

リレーショナル・データベースマシンにおける 検索用入力バッファの最適構成†

速 水 治 夫††

リレーショナル・データベースマシン RINDA は、リレーショナル・データベースに対するインデックスの利用が困難な非定型的検索処理を高速化することを目的とした汎用計算機への付加プロセッサである。RINDA の構成要素である内容検索プロセッサは汎用計算機と、データベースが格納される汎用ディスク装置との間に入出力インタフェースで接続され、ディスク装置からデータベースを読み出して内容検索を実行し、検索結果を汎用計算機へ転送する。内容検索プロセッサは汎用ディスク装置の性能を最大限に生かして内容検索を実行するため、シリンダ単位のマルチトラックリードでデータ読み出しを行い、物理レコード単位でデータ読み出しと内容検索のパイプライン処理を行う。検索用入力バッファの設計にあたっては、データ読み出しと内容検索のパイプライン処理をモデル化し、トラック内物理レコード数と入力バッファ面数をパラメータとして、ディスク装置の回転待ちオーバーヘッドを評価し、トラック内レコード数の2倍の入力バッファ数を循環的に使用すれば、回転待ちオーバーヘッドを回避できることを明らかにした。これに基づき、入力バッファはバンク数が増え可能なメモリで構成し、検索時のトラック内物理レコード数に応じてバンク数を切り替える方式とした。性能の実測評価により、本入力バッファ構成で回転待ちオーバーヘッドを回避していることを明らかにした。

1. はじめに

リレーショナル・データベース (RDB) では、データベースの論理構造が単純な表形式であり、その表 (リレーション) の任意の行 (タプル) と列 (アトリビュート) の集合に対してデータ操作を行うことができる。そのためデータ操作は、ISO および JIS で標準化されたデータベース言語 SQL^{1),2)} に見られるように高水準な言語インタフェースとなっている。

また、RDB が提案されて以来、RDB 処理を高速化するデータベースマシンの開発が盛んに行われている^{3),4)}。これは、RDB のデータ操作の水準が高く、ソフトウェアのみの実現では高速化に限度があり、近年の半導体を始めとするハードウェアテクノロジーの進歩を背景として、専用ハードウェアの支援による高速化が有効となることによる。

これらのアーキテクチャの中に、ディスク装置と汎用計算機との間の I/O ボトルネックの解消を目的としたフィルタプロセッサ (内容検索プロセッサ) を使用する方式も提案されている^{5),6)}。著者らも内容検索プロセッサを構成要素の一つとするデータベースマシン (RINDA) を開発した^{7),8)}。

内容検索プロセッサは、RDB に対するインデックスの利用が困難な非定型的検索処理を高速化する

ことを目的とし、汎用計算機 DIPS シリーズ^{9),10)} と入出力インタフェースで接続され、DIPS 上のソフトウェアであるデータベース管理システムによって制御される。ホスト計算機およびデータベースが格納されるディスク装置は汎用のものを使用するため、既存ユーザの設備を無駄にすることなく内容検索プロセッサを導入することが可能である。

内容検索プロセッサは汎用ディスク装置の性能を最大限に生かすため、シリンダ単位のマルチトラックリードでデータ読み出しを行い、物理レコード単位でデータ読み出しと内容検索のパイプライン処理を行っている。ディスク装置は回転している記録媒体からデータを読み出すため、検索処理の遅れに起因したパイプライン処理の乱れにより読み損ねた物理レコードを再度読み出すまでに回転待ち時間が必要となり、検索処理の遅れ以上のオーバーヘッドとなる可能性がある。このオーバーヘッドを回避するために、内容検索プロセッサでは物理レコード容量に応じて入力バッファ面数とバッファ容量とをサーチ処理ごとに切り替えて使用している。

本論文では、ディスク装置からのデータ読み出しと内容検索のパイプライン処理をモデル化し、トラック内物理レコード数と入力バッファ面数をパラメータとして、回転待ちオーバーヘッドを評価する手法による入力バッファの最適構成法を述べる。2章で内容検索プロセッサの概要を示し、3章でディスク装置アクセス法、4章でモデル化による入力バッファ構成の設計法

† An Optimal Configuration of Search Buffer in Relational Database Machine by HARUO HAYAMI (NTT Communications and Information Processing Laboratories).

†† NTT 情報通信処理研究所

を述べ、5章で性能の実測評価により提案方式の有効性を示す。

2. 内容検索プロセッサの概要

2.1 RINDA の全体構成

RDB の主要な処理は、定型の検索・更新処理と非定型の検索処理である。定型の検索・更新処理ではインデックスを使用することにより、ソフトウェアのみでも十分に実用的な性能が実現されている。しかし、非定型の検索処理ではインデックスの利用が困難なため、表中のすべての行を読み出して検索条件の判定を行うサーチ処理と、表を特定の列の値の順序で並び換えるソート処理の負荷が大きく、多大の処理時間がかかっていた¹⁾。

