データベース マシンによるCODASYL型DBMSの実現と評価

A DATA BASE MACHINE APPLIED FOR CODASYL DBMS

水摩正行 日吉茂樹 箱崎勝也
Masayuki MIZUMA Shigeki HIYOSHI Katsuya HAKOZAKI
日本電気(株) 中央研究所
Central Research Lab. Nippon Electric Co., Ltd.

1. はじめに

最近、ユーザの間では、データベース化の気運が高まっており、本格的な稼動に入っているシステムも多い。そして、データベースが大規模化してくるにつれ、高性能なデータベース管理システム（DBMS）を望む声が強くなってきた。この要望に応えるため、データベース・マシンの研究が各所で行われている。

データベース マシンはホストコンピュータ上のソフトウェアで実現されたDBMSの機能を切り出し、専用プロセッサに割当てたものである。専用化することにより、データベース処理に適したアーキテクチャが導入でき高速処理が可能になる。また、ホストプロセッサとの並列処理によって、スループットの向上をはかることができる。

我々も実験システムのデータベース マシン（GDS: Generalized Database Subsystem）を研究開発した。

GDSは、主記憶共用型のバックエンド プロセッサで様々なタイプのデータモデルをサポートできることを特徴としている。

GDSの詳細な説明、評価結果については、参考文献[1],[2]に報告されている。

データベース マシンの有効性を確めるには、その本体のみでなく、その周辺にまつわる種々の観点からの検討が必要である。

たとえば、次のようなものがあげられる。

(i) データモデルとの適合性

現在、CODASYL データベース，リレーショナル データベース など、数多くのデータモデルが提案されているが、それらのデータモデルへの適合性についての検討が必要である。

(ii) ホストプロセッサからのオフロード量

データベース マシンを導入することによって、どの程度の処理量がホストプロセッサからオフロードできるか、が問題になる。

これは、対象とするデータモデルにも依存するが、データベース マシンを設計する上での重要な性能指標になる。

(iii) 既存システムからの移行方式、あるいは、移行に要する改造コストが問題になる。

上記の問題を検討するため、我々はGDSを使って、現在広く利用されているCODASYL タイプのDBMSを実現した。この実験システムを ADBS／GDS と呼んでいる。

ADBS／GDSは、商用システムであるACOS4のデータベース管理システムADBS リリース6.1（以後 ACOS4／ADBSと略す）を改造して作成された。

本報告書では、ADBS／GDSの実現方式と、それをもとにして得られたデータベース マシンの評価について報告する。

はじめに、実験データベース マシン（GDS）の特徴を示す。そして、その上で実現され

たCODASYL タイプのDBMS（ADBS／GDS）の構成と実現方式について紹介する。最後に、実験システム（ADBS／GDS）の性能評価の結果を示し、その評価結果をもとにして、2種類のバックエンド プロセッサ型データベース マシンの性能予測をする。

2. GDSの概要

2.1 ハードウェア構成

GDS実験システムは、ACOS4 システム400をホストプロセッサとし、バリアン社のミニ コンピュータ V-76をデータベース専用プロセッサ（GDS）として、使用している。

ホストプロセッサとGDSとの間の通信のため、MMI（Main Memory Interface）とPSI（Peripheral Subsystem Interface）が、準備されている。

PSIは、ACOS4 システム400の入出力インタフェース（チャンネル結合）で、プロセッサ間の同期をとるための制御インタフェースとして、使用される。

MMIは、データ転送用として使用され、そのデータ転送時間を短縮するために、ACOS4 システム400の主記憶インタフェースを改造し、GDSから主記憶を直接アクセスできるようにしている。

2.2 GDSの特徴

GDSは、様々な形態のデータベース システムを、その上で容易に実現できるように、CODASYL型DBMS，リレーショナル型DBMSなど、各種のデータモデルに従ったDBMSの基本機能を提供する。

ホストプロセッサから、GDSへ データベースのアクセスを要求する場合、GDIコマンド（Generalized Database Interface）を介して行う。

GDIコマンドは、単一レコード アクセスを基本としており、GDS内部では、ホストプロセッサからの要求プロセスに対応して、多重処理がなされる。

データベース アクセス機能には、レコードの格納構造に従ったアクセス（一次アクセス）と、二次的なパスとして設けられる イメージ アクセス，リンク アクセスがある。

(1) レコードの格納構造

GDSでは、レコードの格納方式として、次の3種類のものが準備されている。

(i) Key Sequenced 構造

従来の索引順編成に相当し、レコードは、インデックスを持ち、物理的にもキー値によって順序付けされる。本編成を採用することにより、キー値によるダイレクト アクセス、キー値の順序に従った順次アクセスが可能である。

