データベースプロセッサ RINDA におけるディスク装置アクセス競合制御方式†

速水治夫††

内容検索プロセッサ (CSP) は，インデックスの利用が困難な非定型の検索処理の高速化を目的として，ホスト計算機と DK との間に接続されるデータベースプロセッサである．CSP はホスト計算機の検索指令を受け，DK 中のデータベース (DB) を検索した結果をホスト計算機へ転送する．定型の検索・更新処理はホスト計算機が実行するため，CSP 配下の DK では CSP とホスト計算機からのアクセスが競合する．特に，CSP は DB の全データを連続的に読み出すので，ホスト計算機からの DK アクセスが長時間待たされ，定型処理のスループットが低下し，ターンアラウンド・タイム (TAT) が増大する恐れがある．これに対し，本論文では以下の DK アクセス競合制御方式を提案する．①CSP には検索範囲を単位検索範囲に区切って検索指令する．②区切りごとにホスト計算機で待っている DK アクセス要求をすべて実行する．③ホスト計算機からの DK アクセス中は CSP は検索指令を受け付けない．上記②，③を実現するため，CSP の I/O インタフェース制御部は，ホスト計算機からの CSP アクセスと DK アクセス，および CSP からの DK アクセスの受付・実行を統合的に制御する．提案方式では，定型処理のスループットは低下せず，単位検索範囲のページ数などを適切に選ぶことにより TAT の増加を一定以下に制御可能なことをシミュレーションにより確認する．

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 (DB) の論理構造は単純な表形式であり，その表の任意の行と列の集合に対してデータ操作を行うことができる．そのためのデータ操作は，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，近年の半導体技術の進歩を背景とした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なかに，ディスク装置 (DK) 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を使用する方式も提案されている[5),6)]．著者らもフィルタプロセッサ型の内容検索プロセッサ (CSP: Content Search Processor) を構成要素の一つとするリレーショナルデータベースプロセッサ RINDA を開発した[7),8)]．

CSP は RDB に対するインデックスの利用が困難な非定型の検索処理（非定型処理と略す）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 DB が格納される DK との間に入出力インタフェースで接続され，DIPS 上のデータベース管理システム (DBMS) によって制御される．以下では，RINDA と DIPS およびこれらを制御する DBMS を含めて RINDA システムと呼ぶ．

CSP が実現する機能は非定型処理のみであり，その他の定型の検索・更新処理（定型処理と略す）や DB の生成処理は DBMS によって実行される．ユーザプログラムとのインタフェースは DBMS が一括して扱っており，ユーザプログラムから CSP を意識する必要はない．これらの定型処理と非定型処理は同一の DB に対して混在して処理要求が発生する．このため，ホスト計算機と CSP から同一の DK に対するアクセスが共存する．

定型処理のための DK アクセスは単純なページアクセスであるが，非定型処理では表中のすべての行を読み出す必要があり，CSP は連続的に DK を使用する．このため，CSP により非定型処理が連続的に実行されると DK の使用中が継続し，同一 DK に対するホスト計算機からのアクセスが受け付けられず，定型処理のスループットが低下し，ターンアラウンド・タイム (TAT) が増大する恐れがある．

これまでに報告されているフィルタプロセッサでは，配下の DK に対するホスト計算機とのアクセス共存・競合制御について述べられていない．CSP のようなフィルタプロセッサでは，上記の DK アクセスの共存・競合制御は重要な問題である．

† A Conflict Control of Disk Access in Database Processor RINDA by HARUO HAYAMI (NTT Network Information Systems Laboratories).

†† NTT 情報通信網研究所

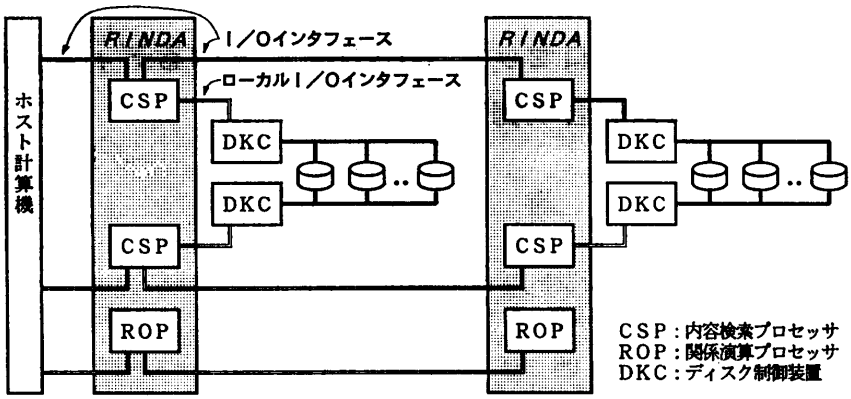


図 1 RINDA システムの構成例
Fig. 1 Typical RINDA system organization.

