBでは、検索用SQL(SELECT)だけでなく、更新系SQL(UPDATE, INSERT, DELETE)も並列化して実行できる。

(b) 運用関連機能(データロード、バックアップ・リカバリ、リランなど)を並列で実行できる。通常の運用機能で並列化できないものがあるとネックとなり、大規模なデータベースの維持は難しい。

(3) 高信頼性

(a) HiRDBの基本的回復機能は、日立のメインフレーム上のリレーショナルデータベース“XDM/RD”

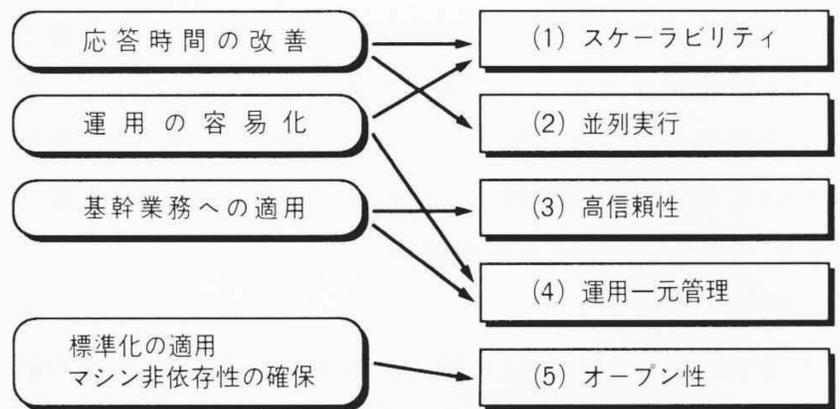


図3 HiRDBの特徴
HiRDBは、基幹系の業務にも耐えられるような、信頼度の高いシステムの中核となる製品である。

(Extensible Data Manager/Relational Database)の技術を踏襲している。

(b) データベース容量増大に対応できるように、回復処理を並列化して強化した。シェアドナッシング方式のため、各データベースアクセスサーバは、おのこのログをもとに自律的に並行して再開始処理を行うことができる。

(c) 各プロセッサ(ノード)障害時は、ノードを切り替えて処理を続行できる。シェアドナッシングではあるが、障害対策用に2台のノードでディスクを共有しておき、HA(High Availability)モニタと呼ばれる監視機構を用いて行う。通常時は片系からだけアクセスするので、シェアドナッシングの特性を失うことはない。

(4) オープン性

(a) HiRDBは、HI-UX/WE2ベースのデータベースであるが、特別なハードウェアに依存していないため、高いポータビリティを持つ。今後、サーバについては、米国IBM社のSP2、米国Hewlett-Packard社、SUN^{※1)}のプラネットフォームへの搭載を予定している。

(b) 各種標準化(ISO SQL, OSI RDA, X/Open^{※2)} XAなどに準拠する。

3 HiRDBの適用

金融機関でHiRDBの特徴を生かすと、ほとんどすべての業務を対象にすることが考えられる。既存システムのソフト資産の継続性が重要となる勘定系オンラインシステムについては、当面メインフレームでのシステム運用

※1) SUNは、米国での米国Sun Microsystems, Inc.の登録商標である。

※2) X/Openは、X/Open Company Limitedの英国ならびに他の国における登録商標である。

が現実的であり、リレーショナルデータベースを必要とする情報系システムへの適用があげられる。さらに、データ処理の高速化技術は、長年の問題点として顕在化しているバッチ処理の改善に適用することが期待できる。

3.1 情報系システムへの適用

情報系システムとしての役割は、銀行内外で発生、入手できる情報(データ)を用い、最終的に銀行としての収益および顧客サービスの向上を図るものである。このため各金融機関は、システムのインフラストラクチャが本来の目的を達成するような最適システムの構築を実現するため、情報系システムの開発を行ってきた。しかし、データベース構築の面で、ハードウェアとソフトウェアの大きな技術的制約が生じてきた。本来、すべての顧客を対象とした管理、分析を行いたいところを、コストパフォーマンスを考慮して、対象顧客を限定し、またデータの事前加工(検索パターンによる集計)を行っていたからである。そのため、情報系システムのデータベース構築にあたっては、(a)大量データの蓄積、(b)大量データの高速度更新、(c)大量データの高速度検索、(d)容易なデータ編集の4項目を構築に必要なシステム要件とした。

