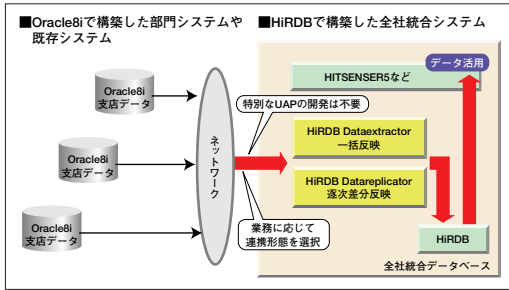


図3-9：Oracle8iとのデータ連携



データベースシステムから専用のプログラムを作成しなくても、HiRDBに情報を簡単に抽出できる。しかも、こういう異種DBMS間で必ず煩わしい問題となるデータ型や文字コードの違いも、自動的に変換する機能を装備しているので、ほとんど意識せずにDBMS同士を連携することができる。

物理的・組織的に離れた部門システム同士の情報をスケラビリティに優れたHiRDBに格納すれば、全社規模のデータの把握や、部門や地域による業績差異の比較を容易に実施できるなど、既存データベース資産の用途を拡大することが可能となる。

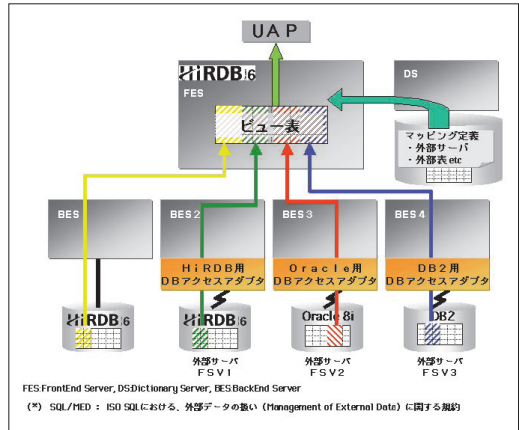
■外部DBへのアクセス

一般に社内の様々な情報が異なるDBMSで管理されているケースは多く、業務ごとや部門ごとといったデータを個別に求めるだけのシステムが複数構築されている。これを統合しようとする、データ型や文字コードに始まりメタデータや用語の統一など、様々な問題が浮上してくる。それらの問題を解決するだけでもコストと時間が発生し、ようやく統合システムが稼働する頃には予算をかなりの勢いで消費していたということにもなりかねない。

こうした問題を解決するのがHiRDB Version 6で提供される新製品「HiRDB External Data Access」である。HiRDB External Data Accessは、このような異なるDBMSが管理する外部のデータベース表を1つのDBMSに移行するのではなく、それらの表を基表としたグローバルな仮想的な表（外部表）を定義し、そのビュー表を通して各DBMSのデータにアクセスすることを可能とする。これは、ISO SQLにおける外部データの扱いを規定したISO SQL Part9（SQL/MED：Management of External Data）の規約に基づいて実装されている。

ユーザーアプリケーションプログラム（以下、UAP）

図3-10：HiRDB External Data Access



はHiRDB上の外部表にアクセスするだけで、自動的にHiRDB、DB2、Oracleなど異なるDBMSで管理される表のデータを参照することになる。したがって、各DBMSからの抽出結果を別のデータベースに保持する従来の方法とは異なり、参照している情報は常に新鮮であることに加え、業務をまたがったデータを容易に参照することができる。

HiRDBでは、パラレルサーバを利用して外部サーバとの連携を行う。基本的には、パラレルサーバを構成するBESを外部サーバとの連携に利用するものである。以下に、具体的に外部サーバと連携する手順を追いかけてみる。

次のSQL文は外部サーバを定義するものである。

```
CREATE SERVER FSV1
TYPE 'HIRDB'
VERSION '6.0'
OPTIONS (USING_BES 'BES2')
```

次に、各外部サーバとのインタフェースを行うユーザーのマッピングを定義する。

```
CREATE USER MAPPING FOR PUBLIC
SERVER FSV1
OPTIONS
(USER 'taro',
PASSWD 'taro')
```