上記の問題を解決するために開発された RINDA では、サーチ処理とソート処理を内容検索プロセッサ (CSP: Content Search Processor) と関係演算プロセッサ (ROP: Relational Operation Accelerating Processor) で超高速に実行する。これにより、汎用計算機では実時間で処理が困難であった大規模データベースに対する非定型検索処理を最大 100 倍以上高速化することを可能にした。CSP は、ディスク装置 (DK) に格納された表をサーチし、検索条件に合致する行の列を選択してホスト計算機に転送する。ROP はホスト計算機から転送されてくる表をソートし、その結果をホスト計算機に送り返す。CSP, ROP が実現している主な機能をそれぞれ表 1、表 2 に示す。ホスト計算機に RINDA を接続したシステム構成例を図 1 に示す。CSP, ROP はそれぞれ独立の入出力インタフェースでホスト計算機に接続される。CSP とディスク制御装置 (DKC) も入出力インタフェースで接続される。CSP と ROP を独立のハードウェアとしたことにより、システムごとに異なるサーチ処理とソート処理の負荷バランスに柔軟に対応することを可能とした。

2.2 CSP の動作概要

CSP のサーチ処理は以下の手順で実行される。

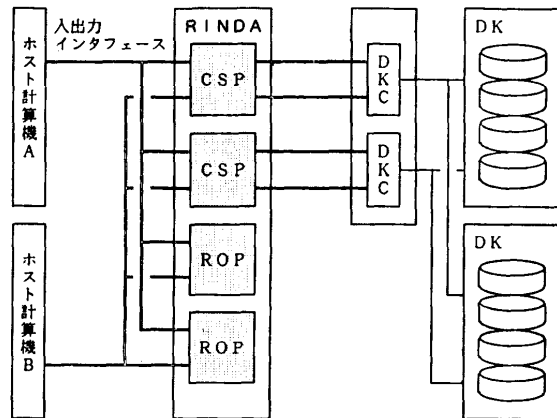
① ホスト計算機のデータベース管理システム (DBMS) が CSP に対するオーダを作成する。オーダ中には、サーチ対象の DK の入出力アドレス、物理レコード容量、物理レコードアドレスなどの物理制御情報と、表 1 の機能を記述した論理制御情報が含まれている。物理レコードは DBMS の扱うページに対応しており、ページ中にはサーチ対象の表の複数の行が

表 1 CSP の主要機能
Table 1 Primary CSP functions.

機能	説明
述語判定	比較述語 <列> <比較演算子><定数>の条件判定.
	IN 述語 <列> [NOT] IN<定数リスト>の条件判定.
	LIKE 述語 <列> [NOT] LIKE <文字列>の条件判定.
	NULL 述語 <列> IS [NOT] NULL の条件判定.
検索条件判定	述語の AND/OR による任意の論理式判定.
出力列の抽出	上記検索条件を満たす行から任意の列の出力.
集合関数演算	上記検索条件を満たす行数の計数 (COUNT(*)).

表 2 ROP の主要機能
Table 2 Primary ROP functions.

機能	説明
ソート	指定された列の値 (ソートキーと呼ぶ) による昇順または降順への行の並び換え。ソートキーは任意の順序の複数列により構成可能.
ふるい落とし	ソートキーの値による二つの表から結合可能性のない行の除去。一方または両方の表から除去が可能.
重複計数/除去	ソートキーの値が重複する行数の計数、および 2 番目以降の行の除去.



CSP: 内容検索プロセッサ, DKC: ディスク制御装置
ROP: 関係演算プロセッサ, DK: ディスク装置

図 1 RINDA システムの構成例
Fig. 1 Typical RINDA system organization.

格納されている。以下では物理レコードを、レコードと略す。

② 作成したオーダをチャンネル・コマンドによって

ホスト計算機から CSP へ転送する。

③ CSP は受け取ったオーダをもとに DK に対しチャンネル・コマンドを発行し、サーチ対象のレコードをマルチトラックリードにより連続的に読み出す。同時に検索条件の判定と出力列の抽出を実行する。この動作を間断なく行うために、DK から読み出したレコードを格納する入力バッファと検索条件合致行の出力列をホスト計算機に転送するための出力バッファは複数個を交代で使用する。

④ 出力バッファ中に検索結果のデータがページ単位にたまった段階で、サーチ処理とは非同期に検索結果をホスト計算機に転送する。

3. ディスク装置アクセス方式

DK の読み出し法には、①1レコード単位の読み出しと、②連続した複数のレコードを連続的に読み出すマルチトラックリードがある。

①の方法は読み出しごとに、対象トラックへのヘッド移動（シーク）とトラック内でのレコード先頭位置検出（ヘッドサーチ）が必要である。連続した複数のレコードを読み出すときに①の方法を使用すると、シークは1回でよいが、ヘッドサーチがレコードごとに必要となりレコード転送時間以上の時間を要する。ただし、複数ユーザが競合して使用する環境ではシークもレコードごとに必要となる。

他方②の方法は、シーク・ヘッドサーチが1回で後はレコード転送が連続するため、読み出し時間が短縮される。

汎用 OS の制御下では、チャンネルと DKC の使用率を低く抑えるために、最大1トラック程度のマルチトラックリードのみが使用されている。ただし、DBMS の場合は OS のファイル管理システムが提供するレコードアクセス法を使用しているためマルチトラックリードは使用されていない。