(1) 大量データの蓄積

対象となるデータ規模は、約300万顧客以上である。さらに統計計数データ(営業店数×勘定科目×履歴数×データ属性(残高、積数、平残など)をテーブル化したもの)

表1 具体的な業務
情報系システムへ適用する場合の業務例を示す。

収 益 管 理	顧客別収益管理	マ ー ケ テ ィ ン グ	顧客管理(渉外支援)
	営業店別収益管理		資金フロー分析
	予実算管理		エリアマーケティング
	予算策定		テレマーケティング
	ALM		取引明細履歴検索

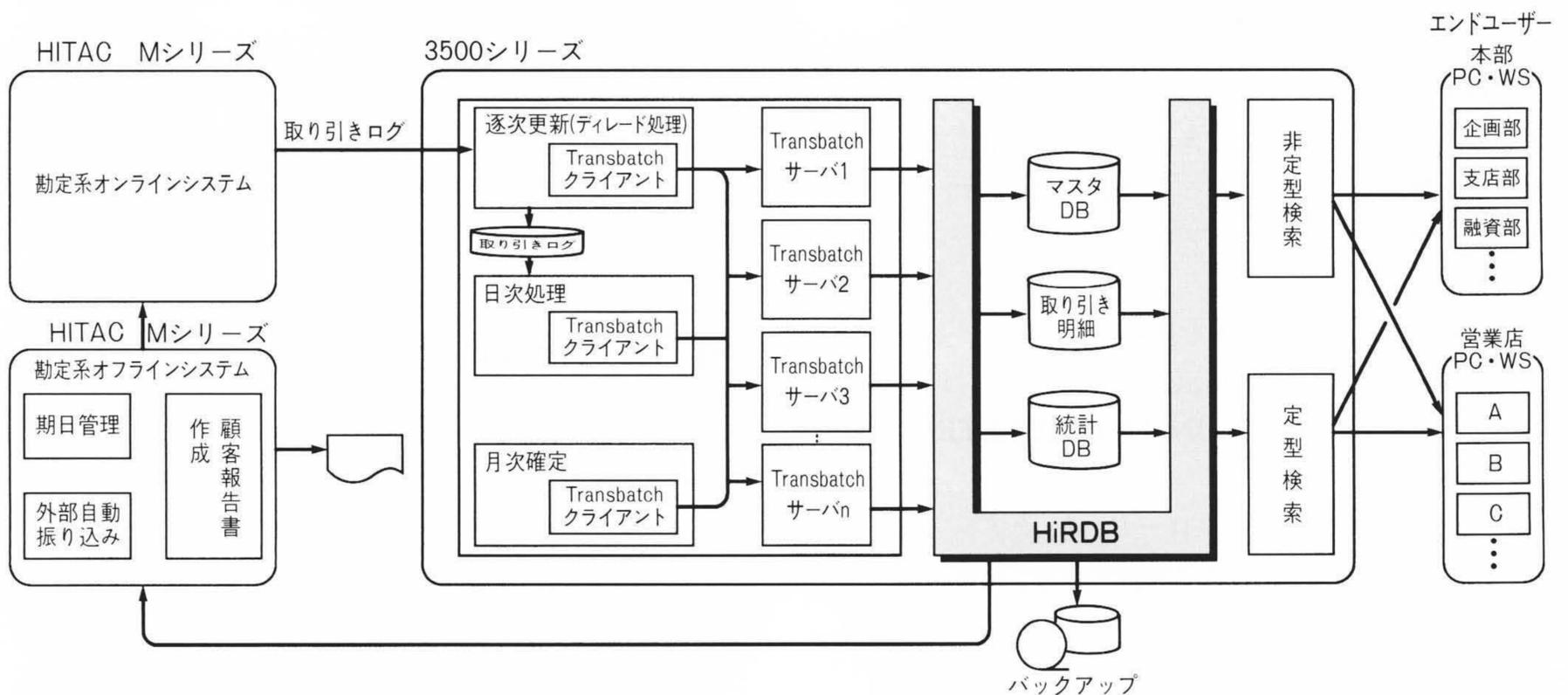
を含めると、勘定系システムで保有している以上のデータを蓄積することになる。データベース容量としては、100 Gバイト以上必要となる。