HiRDB内ではPUBLICとして扱い、実際に外部サーバであるFSV1にアクセスする際のユーザーIDを定義したのである。

この後、実際に使用する外部表を定義する。

```
CREATE FOREIGN TABLE ZAIKO
(SCODE CHAR(4),
SNAME NCHAR(10),
ZSURYO INTEGER)
SERVER FSV1
OPTIONS (TABLE 'A-ZAIKO');
```

以上の定義により、UAPは外部表（ZAIKO）にアクセスすると、自動的に外部サーバ上の表（A-ZAIKO）にアクセスできることになる。いわば、外部表は一種のビュー表と言える。

この方式により、既存のDBMSのデータは一切手を加えることなく、統合したデータベースを構築することができる。そして既存システムをまたがったグローバルな検索を実施することができるようになり横断的な分析が可能となる。

3.2.3 使いやすい高機能OLAPツール

■ 多次元分析ツール「HITSENSER5」

大規模で多様なデータをより多くの視点から分析するためのツールが「HITSENSER5」である。

HITSENSER5はユーザーインターフェースを徹底的に簡略化し、シンプルなマウス操作だけで利用できる。ドリルアップ、ドリルダウン、スライス、ダウンといった分析業務も、煩雑なデータ操作を繰り返すことなく、クリックだけで実現することができる。このため、思考の流れを遮ることがなく、分析者が思いついた観点で次々とデータ分析を試みることができる。

さらに、高速な多次元分析を可能にする多次元データベースとリレーショナルデータベースの両方にシームレスにアクセスする機能を装備し、より大量データの多角的な分析も容易にでき、分析業務の幅がさらに広まる。しかも、SQLを一切意識する必要がなく、日々の売上速

図3-11：多次元分析ツール「HITSENSER5」

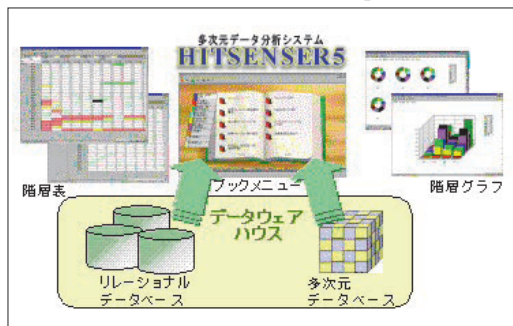


図3-12：多大なデータ表現を可能にするHITSENSER5



報のような帳票イメージの集計から、実績データを多角的に分析して要因を導き出す高度な分析業務まで、多様なデータ活用業務を同じフロントエンドから、しかもノンプログラミングで行うことができる。

HITSENSER5はデータベースの内容を、目的に応じて階層表、階層グラフおよび明細表といった形式で表またはグラフに表示できる。表の場合、セルに表示される値の集計方法を「累計値」や「構成比」に変更できる。また、グラフの場合、棒グラフ、折れ線グラフ、円グラフ、散布図、パレート図、レーダーチャートなどの形式からの選択が可能となっている。このように、目的に応じた表示形式を選択できるので、分析時に数値の新たな意味を発見しやすいのはもちろん、プレゼンテーションの場面でも説得力のある表やグラフとして利用できるのが大きな特長となっている。

3.3 | デジタルコンテンツのための技術基盤

HiRDBはVersion 5以降、数値・文字といったリレーショナルデータに加えて、テキスト・イメージ・映像・地図情報といったデジタルコンテンツも扱え、複雑かつ多様化した企業システムの構築を可能としており、それらデジタルコンテンツを管理する製品群として「HiRDB Universal Server」を用意している。ここでは、このHiRDB Universal Serverのコア技術について説明するとともに、多彩なデジタルコンテンツに対応する各種プラグイン群を紹介する。

