

3W-6

電子ファイリングシステムにおける
データベース照会方法

鈴木定男 (株式会社 東芝)

1. はじめに

近年、光ディスクを記憶媒体として、イメージで構成された書類を管理するファイリングシステムが急速に普及し、オフィスのペーパーレス化に貢献している。このようなファイリングシステムでは、イメージデータを効率よくアクセスするため、データベースを使って書類を管理する方法が主に採用されている。

DP上で動く電子ファイリングシステムも、リレーショナルデータベースを使用して、書類を階層的に管理している。本データベースは汎用的に他のアプリケーションからも参照できるように、照会を実行する処理が、照会中枢と呼ばれる1つのプロセスとして用意されている。一方、電子ファイリングシステムのように、同一のリレーションをいつも類似した形で参照するような場合には、アクセス速度を上げるため、直接データベースをアクセスすることもできる。

電子ファイリングシステムで使用しているデータベースの構造と参照方法について報告する。

2. データベースの構成

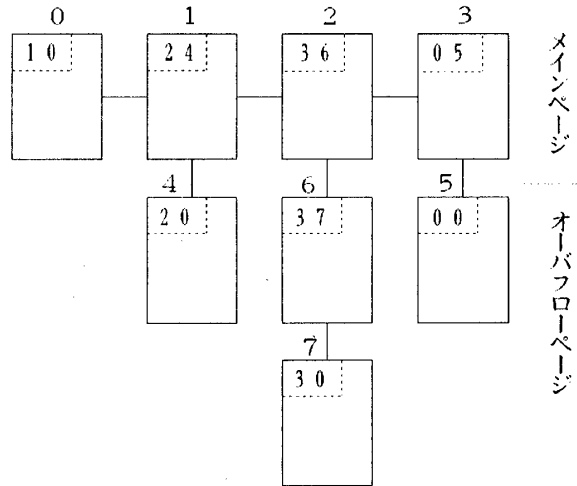
(1) ファイル構造

電子ファイリングシステムで使用しているデータベースは、UNIXライクなファイル上に作られている。1つのデータベースが、1つのディレクトリの下に形成されており、1つのリレーションが、1つのファイルとして作られている。

(2) 格納構造

リレーションは、4Kbyteに区切られたページにより構成されている。総ページ数は、各リレーションによって異なっている。各ページに複数のタプルが格納されているが、どのページに格納されるかは、タプルのキーの値と総ページ数をもとに、ハッシュングすることによって決定される。該当するページが一杯になり、格納できない場合は、オーバフローページに格納される。各ページには、次に示す情報が格納されているヘッダがついており、各ページがチェーン状につながっている。

- メインページ … 次のメインページの番号
- オーバフローページ … 次のオーバフローページの番号
- ライン番号 … 次に使用可能なライン番号



ページは0ページから始まり、最終ページのメインページとオーバフローページの値は、0になっている。各ページの最終部には、ラインテーブルが設定されている。これは各タプルの格納位置を、ページ内のオフセットで示したものである。タプルの個数が増えるにつれて、後からテーブルが伸びてくる形になっている。ヘッダ部のライン番号によって、次に使用可能なセルを知ることができる。

- このことから、目的のタプルを参照する手順は次のようになる。
- I. 該当リレーションをオープンする。
- II. 条件にキーが指定されていれば、ハッシュングにより格納ページを求める。キーが指定されていないければ、0ページからすべてのページを操作することになる。
- III. 各ページのヘッダを読み込み、ライン番号から現在使用されている最終ラインテーブルを調べ、そのページに格納されているタプル数を求める。
- IV. 該当ページ内のすべてのタプルについて処理が完了したら、オーバフローページ、次のメインページに進み処理を繰り返す。

3. ハッシュング

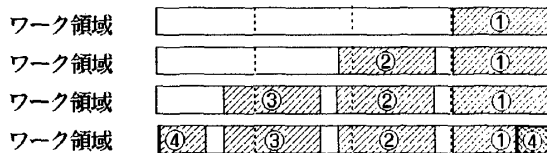
ハッシュングとは、タプルの格納先を決定する手法である。タプルの中のある属性をキーとして、その値をハッシュ関数に与えると格納先が求められる。つまり、キー値とページ番号がハッシュ