(2) 大量データの高速度更新

データベースを有効活用するためには、そのデータの鮮度を保つことが重要な要件となる。そこで、情報系システムで保有するデータベースに対し、日々変化する計数データを逐次更新する必要がある。また月次による計数確定処理が必要になる。その更新性能は、勘定系オンラインシステム(ピーク時約10万件/h, 100万件/d)と同程度必要である。

(3) 大量データの高速度検索

データベース検索の応答時間は、システムの操作性を評価する一つのポイントとなる。検索業務はデータベースの全件数から一定の条件に従ったデータ抽出(条件検索、絞り込み検索)が主になる。具体的な業務としては、複数顧客の一括検索が考えられる。これは、ある顧客と



注：略語説明 WS (Workstation)

図4 新情報系システムイメージ
金融系情報システムへの適用例である。大量データの検索・更新にHiRDBの適用が考えられる。

関連のある顧客(系列企業など)が主となる顧客をもとに一括検索するものである。これらの検索応答時間では、秒オーダーを保証する必要がある。従来この応答時間を短くするために、データベースの細分化を余儀なくされシステムの複雑化を招いていた。

(4) 容易なデータ編集

エンドユーザーがデータベースを活用するためには、データ編集に必要な、(a) ソート、(b) マージ、(c) テーブルの結合(Join)、(d) データの集約(Grouping)の四つの基本的な機能を高速に実行する必要がある。

さらに、エンドユーザーが使い慣れたスプレッドシートへのデータの取り込みにより、データの2次加工、つまりエンドユーザーコンピューティングの推進が図られる。情報系システムへの適用例を表1に示す。

3.2 バッチ業務への適用

情報系システムの充実によってバッチ業務は減少傾向になると考えられるが、オンラインシステムでの一括処理であるBMP(Batch Message Program)データの作成、月次処理、決算処理などは、勘定系オンラインシステムの大幅な変更がないかぎり、依然として高負荷なデータ処理として今後も残る。その対応として、Transbatch(大量データ分散処理支援機能)とHiRDBの組み合わせによる大量データ分散並行処理機能の採用がある。バッチ業務への適用は、情報系システムと一体となって取り組むことにより、いっそう大きな成果が期待できる。具体的には、月末マスタ作成処理を情報系システムに移行することが考えられる。月末マスタ作成は、各口座ごとの月末時点の残高や利息計算を行っている。この処理は通常10時間程度かかって膨大な計算能力を必要としてお

り、システムのスケーラビリティが重要になる。また、既存のバッチ処理自体の削減対象として、エンドユーザー向け帳票作成処理があげられる。これは、情報系システムとしてのデータベースを充実させることにより、本部または営業店のエンドユーザーから要求されていた帳票をデータベース検索によって代えるものである。

3.3 新情報系システム

当面、基幹となる勘定系システムは、実績のあるメインフレームでのシステム構成が現実的である。新情報系システムは、勘定系システム(メインフレーム)と、UNIX^{※3)}システムである日立のクリエイティブサーバ3500シリーズマシンと接続することにより、データの受け渡しを行う。そのシステムイメージを図4に示す。