本論文では，同一の DK に対するホスト計算機と CSP からのアクセスを共存処理し，アクセスが競合する時にはホスト計算機からのアクセスを優先処理する DK アクセス競合制御方式，およびこれを実現するための DK 接続方式を提案し，次にシミュレーションにより上記方式を評価した結果を報告する．2章で CSP の概要を示し，3章で DK アクセス競合制御方式，4章で DK 接続方式を述べ，5章ではシミュレーションによる提案方式の有効性の確認を述べる．

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．定型の検索処理の代表的な例は表の中からユニークなキーを用いて単一の行を選択する処理であり，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．

他方，非定型の検索処理にはあらかじめインデックスの作成されていない列に対して条件を指定した検索や，インデックスの作成が困難な文字列データの列に対する任意文字列の部分一致検索などがあり，これらを従来のソフトウェアのみのシステムで実行すると多大の処理時間がかかっていた．その原因はインデックスが使用できないため，表中のすべての行を読み出して検索条件を判定する処理などの負荷が大きく，多大の処理時間がかかっていたことによる[11]．

上記の問題を解決するために開発された RINDA では，CSP と関係演算プロセッサ (ROP : Relational Operation Accelerating Processor)[12] により，大規模 DB に対する非定型処理を汎用計算機に比べて 10～100 倍高速化することを可能にした．

CSP は DK に格納された表を読み出し，検索条件に合致する行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．RINDA システムの構成例を**図 1** に示す．CSP，ROP はそれぞれ独立の I/O 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 I/O インタフェースで接続される．

2.2 DBMS の概要

(1) 構　　成

ホスト計算機上で RINDA を制御する DBMS の構成を**図 2** に示す．DBMS は，既存の DBMS に RINDA 対応の制御機能を追加した構成である．新規に追加した部分は，言語処理部の RINDA 用最適化機能と RINDA 制御部である．DBMS をこのような構成とした理由は以下のとおりである．

① 同一 DB に対する RINDA を使用したアクセスと使用しないアクセスの混在を可能とする．

② RINDA を使用しない既存ユーザに対して影響を与えないようにする．

(2) 機 能 概 要

ユーザプログラムから発行された SQL 文は，言語処理部により構文解析，意味解析，最適化が行われた後に実行される．RINDA 制御部は RINDA を使用した検索処理を実行する．主な機能は RINDA の実行制御，RINDA との入出力域管理である．データベース制御部は RINDA を使用しないでソフトウェアだけによる検索・更新処理を実行する．また，DB の生成機能や，トランザクション管理，排他制御，資源管理などのデータベース処理として一元管理が必要

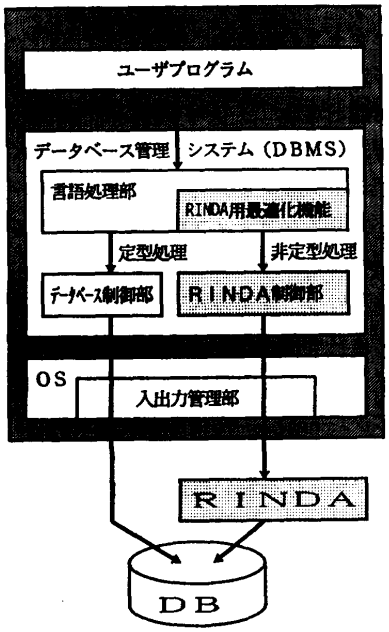


図 2 データベース管理システムの構成
Fig. 2 Database Management System organization.

な制御機能も実現する.

なお，データベース制御部からの DK アクセスや RINDA 制御部からの RINDA アクセスは OS の入出力管理部を通して実行される.

2.3 CSP の動作概要

（1） DBMS による検索指令

DBMS は SQL 文に基づいて CSP に対するオーダを作成する．オーダ中には，検索対象の表が格納されている DK のデバイスアドレス，ページアドレスの範囲（検索範囲と呼ぶ），ページ容量，および検索結果のデータが入ったページを転送する回数（検索結果転送回数と呼ぶ）などの物理制御情報と，表に対する検索条件などを記述した論理制御情報が含まれている.

DBMS は CSP から転送されてくる検索結果を受け取る入力域（検索結果転送回数に等しいページ数）を用意した後，オーダをチャネル・コマンドによって CSP へ転送して検索を開始させる.

第3章で詳しく述べるが，DBMS は検索範囲を複数の単位検索範囲に分割して CSP に検索を指令する.

（2） CSP の検索処理

CSP は受け取ったオーダをもとに DK に対しチャネル・コマンドを発行し，検索範囲のページ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 から読み出したページを格納する入力バッファと検索条件合致行の抽出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.

出力バッファに検索結果のデータがページ単位に蓄積されるごとに，検索結果をホスト計算機に転送する.

CSP は指定された検索範囲の検索が終了する（検索範囲終了と呼ぶ）か，指定された検索結果転送回数のページを転送する（転送回数終了と呼ぶ）と動作を終了する（単位検索処理と呼ぶ）.

（3） DBMS による後処理

単位検索処理終了後，DBMS は入力域の検索結果をワークファイルに退避すると共に終了要因を判定する．転送回数終了の場合は単位検索範囲の検索開始アドレスを更新して CSP に検索を再指令する．検索範囲終了の場合には全検索範囲終了と単位検索範囲終了があり，後者では次の単位検索範囲を指定して CSP に検索を再指令する.