CSP では、DK のデータ転送性能を最大限に生かすため、1シリンダ連続のマルチトラックリードを使用している。本処理を可能とする、入力バッファの構成については次章で述べる。

1シリンダ連続のマルチトラックリードによる性能向上効果を I/O 時間短縮率 ($1/E_M$) で評価すると、 E_M は次のように表される。

$$E_M = T_i / T_M \\ = (1/2 + C_R / C_T) \cdot N_R \quad (1)$$

T_i : 1レコードリードの繰り返しによる1シリン

ダ読み出し時間

T_M : マルチトラックリードによる1シリンダ読み出し時間

C_T : トラック容量

C_R : レコード容量

N_R : トラック内レコード数

ただし、DK を1ユーザで使用している場合を想定している。複数ユーザの場合はシーク時間増によって比率は増加する。

トラックの格納効率が良いうように C_R を設定したとすると、

$$N_R \doteq C_T / C_R$$

であり、本システムで使用されている各種 DK のトラック容量と DBMS で使用するレコード（ページ）容量から、 N_R の採りうる値の範囲は 1~16 である。この範囲で、

$$E_M = 1.5 \sim 9$$

である。

4. 入力バッファの最適構成

連続的なレコードの読み出しと、レコード内データの内容検索を、レコード単位でパイプライン処理するために、レコードを格納する入力バッファは最低2面必要である。1レコード内データの内容検索の処理時間が、常に1レコードの転送（読み出し）時間より短ければ、入力バッファは2面で十分である。すべての条件で上記を満たすように内容検索部を構成することは過剰設計となる。検索時間が転送時間より長くなり、入力バッファが空かずに、次のレコードを入力できなかった時、DK は回転しているため、同一レコードの再読み出しは、早くて1回転後である。このため、検索時間が転送時間よりわずかに増加したことによる実効的な検索時間の増分は、回転待ち時間のために検索時間の増分自体より長くなる。この増分を回転待ちオーバーヘッドと呼ぶ。

以下に、回転待ちオーバーヘッドを回避する最適な入力バッファ構成を述べる。

4.1 入力バッファの最適面数

(1) 設計条件のモデル化

① DK

- 1シリンダ内の各トラックは N_R 等分した容量のレコードで構成される。内容検索は1シリンダを単位として実行する。 N_R は1以上の2の冪乗数とする。

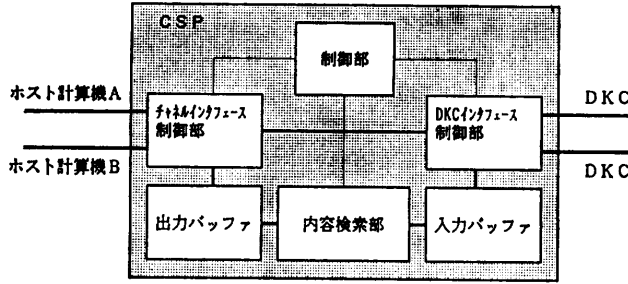


図2 CSPのハードウェア構成モデル
Fig. 2 CSP hardware configuration.

- DKの回転時間を T_R とする。
- レコード転送時間を T_T とすると、
$$T_T = T_R / N_R$$
である。
- ② CSP
 - CSPの内部構成モデルを図2に示す。
 - DKから転送されるレコードを入力し、内容検索部による検索処理が終了するまで保持している入力バッファは N_B 個であり、循環的に使用する。 N_B は2以上の2の冪乗数とする。
 - レコードの転送は連続的に行われる。入力すべき入力バッファに検索未終了のレコードがある場合は、レコードの読み出しは中断し、回転待ち時間 (T_R の整数倍) 後に読み出しを再開する。
 - 内容検索部における検索処理は入力バッファに検索対象のレコードが入った時点で開始する。ただし、複数レコードの並列検索はできない。(転送処理は一定時間間隔で開始され、検索処理はデータリブで開始される。)
 - 内容検索部における1レコードの検索時間を T_s とする。すべてのレコードにおいて、 T_s は一定とする。
 - レコード転送も含めた、1レコード当たりの実効検索時間を T とする。 T の計算において、1シリンダ連続サーチの定常状態における検索時間を評価することとし、最初の1レコードの転送時間は無視する。

(2) 設計法

上記設計モデルにおけるサーチ処理のタイムチャート例を図3に示す。

図3から、以下がわかる。

(i) $T_s \leq T_T$ の場合は、回転待ちは発生せず、 $T = T_T$ となる。

(ii) $T_s > T_T$ の場合は、回転待ちは発生するが、 N_R と N_B の関係によって、検索中断時間が発生しない場合と発生する場合がある。

① 検索中断時間が発生しない場合は、

$$T = T_s \text{ となる。}$$

② 検索中断時間が発生する場合は、

$$T > T_s \text{ となる。}$$

$(T - T_s)$ が回転待ちオーバーヘッドである。

前述のように、 N_R の採りうる値の範囲は1~16である。この範囲において、 N_B をパラメータとして、レコード転送時間で正規化した検索時間 (T_s/T_T) と実効検索時間 (T/T_T) の関係を、タイムチャートの解析により求めると図4となる。