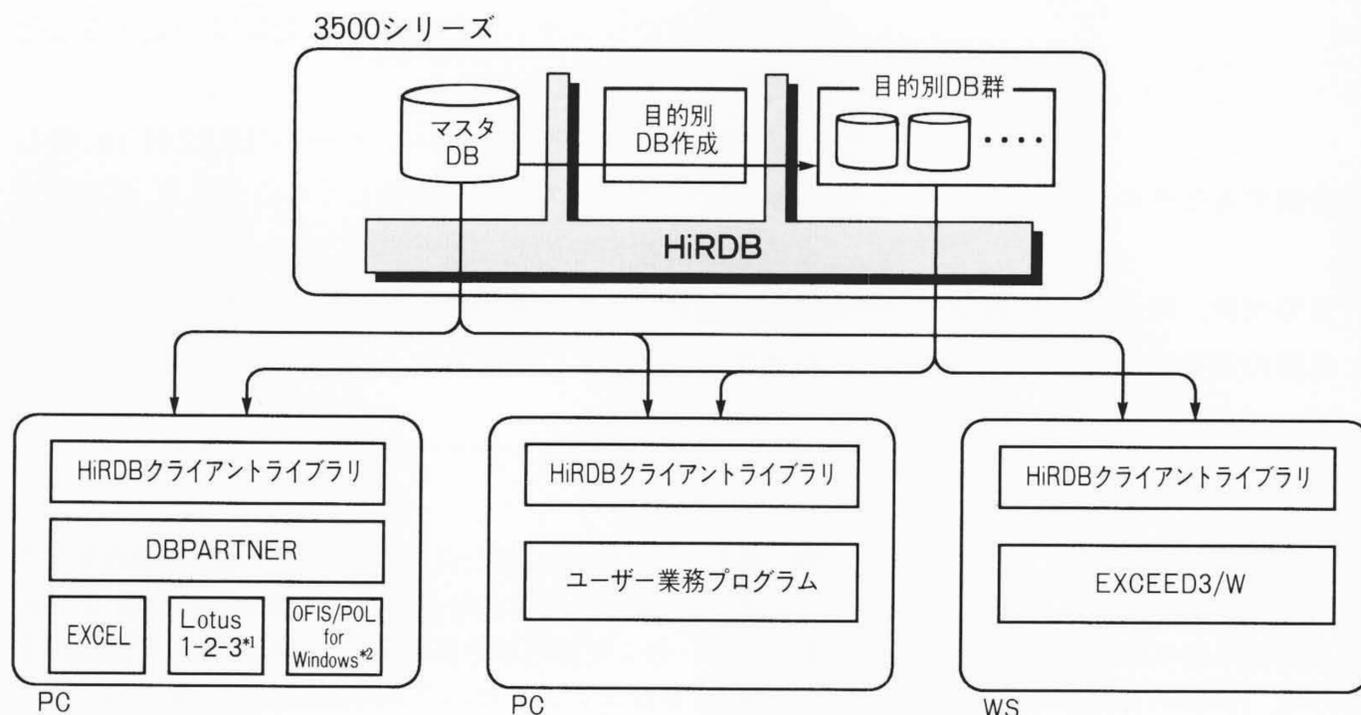
(1) 逐次更新

勘定系オンラインシステムで発生する勘定取り引きログは、発生につど、もしくは一定間隔で新情報系システムに取り込み、マスタデータベースの即時更新を行う(ディレード処理)。この処理は、勘定系システムの能力に追随する必要があり、データ量の伸びとともに性能を向上(スケーラビリティ)させる必要がある。

(2) 日次処理

勘定系オンライン終了後、日次確定や各種目的別データベース(統計データベース、ALMデータベースなど)の作成・処理を行う。日次処理の処理時間については、勘定系システムのバッチ処理同様に大量のデータを処理するため、分散、並列処理の効果が期待できる。

※3) UNIXは、X/Open Compay Limitedがライセンスしている米国ならびに他の国における登録商標である。



注：*1 Lotus1-2-3は、米国Lotus Development Corp.の商品名称である。

*2 Windowsは、米国Microsoft Corp.の商標である。

図5 データベース検索パターン

並列データベース化によるレスポンスの向上の結果、目的別データベースだけでなく、マスタデータベースの操作も可能となる。

(3) 月次処理

日次処理同様、各口座の月末時点での計数を確定し、マスタデータベースの更新を行う。

(4) 非定型検索

データベースの利用形態は、その時々エンドユーザーニーズによって変化する。従来、その変化をシステム部門で新規検索メニュー、もしくは帳票作成という形で対応していた。

しかし、要求した時点から要求したものができまでの時間(ターンアラウンドタイム)が数日から数か月と長く、またシステム部門の開発案件(バックログ)が年々増加する一方であった。非定型検索は、ターンアラウンドタイムの短縮とシステム部門のバックログ解消を行うものである。この機能は、HiRDBとパソコンソフトを連携するソフト“DBPARTNER”(Windows対応DBアクセスツール)や、エンドユーザーが日ごろ使い慣れたスプレッドシートを用いて実現する(図5参照)。