2.4 並列検索処理

CSP の検索処理は表全体を読み出すため，表のデータ量に比例した処理時間を要する．表を複数の DK（ディスクボリューム）に分割格納し，複数（n 台）の CSP で並列検索すれば処理時間を短縮することができる．RINDA システムではユーザプログラムに意識させることなく，複数 CSP・DK を使用した並列検索を実行している．CSPごとの検索結果はDBMS がとりまとめてユーザププログラムに返却する.

ユーザプログラムに CSP 台数を意識させないために，ユーザが総量を指定した単位検索範囲と検索結果入力域のページ数（＝検索結果転送回数）は CSP 対応に n 分割して実行する.

3. ディスク装置アクセス競合制御方式

3.1 問 題 点

2章で述べたように，DBMS のデータベース制御部は定型処理のため DK をアクセスし，RINDA 制御部は非定型処理のため CSP をアクセスする．また，OS の入出力管理部は CSP と DK を独立のデバイスと意識しているため，それぞれ独立な待ち行列でアクセス要求を管理し独立に実行制御する．CSP はホスト計算機からのアクセスを受け付けると DK をアクセスする．CSP の DK アクセスはマルチトラックリードであり，連続的に DKC，DK を保留する．このため，定型処理が一定のトラヒックで処理要求されかつ

同一 DB に対する非定型処理が連続的に実行要求されると，ホスト計算機からの DK アクセスと CSP アクセスに関し何らかの競合制御あるいはスケジューリングを行わないと以下の問題点が生じる．

① CSP に検索させる表は一般的に大きいため，CSP が長時間連続的に DKC，DK を保留しホスト計算機からの DK アクセスが待たされる．

② OS の入出力管理部では CSP アクセスと DK アクセスの待ち行列を独立に管理するため，CSP の検索処理が終了した後に再度 CSP がアクセスされ，ホスト計算機からの DK アクセスがさらに長時間待たされる恐れがある．

③ ホスト計算機からのアクセスで DK がオフライン・シークやオフライン・サーチ中に，CSP が別 DK からマルチトラックリードによるデータ転送を開始すると DKC が長時間使用中となるため，前記 DK からのシーク／サーチの終了割込みが報告できずにホスト計算機からの DK アクセスがタイムアウトになる恐れがある．

上記3点はホスト計算機からの DK アクセスへの影響が大きく，定型処理のスループットが低下し TAT が増大する要因となる．

これに対し，RINDA システムでは以下の DK アクセス競合制御方式を実現した．

3.2 DK アクセス競合制御方式

（1） 問題点①に対処するため，検索範囲を複数の単位検索範囲に分割して CSP の単位検索処理の処理時間を短くする．

（2） 問題点②に対処するため，CSP の単位検索処理が終了した時 IOC 使用中解除割込みを DK のデバイスアドレスで報告する．それを契機としてホスト計算機上の待ち行列にある同一 DKC 配下の DK へのアクセス要求を連続してすべて実行する．

（3） 問題点③に対処するため，ホスト計算機からの DK へのアクセス（コマンドチェイン）が続いている間は CSP は検索指令を受け付けない．

上記(2)，(3)を可能とするため，ホスト計算機と DKC とを直接接続せずにホスト計算機から DK へのアクセスも CSP を経由させる．これにより，CSP の I/O インタフェース制御部がホスト計算機からの DK アクセスの状況も把握して，DKC 使用中解除割込みを報告したり，CSP 自身のアクセス受付を制御する．I/O インタフェース制御部の構成と機能については第4章で述べる．

上記(1)，(2)により，CSP による DK に対する長時間アクセスの実行中，後から実行要求が発生したホスト計算機からの DK アクセスを優先的に実行することが可能となる．

提案方式は優先度が高く・処理時間が短く・ノンプリエンプティブなジョブと，優先度が低く・処理時間が長く・プリエンプティブなジョブのスケジューリング方式（最短処理時間順方式[13]）に近いが，以下の点が異なっている．提案方式では，処理時間の長いジョブはノンプリエンプティブである（DK の動作はノンプリエンプティブである）．このため，提案方式はノンプリエンプティブ・プライオリティ方式[13]にも近いが，以下の2点が異なっている．優先度が低く・処理時間が長いジョブを短い処理単位に区切って実行している．処理単位は一定時間で区切られたものでなく，次節で述べるように単位検索範囲，検索結果転送回数，ヒット率や CSP 台数などによって定まる．

3.3 単位検索処理時間

前節の制御により，CSP の単位検索処理が終了した後ホスト計算機上の待ち行列中の実行可能な DK アクセスはすべて実行される．したがって，ホスト計算機からの DK アクセスが CSP の DK アクセスにより待たされる待ち時間は，平均的には単位検索処理の処理時間（単位検索処理時間と略す）の 1/2 と見積もれる．

CSP は DK からの連続読み出しに重畳して検索処理しているので，単位検索処理時間（T_S）は以下の DK 読み出し時間で見積もれる．

$$T_S=(N_T+\lceil N_T/N_{TS}\rceil)T_R$$
$$N_T=\min(N_{PE}, N_{PR}/H)/(N_{PT}\cdot P) \qquad (3\text{-}1)$$