(ii) Key Randomized 構造

レコードは、特定のキー値と一対一の関係をもち、キー値をハッシングすることにより直接レコードをアクセスする（乱編成に相当）

(iii) Entry Sequenced 構造

レコードは、キー値とは無関係にレコードの発生順に格納される（順編成に相当）。

(2) 二次アクセスパス

GDSは、(1)で述べたレコードの格納構造に従ったアクセス機能の他に、二次的なアクセス機能をサポートするため、次のものが準備されている。

(i) イメージ

イメージは、従来のインバーテット ファイルに相当し、データベース レコードの特定のキー フィールドの値とレコードのアドレスを示したイメージテーブルを持つ。このイメージテーブルを介することにより、キー値によるダイレクト アクセス，キー値による順次アクセスが可能である。

(ii) リンク

リンクは、レコード間の関係を示したもので、親レコードと子レコード集合は一連のリンク ポインタで結合されている。リンクはCODASYL提案言語[3]のセットに相当する。

GDIコマンドには、以上の構成に従った，キー値によるダイレクトアクセス機能と、レコードの相対アクセス機能がある。

附録にGDIコマンドを示す。

3. CODASYL インタフェースの実現

3.1 設計方針

GDSの上で、CODASYLインタフェースを実現する際に、我々は、既存システムとの互換性を重視し、次のような方針で設計を行った。

(i) 商用システムであるACOS4/ADBSリリース6.1を対象とする.

(ii) オブジェクト レベルでのプログラムの互換性を保障するため、言語プロセッサ(スキーマ/サブスキーマ トランスレータ、COBOLコンパイラ)には、手を加えず、既存のものを利用する.

(iii) 実行レベルでのインタフェースは、DMLコマンドを直接解釈し、実行する方式で、ACOS4/ADBSのモニタ部を改造する.

このような方法で設計すると、開発工数は削減され、安全サイドな評価データが収集できる。

3.2 ソフトウェア構成

GDSは、DMLコマンドを実行するための基本的なアクセス機能を全て含んでいる.

このため、ホスト プロセッサ側では、言語処理機能と、DMLコマンドからGDIコマンドへの展開機能が残される.

図1にADBS/GDSのシステム構成を示す.

(1) スキーマ/サブスキーマ トランスレータ

スキーマ/サブスキーマ記述言語で書かれたソース スキーマ/サブスキーマをオブジェクト形式のスキーマ/サブスキーマに変換する.

スキーマ/サブスキーマ記述言語は、ACOS4/ADBSの言語に、いくつかの制限を加えたサブセット言語で、そのトランスレータはACOS4/ADBSのスキーマ/サブスキーマトランスレータをそのまま流用している。

(2) COBOL トランスレータ

COBOL言語で書かれた応用プログラムを内部形式に変換する.

オブジェクト レベルでの互換性を保障するため、COBOL トランスレータは、ACOS4/ADBSのトランスレータをそのまま流用している。

(3) DML実行管理(DMLEXC)

GDSの機能を利用し、応用プログラムから要求されるDMLコマンドの実行を司る。DMLEXCは、ACOS4/ADBSのモニタ部に相当し、その部分を改造する形で実現している.

(4) GDSスキーマ トランスレータ

GDSが直接管理するデータベース データ構造を定義するため、GDSスキーマ記述言語が準備されている。

GDSスキーマ トランスレータは、ソース形式で書かれたGDSスキーマを内部形式に変換する.

