4F-3

多次元クラスタリングに基づくGRACEの二次記憶系の設計

伏見 信也　喜連川 優　田中 英彦　元岡 達

(東京大学　工学部)

0. はじめに

現在までに提案、試作されてきた関係データベースマシンでは、joinやprojection等の処理負荷の重い演算の高速化に主眼がおかれ、二次記憶系に関しては、その多くがインデックス等の補助データ構造を持たず、リレーションのフルスキャンを前提にしていた。このことは単純なselection 処理等に於けるマシンの大幅な性能低下を予想させ、join等の高速化が既に達成されているGRACEでは、この点の強化を試みている。

本稿では、その中でページ化されて二次記憶系に格納されているリレーションへのアクセスについて、特に多次元クラスタリングに基づくアクセスページ数の最小化問題について考察する。

1. 多次元クラスタリングの概念

従来の二次記憶系設計の手法は、リレーションのキー属性についてのみハッシュ、インデックス等を用いてページ分割を行なうものが多く、従って非キー属性による検索に関しては基本的に全ページのフルスキャンを必要とする。一方で、非キー属性も含めて複数属性によりページ分割を行なえば、非キー属性による検索に関してもアクセスページ数は減少し、結果として検索全体に対する平均アクセスページ数は大幅に減少する。このように、多数の属性を用いたページ分割により、selection のpredicate を満足するレコードがページ空間に広く分散しない様制御することを二次記憶系の多次元クラスタリングと呼ぶ。

2. コスト評価

効果的な多次元クラスタリングを行なうには、問い合わせの分布を考慮にいれたページ分割を行なう必要がある。即ち、データベース分布から生じるページ分割値に対する制約の下で、一定の分布を持った問い合わせの空間Qに関するアクセスページ数 f の平均値を最小化することが必要となる。リレーションR＝R($A_1,\ldots,A_k$)に対し多次元クラスタリングを行なった時、Rへのアクセスページ数の平均値 fは次の式で与えられる[1]。

$$\bar{f} = \sum_{j=1}^{n} \int_{c \in W(p_j)} q(c)\, dc$$

$$(W(p_j) = \{c \in Q \mid c \cap p_j \neq \emptyset\})$$

ここに nはページ数、$p_j = \prod_{i=1}^{k} [\alpha_{ij}, \beta_{ij}]$ は k次元直方体に分割されたページ、$c = \prod_{i=1}^{k} [x_i, y_i]$ は一般にrange を指定するselection predicate

$$x_1 \leqq v_1 \leqq y_1 \text{ and } \ldots \text{and } x_k \leqq v_k \leqq y_k$$

であり、更に q(c) はその正規化された分布である。この定式化により、多次元クラスタリングの問題は、コスト関数 $\bar{f}$を最小化するページ分割 $\{p_j\}$ を求める組み合わせ最適化問題に帰着する。

3. ページ分割アルゴリズム

問い合わせ分布 q(c) が与えられた時、アクセスページ数を減少させる基本的な方策は、アクセス頻度の高い部分を出来る限り１ページに収めることである。そこで $\bar{f}$を最小化する近似アルゴリズムとして次の様な手続きを考える。

```
procedure partition(k-dim_space,page_num);
/* "k-dim_space" is divided into "page_num" pages.      */
/* k-dim_space[i] is of the form (low_value,high_value) */
begin
  if(page_num == 1) then begin
    Allocate leaf node, and process it appropriately;
    return;
  end;

  minimun_hit_query = sufficient large value;
  for each attribute i do begin
    (low,high) = k-dim_space[i];     /* save range */
    for j=1 to page_num-1 do begin
      /* fetch next partition candidate */
      x = find_candidate_value(k-dim_space[i],page_num);
      k-dim_space[i] = (low,x);
      hit_query = compute_hit_query(k-dim_space);
      k-dim_space[i] = (x,high);
      hit_query = hit_query + compute_hit_query(k-dim_space);
      k-dim_space[i] = (low,high); /* restore range */
      if(hit_query<minimum_hit_query) then begin
        minimum_hit_query = hit_query;
        partition_attribute = i;
        partition_value     = x;
        partition_page_num  = j;
      end;
    end;
  end;

  Allocate node and arrange links;
  node.partition_attribute = partition_attribute;
  node.partition_value     = partition_value;

  (low,high) = k-dim_space[partition_attribute];
  k-dim_space[partition_attribute] = (low,partition_value);
  partition(k-dim_space,partition_page_num);

  k-dim_space[partition_attribute] = (partition_value,high);
  partition(k-dim_space,page_num-partition_page_num);
end;
```