図4の実効検索時間は次式で表すことができる。

$$T = \max(T_s, T_T) + T_R \cdot \alpha \cdot \beta$$

$$T/T_T = \max(T_s/T_T, 1) + N_R \cdot \alpha \cdot \beta$$

α : 回転待発生頻度 ($0 \leq \alpha$)

β : 検索待発生係数 ($0 \leq \beta < 1$)

α , β の値は、 T_s , N_R , N_B によって定まる。

各 N_R において、 $\beta = 0$ となる、最小の N_B が最適であり、それを表3に示す。

また、検索時間がレコード転送時間よりわずかに長い場合に (例えば、 $T_s/T_T = 1.1$ の場合に)、実効検索時間が回転待ちオーバーヘッドにより増加する比率 (T/T_s) を求めると図5となる。

図5からわかるように、 N_B が表3の値より少ない場合の実効検索時間の増加は N_R が大きい場合ほど著しい。

4.2 入力バッファ構成

入力バッファ1面の容量 (C_B) を一定とすると、その容量は最大レコード容量、すなわちトラック容量 (C_T) に等しくする必要はあるが、 C_T のうち使用されるのはサーチ対象レコードの容量 (C_R) のみである。このため、トラック内レコード数 (N_R) が多い (C_R が小さい) 場合は C_B を小さくできる。

また、DKCインタフェース制御部および内容検索部と入力バッファ全体との、単位時間当たりのデータ転送量は、バッファ面数 (N_B) の多少に依存しない。このため、表3の条件を満たす入力バッファは、総容量が $C_T \times 4$ で、 N_B バンクに区切って使用可能な一つのメモリで構成できる。

しかし、総容量を $C_T \times 4$ とすると、 $N_R \geq 2$ の場合に実際に使用されるバッファ容量は $1/2$ となる。この

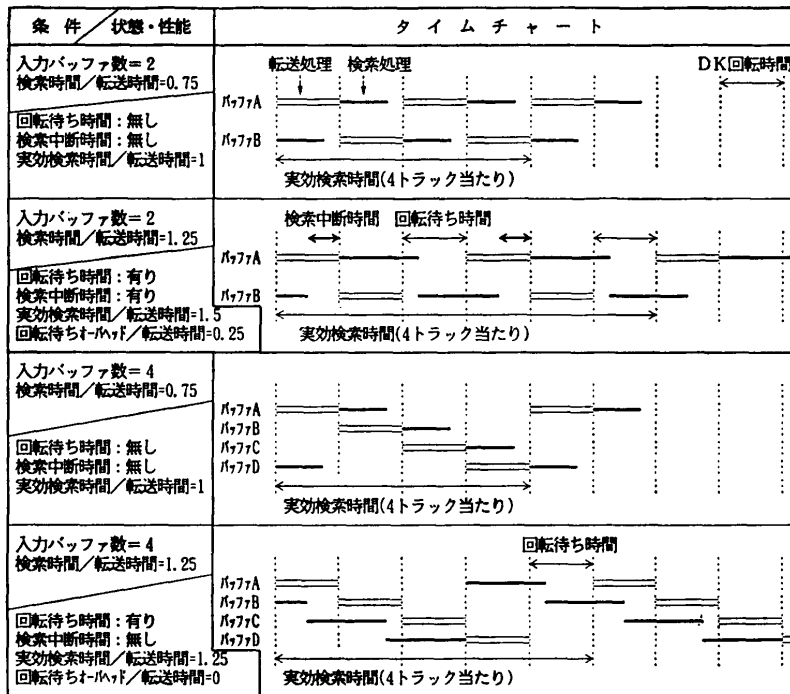


図 3-(a) CSP におけるサーチ処理の流れ (レコード数/トラック=1 の場合)
 Fig. 3-(a) Search operation flow by CSP (the number of records per track=1).