ここで，N_T：単位検索範囲の等価トラック数
N_{TS}：DK のシリンダ内トラック数
T_R：DK の回転時間
N_{PE}：単位検索範囲のページ数
N_{PR}：検索結果転送回数（ページ数）
N_{PT}：DK のトラック内ページ数
H：検索条件適合率（ヒット率）
P：CSP の並列数
$\lceil X\rceil$：Xを越える最小の整数
$\min(X, Y)$：X，Yの小さいほう

ただし，DB の格納ページと検索結果転送ページの容量は同一である．

なお，$N_{PE}\leqq N_{PR}/H$ の場合は検索範囲終了となり，$N_{PE}>N_{PR}/H$ の場合は検索範囲終了となる前に転送

回数終了となる．したがって，N_{PE}，N_{PR} が一定で H が大きくなると T_S は小さくなる．

ホスト計算機からの DK アクセスの待ち時間は，$T_S/2$ 以外に先行する定型処理の DK アクセスによる待ち時間もあるが，その寄与分は $T_S/2$ に比べて小さい．このため，式(3-1)に基づいて，各システムの所要 TAT 達成に必要な N_{PE}，N_{PR}，P などを決めることができる．

4. ディスク装置接続方式

4.1 I/O インタフェース制御部の構成

I/O インタフェース制御部には，**図3**に示すようにホスト計算機からの I/O インタフェースと CSP が DKC を接続する I/O インタフェースとの間にスイッチ機構付のバイパスを設けている．ここで，CSP と DKC との間の I/O インタフェースをローカル I/O インタフェースと呼ぶ．ホスト計算機からの I/O インタフェースには CSP が複数台接続されたり，他の種類の周辺装置が混在して接続される．

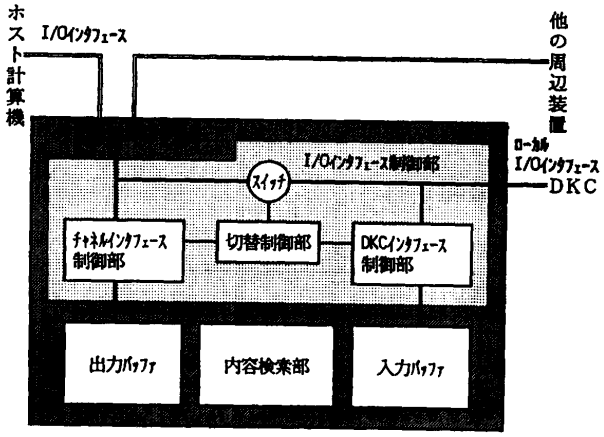


図 3 CSP のハードウェア構成
Fig. 3 CSP hardware configuration.

I/O インタフェースとローカル I/O インタフェースをとおして周辺装置のデバイスアドレスはユニークに付与される．CSP は自分自身のデバイスアドレスとローカル I/O インタフェースに接続される DK のデバイスアドレスを管理している．次節で述べる制御により，2階層の I/O インタフェースを仮想的に1階層の I/O インタフェースとしてホスト計算機に見せている．ホスト計算機は CSP の存在を意識することなく CSP 配下の DK にアクセスする．

本方式では I/O インタフェースを仮想的に1階層としたが，実体は2階層であるため以下の利点がある．

CSP が動作中，ローカル I/O インタフェースはマルチトラックリードのため使用中となっているが，I/O インタフェースはブロッキングした検索結果を転送するとき以外は解放されている．このため1本の I/O インタフェース配下に複数 CSP を接続して同時動作が可能である．

4.2 スイッチ制御

I/O インタフェース制御部は常時 I/O インタフェースを監視しており，**表1**に示す状態遷移に基づいてローカル I/O インタフェースとの分離・直結を制御する．

スイッチ制御を要約すると以下のとおりである．

① 初期状態ではローカル I/O インタフェースと I/O インタフェースは直結されている．

② CSP の検索処理が起動された場合はローカル I/O インタフェースを分離して独立に使用する．この間のホスト計算機からの DK アクセスには DKC 使用中を報告しアクセスを受け付けないが，アクセス履歴

表 1 I/O インタフェース制御部の状態制御
Table 1 Status control in I/O interface controller.

状態番号・状態名		①初期状態	②CSP 動作状態	③DKC 動作状態
内容	I/O インタフェースへの応答主体	—	CSP	DKC
	ローカル I/O インタフェースの接続	直結	分離	直結
状態遷移となるイベント	CSP アクセス	状態②へ遷移	—	CSP 使用中報告 (アクセス履歴保持*2)
	CSP アクセス終了	—	状態①へ遷移 *1 に対し使用中解除割込	—
	DK アクセス (ホスト計算機)	状態③へ遷移	DKC 使用中報告 (アクセス履歴保持*1)	DKC の制御に委ねる (DK 多重動作可能)
	DK アクセスすべて終了 (ホスト計算機)	—	—	状態①へ遷移 *2 に対し使用中解除割込