3.3.1 HiRDB Universal Serverのコア技術

■ 抽象データ型、継承などの基本機能もサポート

オブジェクトリレーショナルデータベースであるHiRDBは、国産では初めて本格的なSQL99の規格に準拠し、抽象データ型、ユーザ定義関数などのオブジェクト指向機能をサポートした。抽象データ型とはDBで扱う対象を抽象化した概念で定義できるデータ型で、その抽象データ型のデータ操作やユーザ固有のデータ操作をユーザ定義関数として定義できる。また、それらの抽象データ型やユーザ定義関数をSQL文で利用できることになり、複雑な構造を持つデータもSQLで容易に操作可能となった。

■ デジタルコンテンツに最適の

オブジェクトリレーショナル機能

文書、画像、地図情報など企業で扱うデータは複雑かつ多様化し、それらデジタルコンテンツを活用したシステムの構築や、顧客に対する情報サービスを提供したいというニーズが高まっている。従来であれば、これらのデータはBLOB型として格納するのが一般的であった。しかしこれではコンテンツ情報を単に格納しているだけに過ぎず、コンテンツの内容自体を有効に活用しているとは言い難いものであった。これを解決する糸口となったのがオブジェクト指向である。

HiRDBは、リレーショナルデータモデルにオブジェクト指向の概念を取り込んで、オブジェクトリレーショナルデータベース管理システムに拡張した。マルチメディアデータなどの複雑な構造を持つデータと、そのデータに対する操作を一体化してオブジェクトとして扱うことができ、従来のリレーショナルデータベースと同様にSQLインタフェースによって、マルチメディアデータを

図3-13：抽象データ型、ユーザ定義関数のサポート

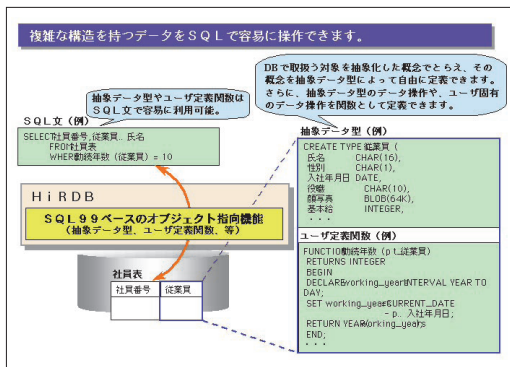
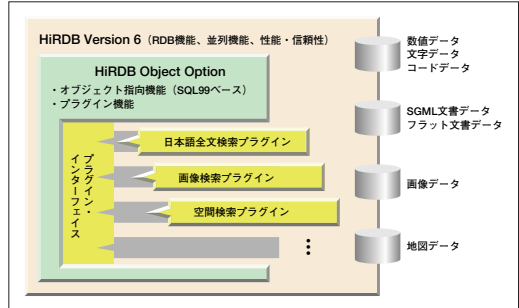


図3-14：HiRDB Universal Serverのプラグイン



容易に操作できるようになった。また、既存のRDBの資産も今まで通り利用することができ、新しいシステムにデータを移行するというような作業は一切発生しない。

さらに、映像・画像・地図情報・文書といったマルチメディアデータの定義情報とそのデータを操作する機能を、プラグインというパッケージ製品が提供しており、これらのプラグインを、HiRDB Object Optionを追加したHiRDBに登録するだけで、マルチメディアデータを簡単に扱うデータベースシステムを構築することができる。プラグインに対応したデータロードも行えるため、簡単にデータベースを構築できるのも特長の1つである。

プラグインの処理はSQLの処理と同じプロセスで実行されるため、プロセス間通信などのオーバーヘッドが発生せず、デジタルコンテンツデータを高速に処理することができるのが大きな特長である。

3.3.2 デジタルコンテンツのための強力なプラグイン群

■ 高速全文検索「HiRDB Text Search」プラグイン

このプラグインは、大量のSGML文書を正確かつ高速に検索できる機能を提供するプラグインである。SGMLのような文書の構造を定義する言語で書かれたファイルを文書として登録でき、しかも検索条件にはその文書に定義した構造を意識することができる (『全文検索』) ので、よりきめ細やかな検索が可能となる。例えば、「文書中の『見出し』として定義された構造内の文章中に“インターネット”という単語が含まれる文書を探す」といったことができる。この場合、『見出し』以外の例えば『タイトル』に“インターネット”が含まれていても検索対象から除外されることになる。