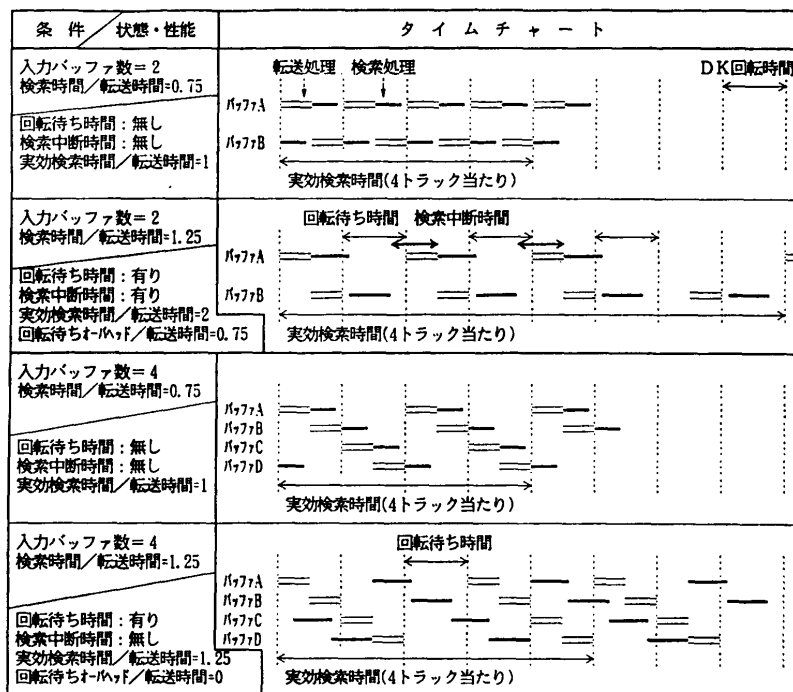


図 3-(b) CSP におけるサーチ処理の流れ (レコード数/トラック=2 の場合)
 Fig. 3-(b) Search operation flow by CSP (the number of records per track=2).

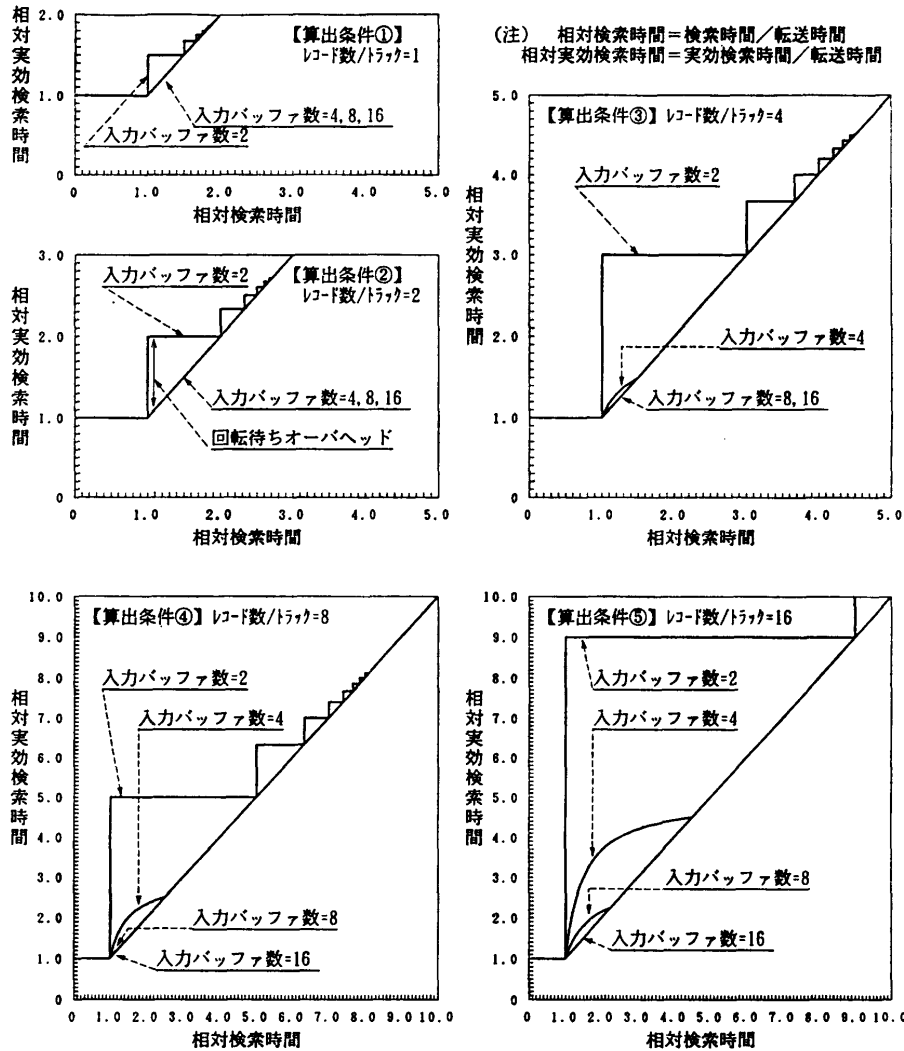


図4 レコード転送時間で正規化した検索時間と実効検索時間
Fig. 4 Search time and effective search time normalized by transfer time.