関数によって、1対1に対応づけられる。ここで重要なのは、ランダムな値を取るキーに対して、そこから求まるページ番号が、片よりがなく一様なチラバリを持たなければいけないということである。ページからタプルを検索して求めるときに、格納されているタプル数の少ない方が速く見つかるわけで、この意味から各ページに平均してタプルが入っている必要がある。

(1) ハッシュ関数

電子ファイリングシステムで使用しているハッシュ関数が、どのような方法で、ページ番号を決定しているかを簡単に述べる。

ワーク領域を用意し、そこにキーの値を1byteずつ、ある一定の桁ずつ左にシフトしていきながら、格納していく。シフトが限界に達したときは、ワーク領域の下ビットに戻って格納され、既に入っているビットとの排他的論理和がとられる。格納が終了したら、ワーク領域をページ数で割り、その余りがハッシュ値となる。



(2) 改善

ここでハッシュ値の発生状況を調べるため、次のようなケースを例にとって考える。

キー値 … 2byte, ページ数 … 4ページ

ワーク領域 … 4byte

このとき発生するハッシュ値は次のようになる。

キー値	ワーク領域	ハッシュ値
0000H~00FFH	0000??00H	0
0100H~01FFH	0000??01H	1
0200H~02FFH	0000??02H	2
0300H~03FFH	0000??03H	3
0400H~04FFH	0000??04H	0

この例ではキーがランダムな値をとれば、一様なバラツキが得られるが、sequentialな値をとると、最終的には一樣になるが、その過程として一樣とはいえない。また、ワーク領域4byteに対して、キー長が2byteのため実質的に排他的論理和が発生せず、ワーク領域の値が単純化してしまっている。

さらに、総ページ数も関係している。この例のように最終的に4で割るということは、16進数の下1桁の値でハッシュ値が決まることになる。一般に2, 4, 8, 16のようなページ数だと下数桁でハッシュ値が左右されることになり好ましくない。それよりも約数を持たない数がページ数として適当である。

したがって、当ハッシュが有効に使用されるためには次の改善が考えられる。

- A. キーの総byte数が、ワーク領域の長さを越えるようにする。
- B. キーとして、ランダムな値が発生するものを選択する。
- C. ページ数を素数にする。

4. 参照方法

データベースを参照するには、次の2つの方法がある。

(1) 照会中樞

データベースのリレーションからタプルを取り出したり変更するといった処理要求を実現させるには、照会中樞に対して、照会文を送信しなければならない。つまり、1つのデータベースに対して、照会中樞を司る1つのプロセスがバックステージパーティションで動作しており、このプロセスと交信することでデータベースをアクセスすることになる。

照会文はSQLに類似した形をしており、照会中樞はまず照会文の文法解析を行って、要求を実行することになる。また、1つのデータベースには、システム表と呼ばれるリレーションがいくつか存在している。これはデータベース内のリレーションやその属性等を管理しているリレーションであり、照会文を実行すると、まずこのシステム表が参照され、ターゲットとなっているリレーションの種々の情報を得たうえで、処理が実行される。

(2) 直接操作

(1)の方法では次の処理に時間がかかる。

- A. プロセス間通信
- B. SQLの文法解析
- C. システムテーブルの参照

そこでスピード向上に重点をおくときは、照会中樞を介さず、直接リレーションを参照することができる。各リレーションは1つのファイルと見なせるので、直接オープンして読み書きが可能である。ただし挿入・削除ではタプルの個数が変化するため、システムテーブルも更新する必要がある。よって当アクセス方法は、取り出し・更新のみ可能である。

このとき、照会文は次のように簡単に定式化することで、文法解析の時間を短縮している。

```
SELECT 照会項目リスト FROM 表名 WHERE 検索条件
```

↓
リレーション識別子、照会タイプ、条件

5. おわりに

リレショナル型データベースでは、構造化照会言語による照会方法が一般的である。しかし、性能面からみて得策でない場合が多い。今回の場合は構造・量が限定された場合での直接照会方法を考案した。

データベースは電子ファイリングシステムに限らず、他の多くのシステムで利用されている。さらに活用の幅を広げるためには、各場面に則した照会方法・格納構造等がとれるような柔軟な設計をする必要がある。