さらに、HiRDB Text Search Conceptual Extensionを組み込むことで「概念検索」を利用することも可能である。全文検索時には検索のためのキーワード (検索タ

ーム)を指定しなければならないが、概念検索においては自分の探したい情報に関係の深い文書(種文書)を検索条件として指定できるため、わざわざ検索タームを考える必要はない。このような手順で検索を繰り返せば、新たな思考や発想の手がかりを容易に見つけだすことができる。

その他にも全文検索には欠かせない「近傍条件検索」、「同義語・異表記展開検索」、「構造名を指定した検索」、「重み付き検索」、「スコア検索」などの機能も提供している。

以下に、具体的な検索例を示す。下記のようなSGML文書データが格納されているとする。

<記事>

<誌名>DB Magazine 11</誌名>

<本文>

<見出し>HiRDB</見出し>

<内容>

HiRDBはマルチメディアに…

</内容>

</本文>

</記事>

こうした文書が格納されているデータベースから、『内容』部分に“マルチメディア”を含む文書の件数は次のSQLで求めることができる。

```
SELECT count(*) FROM t1
WHERE contains(doc, '記事[本文[内容{"マルチメディア"}]]') IS TRUE;
```

■ 画像特微量検索「HiRDB Image Search」

プラグイン

このプラグインは、画像の色や形などから特微量を抽出する機能と、特微量を比較して類似する画像を検索する機能を提供するプラグインである。従来であれば、電子化された画像データそのものと、その画像を検索するためのキーワードを同時に格納する必要があったが、キーワードは個人々々のものの考え方に大きく依存するため、実際には目的の画像データをキーワードによって検索することは困難なケースが多い。HiRDB Image Searchプラグインはこうした問題を解消し、画像の色や形のイメージを条件として、その画像と「似た」画像を検索することを特長としている。以下に、具体的に特微

量検索の仕掛けや利用方法について説明する。

HiRDB Image Searchプラグインは、検索元の画像の特微量とデータベースに登録されている画像の特微量とを比較し、「類似度」と呼ぶ数値を算出する。類似度は、似た画像ほど小さくなる。例えば、完全に一致した画像では、類似度はゼロである。この算出された類似度を昇順(似ている順)に並べた結果が、特微量検索の検索結果として返却される。

検索の条件には色や形、配置のどれを重視するかも指定できるので、「こんな感じの画像」により近いものを検索することができる。例えば、画像中にある建物や山などの形状を重視して色合いについてはほとんど無視する、あるいは逆に「青空に白い雲が浮かんでいる画像」というように色合いを重視するといった指定が可能となっている。

■ データ増加による負荷を抑える外部ファイル連携「HiRDB File Link」

「HiRDB File Link」は、画像データなど大容量かつ多くのデータを任意のサーバ(ファイルサーバ)に格納し、HiRDBのデータベースには、データのリンク情報を登録する保管形態を実現するためのプラグインである。この保管形態では、データが増大してもデータベースの負荷を最小限に抑えることができ、検索性能に影響を及ぼすことがない。また、デジタル化されたデータであればどのようなフォーマットのデータでも保管できることになる。例えば、BMP形式やJPEG形式などの画像データ、ワープロソフトや表計算ソフトで作成した文書データ、あるいは動画データ、音声データなどが考えられる。

このHiRDB File Linkと、前述のHiRDB Image Searchプラグイン、およびデジタルコンテンツ出力ユーティリティ「Image Database Access」の3つの製品を組み合わせることで、WWWを基盤とした検索システムを容易に構築することができる。さらに、独自にCGIプログラムを作成することなく、用意されたCORBAオブジェクトを操作するテンプレートを作成するだけで、簡単にシステムを構築することができる。