ため、総容量を $C_T \times 2$ とし、 $N=1$ のときに、 $N_B = 2$ とし、 $N_R \geq 2$ のときは表3の最適構成と同一とする近似最適構成が考えられる。実際、本システムで主に使用されるレコード容量は、 $N_R \geq 2$ に相当するものである。

さらに、実際に使用されるレコードの容量はDBMSが定めている最大値 (C_{Rmax}) と最小値 (C_{Rmin}) の間の値であるため、総容量を $C_{Rmax} \times 2$ とする表4のバッファ構成が採れる。ここで C_{Rmax} は、システムで使用される各種DKの中で最大の C_T とほぼ等しい(僅かに小さい)。

以上の設計による、入力バッファ構成例は図6に示す。入力バッファメモリは16バンク構成でバンクアドレス・レジスタ(4ビット)の下位3ビットはバン

ク内アドレス・レジスタの上位としても使用できるようになっている。アドレス制御部がサーチ対象レコードの容量に応じて上記3ビットの役割を制御している。

このようなバッファ構成により、回転待ちオーバーヘッドは回避され、実効検索時間は次式となる。

$$T = \max(T_s, T_T) \tag{2}$$

本入力バッファ構成は、容量が最大トラック容量のほぼ2倍であり、トラック単位のダブルバッファをレコード数/トラックに対応して細かく分割、制御するものに相当する。きめ細かく分割、制御することにより、トラック単位のダブルバッファでは回避できない回転待ちオーバーヘッドを回避している。また、最初のレコード入力から検索開始までの時間は、トラック単

表3 レコード数/トラックに対する最適な入力バッファ面数

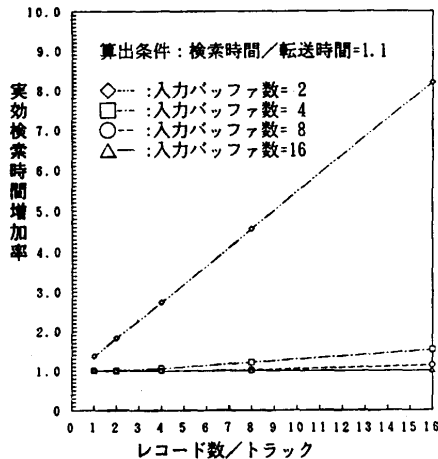
Table 3 Optimal number of input buffers for number of records per track.

レコード数/トラック (N_R)	最適なバッファ面数 (N_B)
1	4
2	4
4	8
8	16
16	16

表4 レコード容量に対する入力バッファ構成
Table 4 Input buffer configuration for record capacity.

C_R	対応する N_R	$C_B \times N_B$
$C_{Rmax}/2$ 超 $\sim C_{Rmax}$	1	C_{Rmax} 2
$C_{Rmax}/4$ 超 $\sim C_{Rmax}/2$	2	$C_{Rmax}/2$ 4
$C_{Rmax}/8$ 超 $\sim C_{Rmax}/4$	4	$C_{Rmax}/4$ 8
C_{Rmin} 以上 $\sim C_{Rmax}/8$	8	$C_{Rmax}/8$ 16

(注) $C_{Rmin} = C_{Rmax}/16$
 C_R : レコード容量, N_R : レコード数/トラック,
 C_B : バッファ容量, N_B : バッファ面数



(注) 実効検索時間増加率 = 実効検索時間 / 検索時間

図5 トラック当たりのレコード数と実効検索時間増加率

Fig. 5 The ratio of effective search time to search time.

位のダブルバッファでは1トラックの転送時間であるが、本最適構成では1レコードの転送時間であり、レコード数/トラックが多いときにこの差は大きくなる。

5. 性能評価

5.1 評価条件

入力バッファ構成、および入力バッファ構成検討の前提となっているディスク装置アクセス方式(マルチトラックリードによる I/O 時間短縮)の効果を実文書データベース¹²⁾に対する文字列検索により評価する。評価に用いたデータベースの構成を表5に示す。評価に用いたシステムは、ホスト計算機が小型汎用機の DIPS-V 30 E¹⁰⁾, CSP が1台、容量が1.3G バイト、データ転送速度が3M バイト/秒の DK および DKC が各1台の構成である。

処理時間を実測する問合せを表6に示す。このう

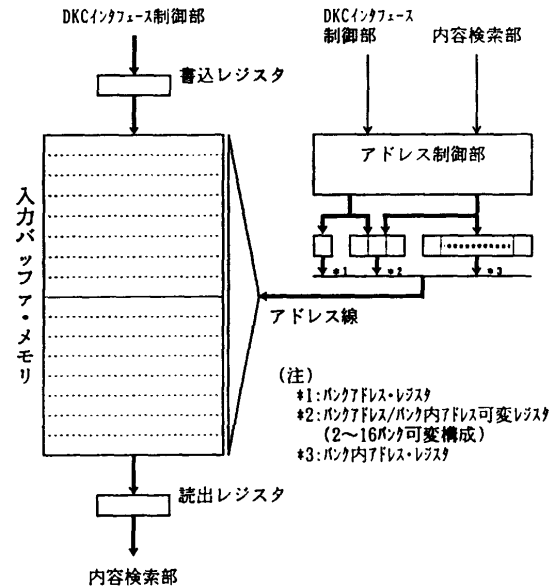


図6 入力バッファの構成例

Fig. 6 Input buffer configuration.