をスタックする．CSP の検索処理が終了すると，DKC 使用中解除の割込みをスタックした DK アドレスで報告する．

③ ホスト計算機から DK にアクセスがあると DKC 動作状態にする．この間も I/O インタフェース制御部はホスト計算機と DK との応答を監視しており，オフライン・シーク/サーチのときでも初期状態に戻さず，一連のコマンドチェインが終了した後で初期状態に戻す．この間にホスト計算機から他の DK へのアクセスがあると，その受付は DKC の制御に委ねられ，複数 DK への多重アクセスは可能である．DKC 動作状態の間は CSP へのアクセスは受け付けない．

上記制御により以下の効果がある．

① CSP からの DKC 使用中解除の割込みにより，CSP の単位検索処理中に待ち行列に入ったホスト計算機の DK アクセスはすべて実行される．

② ホスト計算機の DK アクセスによるオフライン・シークやオフライン・サーチ中に CSP アクセスは受け付けられないので，ホスト計算機からの DK アクセスがタイムアウトとなることはない．

5. シミュレーションによる評価

5.1 シミュレーションモデル

定型処理と非定型処理は論理的な表や物理的な DK など種々のレベルで混在処理されるが，ここでは単一の表に対する定型処理と非定型処理の混在処理をモデ

表 2 シミュレーションモデル
Table 2 Simulation model.

項　　目	内　　容
システム構成	図5参照，DKC4，DK4 は共通 モデル① (CSP 並列数：1)：CSP0，DKC0，DK0 モデル② (CSP 並列数：2)：CSP0/1，DKC0/1，DK0/1 モデル③ (CSP 並列数：4)：CSP0～3，DKC0～3，DK0～3
評 価 処 理	(1)非定型処理 (CSP による検索処理)： 表全体を読み出し，検索条件を判定 (2)定型処理 (インデックスによる1件検索)： ・インデックス上段はメモリ上 ・インデックス最下段と表のため，DK を2回アクセス
評 価 対 象	I/O 系処理時間 (CSP 処理と DK 処理)
評 価 項 目	①スループット ②ターンアラウンド・タイム (TAT)
トラヒック	(1)非定型処理：1検索処理終了後，連続して次の検索要求発生 (2)定型処理：1万/時，2万/時，3万/時
データベース構成	・インデックス下段と表は同一 DK にある． ・モデル① (CSP 並列数：1)：DK0 に格納 モデル② (CSP 並列数：2)：DK0/1 に均等格納 モデル③ (CSP 並列数：4)：DK0～3 に均等格納 ・表の容量：1076ページ (ページ容量：20KB) (表の容量は拡張ウィスコンシン・ベンチマークの10万行のデータベースに相当)
非定型処理の詳細モデル	・検索条件適合率 (ヒット率；H)：1%，10% ・単位検索範囲のページ数 (N_{PE})：16，32，64，128，256 ・検索結果転送回数 (N_{PR})：4 ・単位検索範囲と検索結果入力域は CSP 並列数で分割 ・単位検索処理ごとに検索結果入力域はワークファイルに退避
DK 諸元	・平均シーク時間：15 ms (常に平均シーク時間で評価) ・回転時間 (T_R)：16.6 ms ・アクセス時間 (T_{DK})：39.9 ms (トラック読出時間で評価) ・シリンダ内トラック数 (N_{TS})：15 ・トラック容量：47 KB (トラック内ページ数 (N_{PT})：2⇦ (ページ容量：20 KB))

ルとして，提案方式を評価する．

具体的には，単一の表に対する定型処理が所定のトラヒックで処理要求されかつ非定型処理が連続的に処理要求される場合に，提案方式のもとで定型処理のスループットが低下せず，TAT の増加は主に式(3-1)により制御可能なことを確認する．

提案方式はDKアクセスの競合に関するものであるから，I/O 処理に着目してシミュレーションする．

典型的な定型処理はインデックスを索引し該当ページから1行のデータを読み出す処理である．インデックスの上段はメモリにあり，インデックスの最下段は表と同一DKに格納されているので，シミュレーションにおける定型処理は同一DKを2回アクセスする処理とする．

RINDA を使用した検索では利用者の指定によりデータページにロックをかけずに処理を実行する排他制御の超越検索機能を実現している[8]．本機能を用いた場合は，定型処理が更新途中のデータを参照することがあり，同一条件で検索処理を繰り返すと結果が矛盾する可能性がある．しかし，非定型処理ではこの現象を許容できる場合が多い[14]．

本シミュレーションモデルの定型処理は参照処理であるため，排他制御のロック処理を考慮に入れず評価している．上記の排他制御の超越検索機能の使用を前提とすれば，更新処理の定型処理との競合の場合についても本シミュレーションによる評価結果は適用可能である．

シミュレーションモデルを**表2**に，システム構成を**図4**に，シミュレーションの概略フローを**図5**に示す．シミュレーション言語は SLAMⅡ[15]を使用した．

上記モデルにおいて，定型処理のみでトラヒックが3万件/時で DK 使用率は約0.6となる．DK 使用率の0.6はほぼ上限とみなせる値であり，シミュレーション条件として十分と考える．

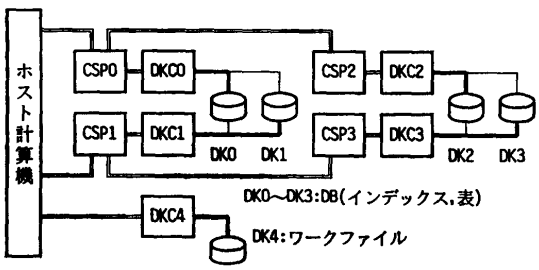


図4 シミュレーションモデルのシステム構成
Fig. 4 System organization at simulation model.