■ 高度な空間検索を提供する

「HiRDB Spatial Search」プラグイン

このプラグインは、地図情報などの空間データ(2次元データ)を検索できる機能を提供し、地図そのものの検索や、顧客名、建物の位置を条件とした検索などを可

能にするプラグインである。従来のように顧客名称などの属性情報と空間データを別々のデータベースとして格納するのではなく、これらを一元管理することができる。

検索機能においては、例えば「駅から半径100メートル以内にある銀行を探す」のように検索範囲内に含まれるかどうかを判定したり、あるいは「ある地域内を通過する道路、線路、河川などを探す」のように検索範囲の図形に交差するかどうかを判定したりできる。こうした検索手段を駆使すれば、コンビニエンスストアやファーストフードなどの出店計画のため、利用者が集まる場所(学校、駅、大通り)を分析するシステムや、レストランガイドや観光案内をWWWを経由して地図と一緒に扱えるシステムを容易に構築することができる。後者のシステムはさらにモバイル端末や携帯電話からアクセスできるなど、幅広い利用が可能となる。

HiRDB Spatial Search プラグインでは、性能が最も重視されており、米国「オープンGISコンソーシアム」の定める Binary Geometry (SFS BG) タイプで世界初の認証を取得している。この技術は、数値変換を行わずダイレクトに図形としてアクセスし、転送するデータ量を最も小さく抑える技術であり、オープンGISコンソーシアムの定める3つのタイプのうち、最も高速なデータ表現方法となっている。

4 高性能を支える技術的バックボーンとアーキテクチャ

4.1 | 高性能を実現する高度な技術の数々

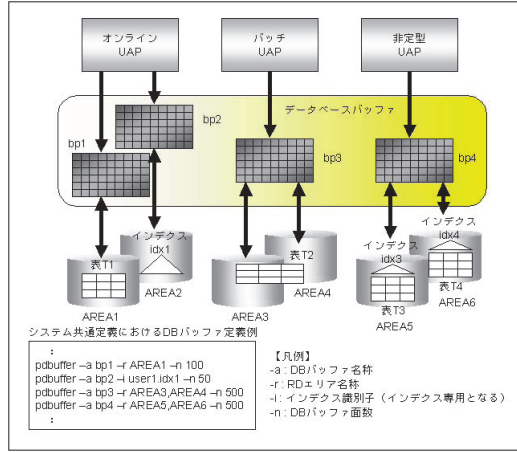
Version 6において大きな進化を遂げたHiRDB。しかし、より高い性能に対する追求は、Version 1開発当初から継続的に行われてきており、随所に高度な技術が採用されてきた。ここでは「バッファ管理機能」、「インデクス機能」、「ブロック化機能」などに着目し、HiRDBの高性能を支える基礎機能を紹介する。

4.1.1 I/O効率を向上させるバッファ管理機能

■ ディスクI/Oを削減するインディペンデント・バッファリング

一般にデータベースシステムは性能を向上させるために、ディスクのI/Oを減らすことが基本となる。このためにディスク上のデータをバッファ上で管理し、バッファヒット率を向上させることによりI/Oを減らし全体と

図4-1：インディペンデント・バッファリング



しての性能を向上させる。

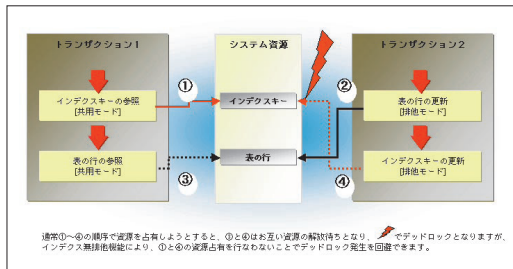
HiRDBでは、こうしたバッファ管理をきめ細かく制御することができる。例えば、表データとインデクスデータのバッファを個別に割り当てたり、アクセス頻度の高いインデクスを専用バッファに割り当てたりすることができる。こうしてバッファ間の干渉をなくすことにより、バッファの大きさを超えるような表を全件検索したような場合でも、アクセス頻度の高いインデクスがバッファ上から吐き出されることがなくなるのである。したがって、インデクス検索や全件検索などのさまざまな処理が混在する環境のもとでも、安定したレスポンスを確保することが可能である。