ち、Q1 は固定長日本語文字列に対する文字列検索であり CSP を使用しない場合と、使用する場合の処理時間を比較し、CSP による I/O 時間短縮効果の評価するものである。Q2 は平均 2.5K バイトの可変定長日本語文字列に対する複数キーワードの文字列検索であり、CSP に対して高負荷を与えて、バッファ構成による性能差を評価するものである。Q2 は汎用計算機にとって極めて高負荷であるため、CSP を使用しない場合は、実行が制限されている。文献8)で報告した性能評価で使用した拡張ウイスコンシン・ベンチマーク^{13), 14)}では上記 Q2 のような高負荷を CSP に与えられないので、本データベースにより評価する。

処理時間は、DBMS が評価プログラムから SQL 文を受け取ってから検索結果をすべて評価プログラム

表 5 性能評価に用いたデータベース
Table 5 Database in the test.

項目	内容
データベースの内容	文書 (社内実文書).
列構成	文書管理情報 固定長文字 (合計 33 バイト).
	表 題 固定長日本語文字 (254 バイト).
	本 文 可変長日本語文字 (平均2.5Kバイト).
行 件数	1000.
容 量	約 3 Mバイト.
格納レコード容量	20480バイト.

表 6 性能評価に用いた問合せ
Table 6 Executed queries in the test.

問合せ	SQL 文
Q1 標題に対する 文字列検索	SELECT*FROM 表 WHERE 標題 LIKE '%○△□%' (OR 標題 LIKE '%○△□%')..... (キーワード数-1) 回繰り返し.
Q2 本文に対する 文字列検索	SELECT*FROM 表 WHERE 本文 LIKE '%○△□%' (OR 本文 LIKE '%○△□%')..... (キーワード数-1) 回繰り返し.

(注) ○△□: キーワード.

に返却し終わるまでの経過時間である。処理時間は CPU 時間と I/O 時間に分けて測定する。CSP を使用する場合の I/O 時間はほとんど CSP 処理時間である。

5.2 評価結果

CSP を使用しない場合と、使用する場合の Q1 の処理時間を図 7 に示す。また CSP を使用する場合の Q2 の処理時間の I/O 時間 (すなわち CSP 処理時間) を図 8 に示す。なお、図 8 には CSP の入力バッファ構成を 2 面とした場合の CSP 処理時間の推定値も示す。各図において、処理時間は、CSP を使用し Q1 でキーワード数を 1 として実行した場合の処理時間を 1 とした相対値である。

図 7 から、CSP を使用することにより、キー数に依存せずに、I/O 時間はほぼ 1/2 に短縮されることがわかる。

図 8 から、CSP に高負荷を与えた場合は、一定の負荷 (キーワード数) まで CSP 処理時間は一定であるが、それ以降は負荷に比例して CSP 処理時間は増加することがわかる。

5.3 考 察

(1) ディスク装置アクセス方式評価

4.1 節で述べた、マルチトラックリードによる I/O 時間短縮率 ($1/E_M$) は式 (1) のとおりである。本実験においては、 $C_T=47K$ バイト、 $C_R=20K$ バイト、 $N_R=2$ であるから、 $E_M \approx 2$ となり、I/O 時間はほぼ 1/2 に短縮される。

(2) 入力バッファ構成評価

Q2 においてキーワード数が 5 以下の場合、および Q1 の CSP 処理時間はレコード転送時間 (T_T) に等

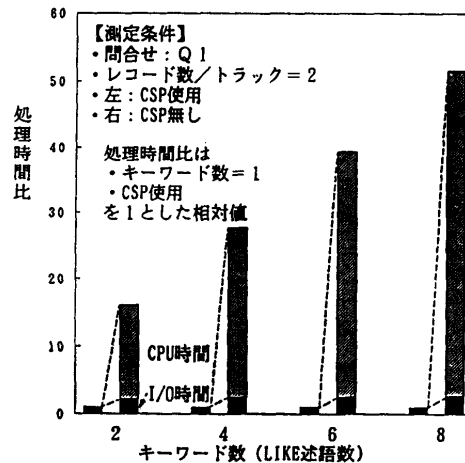


図 7 CSP による性能向上効果
Fig. 7 Performance improvement by CSP.

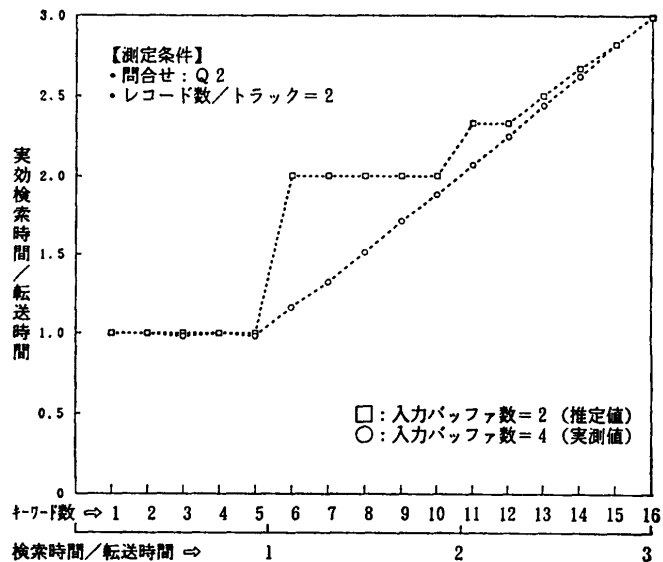


図 8 レコード転送時間で正規化した検索時間と実効検索時間 (CSP 処理時間)
Fig. 8 Search time and effective search time normalized by transfer time.