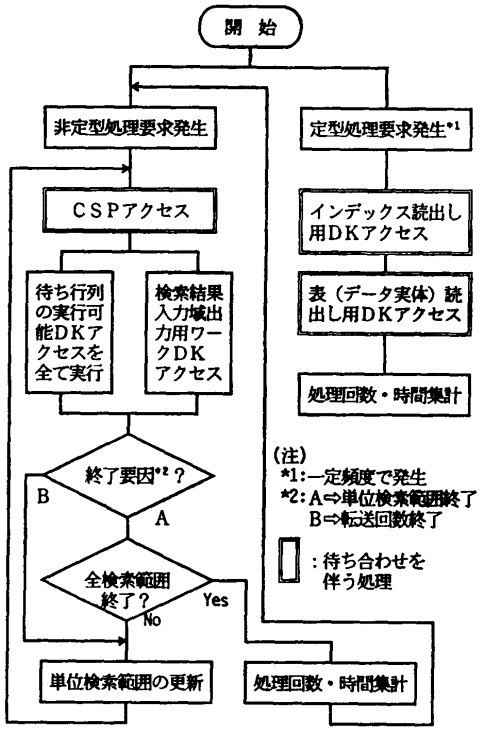


図5 シミュレーションの概要
Fig. 5 Outline of simulation flowchart.

実効的な単位検索範囲は，式(3-1)に示したように $\min(N_{PE}, N_{PR}/H)$ であるから，N_{PR} が大きくなるほど H の影響を受けなくなる．現実的な H の範囲である0.01～0.1で H の違いによる挙動の変化をみるために，$N_{PR}=4$ とする．

表の容量は拡張ウィスコンシン・ベンチマーク[16]の10万行のDBに相当する大きさである．提案方式ではCSPは単位検索範囲ごとに検索するので，表の容量は定型処理のスループットと TAT に影響しない．

5.2 実機測定条件

単位検索処理時間の計算式（式(3-1)）を確認するために，拡張ウィスコンシン・ベンチマークの10万行の DB を用いて非定型処理の TAT を実測する．測定システムは，ホスト計算機が小型汎用機の DIPS-V 30 E，CSP が1台，表3の DK 諸元に相当する DK と DKC が DB 用とワークファイル用に各1台で構成する．TAT 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ワークファイルに格納するまでの経過時間のうち，CSP と DK の動作時間（シミュレーションの TAT に相当）をホスト計算機上で測定した値である．

5.3 評価結果と考察

5.3.1 単位検索処理時間

シミュレーションおよび実機により測定した非定型処理単独走行時の TAT と単位検索処理時間を，**図6**に示す．

（1） TAT

単位検索範囲を小さくすると，単位検索処理ごとのオーバヘッドである検索結果のワークファイル退避と検索開始時の DK のシーク／サーチの回数が増加するため，TAT は増加する．

ヒット率が 10% の場合はホスト計算機へ転送する検索結果のデータ量が多いため，単位検索範囲が 40 (=4/0.1；式(3-1)参照) ページ以上では単位検索範囲を最後まで検索する前に転送回数終了となるので，実効的な単位検索範囲は 40 ページとなる．このため，単位検索範囲が 40 ページ以上で TAT はほぼ一定である．

以上の傾向を含めて，シミュレーション結果は実測値と良く一致している．

（2） 単位検索処理時間

シミュレーション結果の TAT から検索結果のワークファイル退避処理時間を除いた値を単位検索処理の回数で割った値が単位検索処理時間であり，これを T_s' とする．図6にみられるように，T_s' は式(3-1)から得られる T_S と良く一致している．（単位検索範囲が 16 ページの場合を例として，T_S の計算値を以下に示す．パラメータは表3から，$N_{TS}=15$，$T_R=16.6$，$N_{PR}=4$，$N_{PT}=2$ であり，評価条件の $N_{PE}=16$，$H=0.01$，$P=1$ を用いて $T_S=149$ (ms) となる．）

以上により，単位検索処理時間は式(3-1)で見積もれることが確認できた．

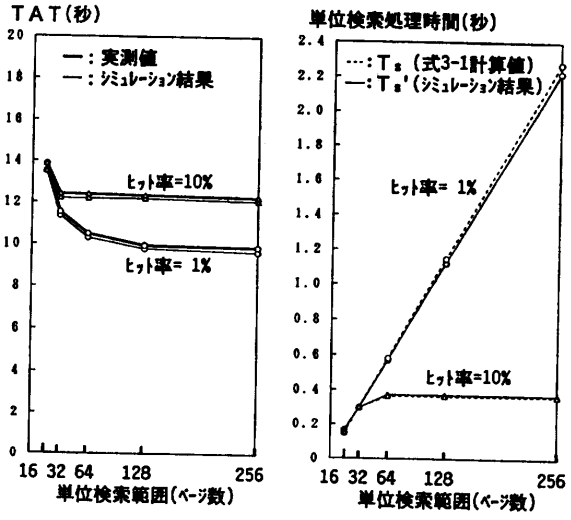
5.3.2 競合制御の評価

シミュレーションにより測定した定型処理・非定型処理競合走行時のスループットと TAT を**図7～図9**に示す．

（1） 定型処理

①スループット

単位検索処理が終了するごとに待っていた定型処理の DK アクセスは実行されるので，定型処理による DK 使用率が 1 以下である限り定型処理はすべて実行される（図7）．