(5) 定型検索

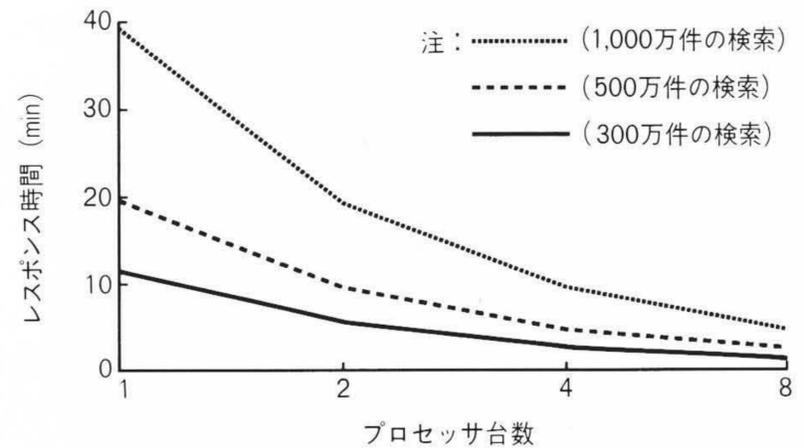
エンドユーザーから要望の多い照会処理やそれが不変的なものは定型メニューとして作成し、システムの操作性を高める必要がある。その際のプログラム開発では、非定型検索によるデータ検索と、スプレッドシートによるデータ編集を行ってプロトタイプを作成し、それを手直ししていくスパイラルアプローチが可能である。さらに、オブジェクト指向言語でのユーザーアプリケーションプログラム開発によって生産性向上が図れ、常に生きた情報系システムにすることが可能になる。

これらHiRDBが提供する機能を使うことにより、操作性、信頼性、生産性の高い情報系システムの構築を図ることができ、本来求められているシステムへ一歩前進することが可能である。

4 性能と規模

マスタデータベースを全件検索するモデルを用いて、並列機の台数規模を見積もる。

性能は、データベースの行長や列数、取り出し行数などによって変動する。性能と規模の例を図6に示す。



注：3500/540, または545程度のマシンを想定(行長500バイト, 出力行数1,000件の場合)

図6 全件検索性能

プロセッサ台数に応じて、スケーラブルな応答が期待できる。

1,000万件の検索を1台のプロセッサで行った場合、約40分かかるケースが、4台では約10分というようにスケーラブルな検索性能が期待できる。

5 おわりに

並列データベースの秘めている可能性は非常に高く、特に金融情報系のように大量の情報を扱う場合は、その効果が期待できる。

並列データベースを用いて再構築を行うことは、業務の形態にも少なからず影響する。現状、膨大な帳票を出力しているが、参照頻度の低いものはその場で検索することができるようにし、帳票を出力しない方向になると思われる。また、参照頻度の高いものは目的別データベースを事前に作成するほうが効率が良い、これもプログラムレスでパラメータ類を指定するだけで作成できるようになる。エンドユーザーがみずから目的別データベースを自由に構築する、または改良することもありうる。これによってシステム部門の負荷を大幅に削減することができると思う。

スケーラブルデータベースサーバ“HiRDB”は、新しい並列データベースとして開発したものであり、これによって金融情報系での超高速検索とスケーラビリティの実現が可能となる。今後も機能、性能、信頼性のニーズにこたえて、“HiRDB”の技術の向上に努めていく考えである。

参考文献

- 1) 鳥居, 外: 高拡張性を旨とした並列RDBサーバ, 電子情報通信学会技術研究報告, DE94-49(1994-9)
- 2) 並列マシン向けDBMS技術, 90年代半ばの実用化を目指す: 日経エレクトロニクス, No.586, 1993年7月19日号
- 3) 藤原, 外: 並列RDBシステムにおける通信機能の実現方式, 情報処理学会第49回全国大会講演論文集, 7W-3, 1993
- 4) 正井, 外: 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, No.630, 1995年2月27日号