即ち、増加するアクセスページ数が最小の分割値を選びながら再帰的にページ分割を行なう。このアル

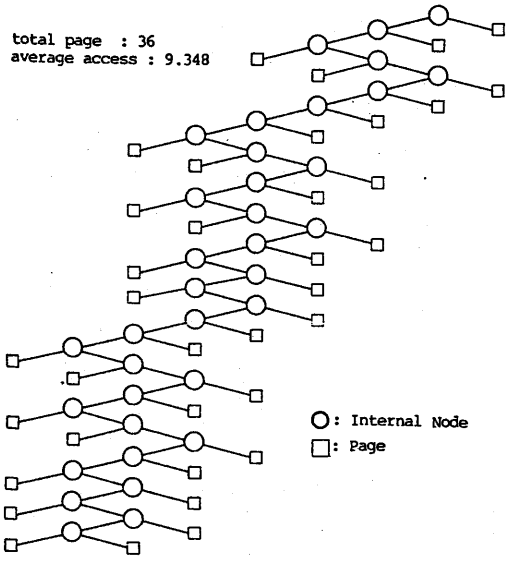
ゴリズムにより生成されるインデックスはk-d tree [2]の一般化された形になっている。

4．アルゴリズムの評価

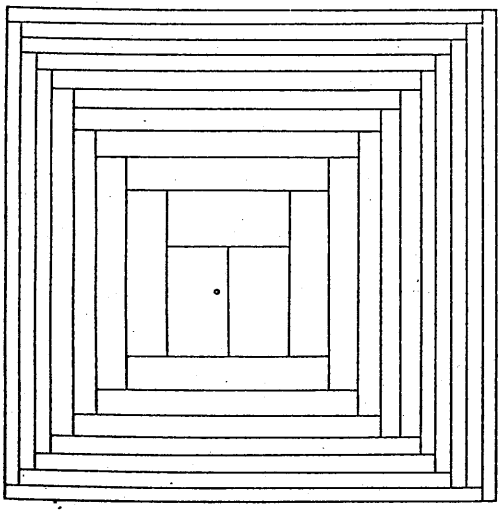
3．で述べたアルゴリズムにより生成されたインデックス及びページ分割の例を図1(a)，(b)に示す。ここにデータベースは一様に分布しており、一方、問い合わせの分布は図1(b)中○の点をそのpeakに放射状に広がるものである。

図から明らかな様に、ページ分割は問い合わせ分布のpeak、即ちアクセスの頻繁な部分を横切らない様に行なわれており、問い合わせの分布が充分反映されていることがわかる。

尚、このように問い合わせ分布に偏りがある場合には、インデックスはほぼ一次元状に成長しバランスはしない。



(a) Multidimensional Binary Index Corresponding to the Page Partition of Fig.1(b)



(b) Multidimensional Page Partition for Query Distribution with biased peak

Fig.1 An Example of Multidimensional Page Partition generated by "partition"

5．問い合わせ分布の為のデータ構造

3．で述べたアルゴリズムを実装するには、問い合わせ分布q(c)をデータとして保持する必要がある。しかし、selection predicate をそのまま記憶していたのでは必要とされる記憶容量は膨大なものとなり現実的ではない。

一般にページアクセス数の増加に大きく寄与するのは分布q(c)のpeakの近傍であり、peakをはずれた分布情報は適度に縮退させることが出来る。そこで、一定の記憶容量内に問い合わせとその出現頻度を保持し、容量を溢れた場合には頻度の低い問い合わせの値の下位のbitを順にdon't care bit とすることにより、空領域を確保するデータ構造を採用した。図2に縮退された分布情報の一例を示す。

500 Data Items
Buffer Capacity = 16 Data Items

Gradated Statistics (val : gradated value, gb : gradated bit)

val gb	val gb	val gb	val gb	count
1fff[13]	3ff[10]	7f[0]	3ff[10]	29
1ff[9]	3f[5]	1f[4]	7f[7]	23
1ff[9]	1f[5]	3f[5]	7f[7]	31
3ff[10]	ff[8]	17[2]	7f[7]	44
3ff[10]	1fff[13]	7f[7]	3f[0]	27
1f[4]	ff[8]	1f[5]	1f[5]	45
7f[6]	f[4]	7f[7]	1f[5]	7
3fff[14]	fff[11]	7f[5]	fff[12]	4
3f[5]	7f[7]	f[4]	7[3]	22
ffff[16]	1ffff[17]	ff[8]	1ff[8]	15
f[4]	1[1]	1f[5]	7[2]	58
3f[6]	3f[6]	f[3]	1f[5]	59
1ff[9]	ff[8]	1f[5]	f[3]	26
7ff[10]	1ff[9]	7f[7]	ff[8]	25
f[4]	1[1]	1f[5]	3[2]	85

Fig.2 An Example of Gradated Statistics

6．おわりに

多次元クラスタリングに基づく平均ページアクセス数の最小化の問題に関し、問い合わせ分布を考慮したページ分割のアルゴリズムについて考察した。トランスポーズドファイルとの関係等については別稿に譲る。

［参考文献］

［1］伏見他：「GRACEに於ける二次記憶系の構成」 第25回情報処理全国大会予稿集 1982

［2］Bentley, J. L.,: 「Multidimensional Binary Search Trees Used for Associative Searching」 CACM, no.9,1975