しい。

3.2 節で述べたように、CSP の内容検索部の検索時間 (T_s) は検索条件の負荷増によって増加し、 T_s が T_r より長くなったとき DK の回転待ちが発生する。Q2 においては、キーワード数が 6 以上のとき DK の回転待ちが発生する。本実測ではレコード数/トラック=2 であるため、図 4 の算出条件②に相当する。実測においては CSP が入力バッファ数を 4 (最適構成) としているため、図 4 の入力バッファ数=4 の場合の実効検索時間と図 8 の実測値は一致しており、回転待ちオーバーヘッドはない。図 8 において、CSP の入力バッファ数を 2 とした場合の実効検索時間の推定値(図 4 から推定)と実測値を比較すると、キーワード数が 6 から 11 において処理時間の短縮効果がある。

6. おわりに

本論文では、サーチ処理を実行する CSP における検索入力バッファの構成法について述べた。要約すると以下のとおりである。

CSP は汎用ディスク装置の性能を最大限に生かすため、シリンダ単位のマルチトラックリードでデータ読み出しを行い、レコード単位でデータ読み出しと内容検索のパイプライン処理を行う。

データ読み出しと内容検索のパイプライン処理をモデル化し、トラック内レコード数と入力バッファ面数をパラメータとして、回転待ちオーバーヘッドを評価し、トラック内レコード数の 2 倍の入力バッファ面数を用意すれば、回転待ちオーバーヘッドを回避できることを明らかにした。これに基づき、入力バッファは複数バンクに変更可能なメモリで構成することとした。

性能の実測評価により、本設計による入力バッファ構成では、CSP の高負荷状態において回転待ちオーバーヘッドが生じていないことを確認した。

謝辞 RINDA の開発を推進していただいた NTT 情報通信網研究所の石野福彌所長、松永俊雄主席研究員、拜原正人情報処理研究部長を始め関係者皆様に深謝いたします。

参 考 文 献

- 1) ISO 9075: Information Processing Systems—Database Language SQL (1987).
- 2) JIS X 3005: データベース言語 SQL (1987).
- 3) 喜連川, 伏見: データベースマシン, 情報処理, Vol. 28, No. 1, pp. 56-67 (1987).

- 4) 清水: データベースマシンの動向, アドバンスト・データベース・システムシンポジウム論文集, pp. 31-40 (1987).
- 5) Babb, E.: Implementing a Relational Database by Means of Specialized Hardware, *ACM Trans. Database Syst.*, Vol. 4, No. 1, pp. 1-29 (1979).
- 6) Ozkarahan, E. A. and Penaloza, M. A.: On-the-Fly and Background Data Filtering System for Database Architectures, *New Generation Computing 5*, OHMSHA and Springer-Verlag (1987).
- 7) 速水, 井上, 福岡, 鈴木: リレーショナルデータベースプロセッサ RINDA のアーキテクチャ, 情報処理学会計算機アーキテクチャ研究会資料, 88-ARC-73-12, pp. 85-92 (1988).
- 8) 井上, 速水, 福岡, 鈴木, 松永: データベースプロセッサ RINDA の設計と実現, 情報処理学会論文誌, Vol. 31, No. 3, pp. 373-380 (1990).
- 9) 小柳津, 塩川, 木ノ内, 安保: DIPS-11/5E シリーズの実用化, NTT 研究実用化報告, Vol. 36, No. 1, pp. 49-56 (1986).
- 10) 矢沢, 平野, 山口, 岡田: DIPS-V 30E のハードウェア構成, NTT 研究実用化報告, Vol. 37, No. 9, pp. 523-532 (1986).
- 11) 井上, 北村, 速水, 中村: 情報提供サービスに適用可能な超大規模リレーショナル・データベースマシン, 情報処理学会データベースシステム研究会資料, 85-DB-47-5 (1985).
- 12) 速水, 黒岩, 中村: 文書管理システムへの RINDA の適用効果について, 電子情報通信学会秋期全国大会, D-65 (1990).
- 13) Bitton, D., DeWitt, D. J. and Turbyfill, C.: Benchmarking Database Systems—A Systematic Approach, CSTR #526, Univ. of Wisconsin-Medison (1983).
- 14) DeWitt, D. J. et al.: A Single User Evaluation of the GAMMER Database Machine, *Proceedings of the 5th IWDM*, pp. 43-59 (1987).

(平成 3 年 3 月 25 日受付)
(平成 3 年 9 月 12 日採録)



速水 治夫 (正会員)

昭和 22 年生。昭和 45 年名古屋大学工学部応用物理学科卒業。昭和 47 年同大学院工学研究科応用物理学専攻修士課程修了。同年、日本電信電話公社入社。現在、NTT 情報通信網研究所 基本アーキテクチャ研究部 主幹研究員。主に DIPS ハードウェアシステムの研究実用化を行い、現在データベースマシンの研究実用化に従事。