さらに、HiRDBではシステム用のRDエリアや、BLOB専用RDエリアにもバッファを定義することができ、様々な環境においてバッファ制御をカスタマイズできる。

■ 効率の良いI/Oと性能劣化防止を実現するデフォードライト機能

一連のデータベース操作のトランザクションを完結させるため、バッファ上の更新データを確実にデータベースに反映させるための要求命令がCOMMITである。通常、COMMITが発行されると、バッファ上の更新データはディスクへ書き出し、データベースとしての一貫性を保証する。しかし、HiRDBのデフォードライト機能では、COMMIT文が発行されてもすぐにディスクへ書き込まない。更新ページ数があらかじめ設定したある一定の値に達した時点でバックグラウンドにて書き込みを行う。更新ページ数がある一定の値 (HiRDBが決定する

図4-2：インデクス無排他機能



値)に達した時点ではデフォードライトトリガという。

デフォードライト時にディスクへ書き込む更新ページ数は、システム共通定義のpdbufferオペランドの-wオプションで指定することができ、デフォードライトトリガでの更新ページの出力比率を基にHiRDBが決定するようになっている。このデフォードライト機能により、COMMIT文の発行時点でのディスクへの書き込みを抑制し、同一ページへの更新が多発するような業務においては特に効率良くI/Oを制御、性能劣化を防ぐことができる。

4.1.2 高度なアクセス性能を提供するインデクス機能

■ デッドロックを解消するインデクス無排他機能

HiRDBはキー値レベルの排他制御を採用し、同時実行性を向上させていた。しかし、さらに向上させようするとインデクスのキー値と行との間でのデッドロックが発生しやすくなり、性能のボトルネックとなることが多かった。そこでHiRDB Version 5.0において、インデクスの無排他機能が実装された。

このインデクス無排他機能は、インデクス・アクセスにおいてはキー値排他を取得しないため、図に示したように更新業務と検索業務との間でのデッドロックが発生しなくなる。これによりアプリケーションの同時実行性を一段と飛躍させることが可能となった。

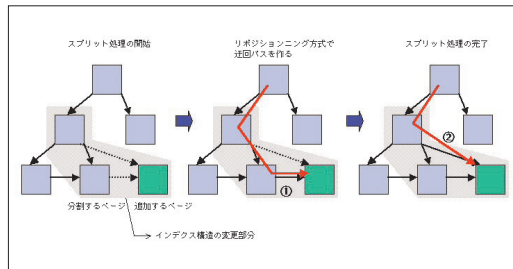
■ 同時実行性を向上させる

リロケータブル・インデクス機能

HiRDBでは更新中のインデクスに対するアクセス能力向上のため、リロケータブル・インデクス制御を行っている。

HiRDBにおけるインデクスはB木構造をしており、インデクスキーを追加する場合にインデクスページ内の空き領域不足で書き込みができなくなると、スプリット処

図4-3：リロケータブル・インデクス機能



理を行う。スプリット処理では、新しく割り当てたページにインデクスの検索パスを付け直す。追加するページへの検索パスは図4-3に示した①、②の順に作る。①のパスが完成した時点で追加したページの検索を許可する。こうするとスプリット前のページを参照し、検索していても求めるデータに正しくアクセスできる。これを「リポジショニング方式」と呼ぶ。これにより、トランザクションでインデクスを利用することができるのである。たとえインデクススプリットを行ったトランザクションがスプリット完成後にROLLBACKしてもインデクスは元の状態に戻さず、そのままスプリットを完成させるようにしてある。このように、リポジショニング方式を採用したインデクスにより、多くの検索・更新のトランザクションが発生するシステムにおいても、高い同時実行性が得られる。

4.1.3 大量データアクセスに有効なブロック化機能

■ 列数に依存しない安定した性能を維持する行インタフェース

一般的にRDBMSでは、列単位でデータを操作する列インタフェースが基本となっており、列数が非常に多く

図4-4：行インタフェースによる列オーバヘッドの解消