【評価条件】非定型処理単独走行，CSP台数:1

図 6 非定型処理の TAT と単位検索処理時間
Fig. 6 Search time of all extents and single extent.

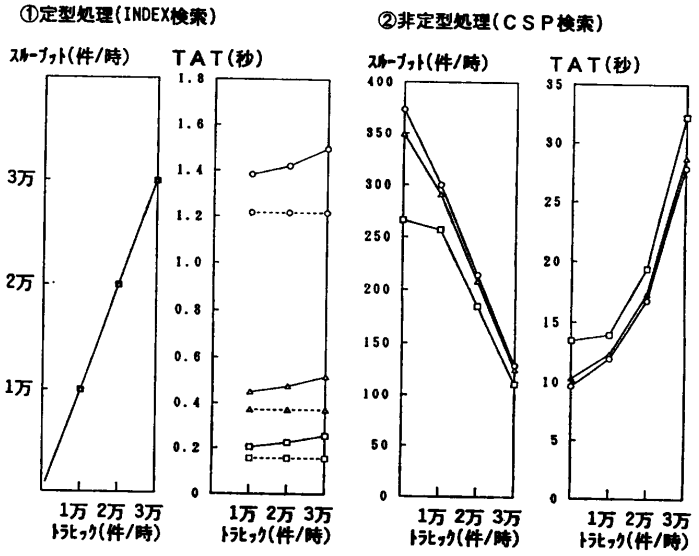
②TAT

シミュレーションモデルにおいて，定型処理の TAT (T_T) は以下で表せる．

$$T_T = W_1 + W_2 + T_{DK} \cdot 2 \qquad (5\text{-}1)$$

ここで，W_1：単位検索処理による待ち時間
$= T_S/2$ (式(3-1)参照)

W_2：先行する定型処理の DK アクセスによる待ち時間



【評価条件】CSP台数:1，ヒット率:1%，○：単位検索範囲ページ数=256，△：単位検索範囲ページ数= 64，□：単位検索範囲ページ数= 16，—：シミュレーション結果，---：$T_s/2+T_{DK}\cdot2$ 計算値(式3-1)

図 7 定型処理のトラヒックと性能の関係
Fig. 7 Relationship between traffic and performance.

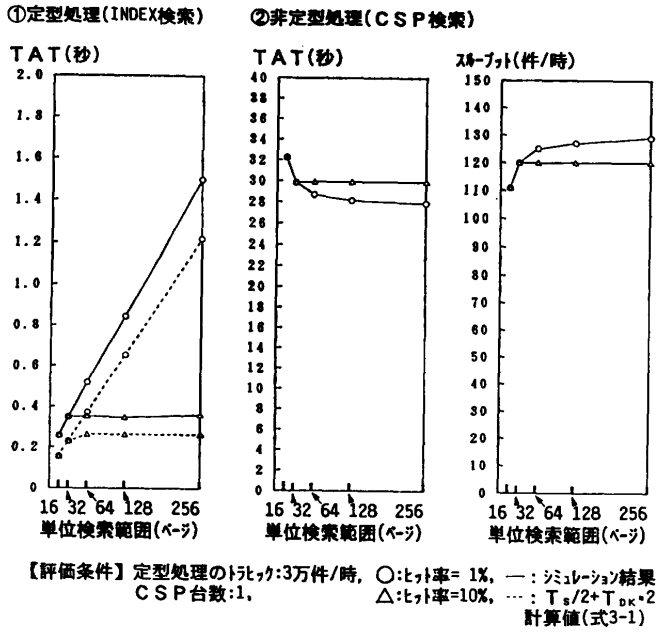


図 8　単位検索範囲と性能の関係
Fig. 8　Relationship between search extent and performance.

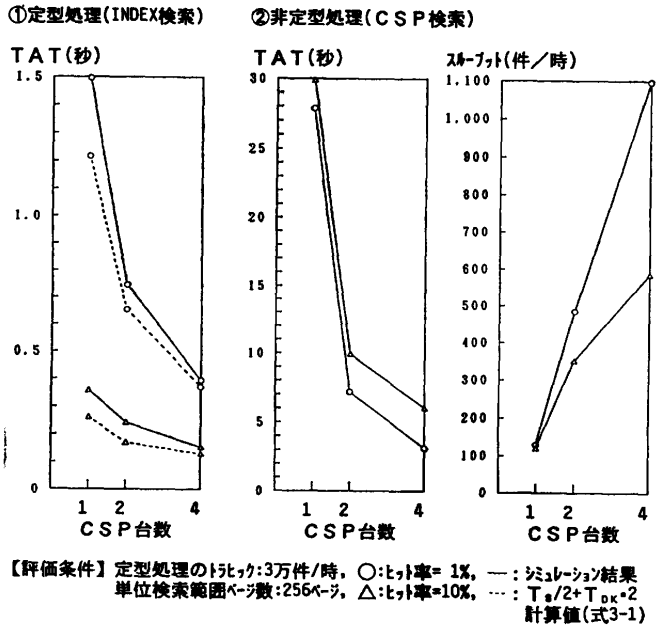


図 9　CSP 台数と性能の関係
Fig. 9　Relationship number of CSPs and performance.

T_{DK} : DK アクセス時間

W_2 は非定型処理との競合がなくても存在する待ち時間であり，定型処理のトラヒックの増加と共に増加する．

図7～9の定型処理の TAT のグラフに，T_T の主要項である $T_S/2+T_{DK}\cdot 2$ の計算値を示す（T_S は式(3-1)の計算値）．図7～9において，T_T の測定値の傾向は上記計算値の傾向と似ており，測定値と計算値との差が W_2 とみなせ，$W_2 \ll W_1$ であることが確認できた．また，T_S は式(3-1)で見積もれることが確認されており，式(3-1)に基づいて定型処理の所要 TAT 達成に必要なパラメータを決定することができる．

提案方式を採らない場合は，TAT の増加は検索対象の表の大きさそのものに依存して大きくなる．

（2） 非定型処理

非定型処理のスループットと TAT は表の大きさに依存するため結果の絶対値は一例であるが，各々の相対的な関係は表の大きさに依存しない．

①TAT

定型処理のトラヒックの増加と共に，単位検索処理の間隙に実行される定型処理の DK アクセスの回数は増加するため，非定型処理の TAT は増加する．

全検索範囲を複数 CSP により分割検索するため，CSP 台数の増加により TAT は減少する．

単位検索範囲と TAT の関係は，5.3.1項で示した考察と同様である．

②スループット

非定型処理は1多重で連続的に処理しているので，スループット≒1/TAT である．

6. おわりに

本論文では，DBMS による定型処理と CSP による非定型処理が混在処理される RINDA システムにおける DK アクセス競合制御方式，および本方式を実現するための CSP の I/O インタフェース制御部の機能と構成を述べた．要約すると以下のとおりである．

（1） DK アクセス競合制御方式の要点は以下である．

① CSP による検索は全検索範囲を単位検索範囲に区切って実行する．

② 区切りごとに，ホスト計算機上の待ち行列にある実行可能な DK アクセス要求をすべて実行する．

③ ホスト計算機からの DK アクセス中は CSP は

検索指令を受け付けない.

(2) 上記②, ③を可能とするため, ホスト計算機とDKCとを直接接続せずにホスト計算機からDKへのアクセスもCSPを経由させる. これにより, CSPのI/Oインタフェース制御部がホスト計算機のDKアクセスの状態を把握してDKC使用中解除割込みを報告したり, CSP自身のアクセス受付を制御する.

上記①, ②により, CSPによるDKアクセス中に, 後から実行要求が発生したホスト計算機からのDKアクセスを優先的に実行することが可能となる.

以上の提案方式のもとでは, 定型処理のスループットは低下せず, TATの増大の程度は単位検索範囲のページ数やCSP台数などによって制御可能であることをシミュレーションにより確認した.

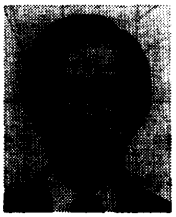
RINDAは現在いくつかのユーザシステムに導入されており, 今後は実システムにおいてDKアクセス競合制御方式の効果を確認する予定である.

謝辞 RINDA開発を推進いただいた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とする関係者皆様, シミュレーション, 性能測定にご協力いただいた野瀬純郎主幹研究員, 中村敏夫主任研究員, 黒岩淳一研究主任, 田中幸子さん, ならびに貴重なコメントをいただいた査読者の方々に感謝いたします.

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・システムシンポジューム論文集, 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 (1988).
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) 武田, 佐藤, 中村, 速水: 関係演算高速化プロセッサ, 情報処理学会論文誌, Vol. 31, No. 8, pp. 1230-1241 (1990).
13) 川口, 藤井 (訳): オペレーティング・システムの理論, 日本コンピュータ協会(1987). (Coffman, E. G., Jr. and Denning, P. J.: *Operating Systems Theory*, Prentice-Hall, Inc. (1973).)
14) 井上, 中村, 芳西, 片岡: データベースプロセッサRINDAの制御プログラム, *NTT R&D*, Vol. 38, No. 8, pp. 869-876 (1989).
15) 森戸, 相沢: SLAMIIによるシステム・シミュレーション入門, 構造計画研究所 (1987).
16) DeWitt, D. J. et al.: A Single User Evaluation of the GAMMA Database Machine, *Proc. of the 5th IWDA*, pp. 43-59 (1987).

(平成3年7月10日受付)
(平成4年10月2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. DIPSハードウェアシステム, データベースマシン, 情報検索システムの研究実用化に従事. 電子情報通信学会会員.