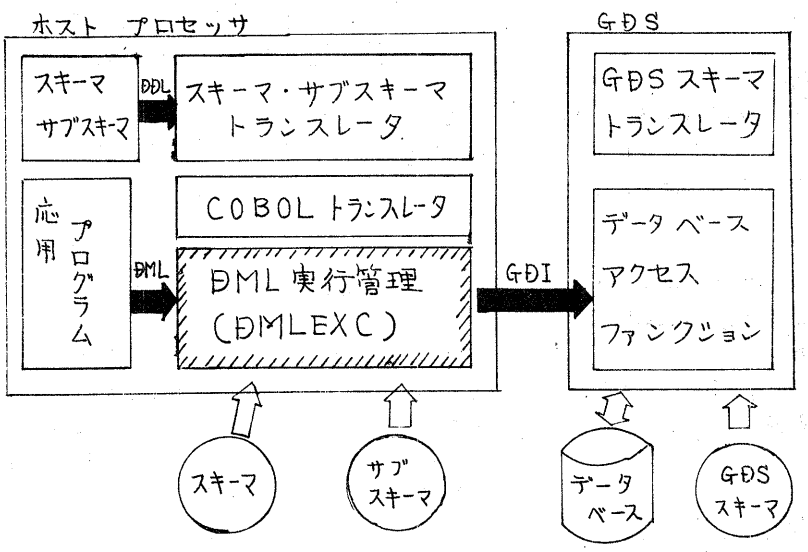


図1. ADBS/GDSの構成

: 新規作成部

(5) データ アクセス ファンクション

GDSが提供するデータ アクセス機能の集まりである。

3.3 GDS アクセス機能との対応

ADBS/GDSは、GDSの機能の一部を使って、実現されている。

(1) ADBSスキーマとGDSスキーマとの対応

データベース データ構造は、ほぼ1対1の関係にある。即ち、ADBSスキーマで表現されるレコード，データ アイテム，セット，レルム のそれぞれに対応して、GDSスキーマには、レコード，フィールド，リンク，エリアの概念がある。

但し、リンクは、CODASYL提案言語のSETに相当するが、SET SELECTION句の概念がない[3]。このため、多段階月にまたがった セットの自動アクセス制御は、ホスト プロセッサ上のDML実行管理部で行われる。

また、GDSのフィールドは、基本データ項目のみ取り扱うので、データ アグリゲートが指定されると、ホスト プロセッサ側で、対応するフィールドへ変換する必要がある。

(2) レコードの格納構造

GDSにおけるレコードの格納構造は、ADBSスキーマで記述されるレコードのLOCATION MODE句に従って、下記のように対応させている。

<ADBSスキーマ>		<GDSスキーマ>
・CALC モード	——	Key Randomized 構造
・VIA セット モード	——	Entry Sequenced 構造

(3) DMLコマンドとGDIコマンド

DMLコマンドは、GDSが提供するGDIコマンドの内、リンク アクセス機能，Key Randomized 構造に従ったダイレクト アクセス機能を利用して、実現されている（附録参照）。

簡単なDMLコマンドについては、DMLコマンド1回につき、対応するGDIコマンドが1回発生する。しかしながら、セット間の関連性に基ずくアクセス コマンドでは、ADBSスキーマの構造に従って複数回発生する。

3.4 DML実行管理

DML実行管理（DMLEXC）は、ユーザプログラムから渡されるDMLコマンドを、解釈し、GDSと交信しながら、要求処理を実行する。以下、DMLEXCの主な機能について、述べる。

(1) GDIコマンドへの展開

DMLコマンドとGDIコマンドは、ほぼ1対1の関係にあるが、完全には一致していない，また、その内部表現形式も異なる。このため、DMLEXCはオブジェクト スキーマ，オブジェクト サブスキーマを参照して、その変換処理を行う（図2）。

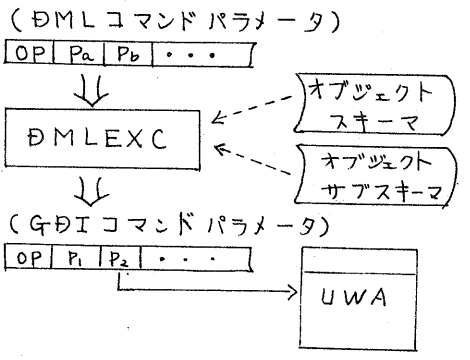


図2. GDIコマンドへの展開

GDSは、要求されたレコード情報を、アプリケーション プログラムで準備したユーザ ワーキング エリア（UWA）へ、直接転送する。このため、対応するUWAのアドレス，及び、転送すべきフィールド名，フィールド長，オフセット情報をGDI コマンド パラメータとして組み込みGDSに渡す。

(2) カレンシー インジケータの維持

GDS内部では、アクセス パスに対応してカレンシー インジケータを維持している。しかしながら、CODASYL提案言語で規定しているカレンシー インジケータのように、特定レコードを選択すると同時に、関連する他のカレンシー インジケータを、自動的に修正する機能は持たない。そこで、DMLEXCが、CODASYLの概念に従ったカレンシー インジケータを維持する。

(3) エラー処理

GDSあるいは、DMLEXCがエラーを検出すると、その対策処理を行い、アプリケーション プログラムにエラー メッセージを送る。

(4) GDSとの制御インタフェース

GDSとの交信は、一般に次の手順で行われる。

(i) START-TR
GDSとの通信を開設するためのコマンドで、最初に発生される。GDSは、要求タスクに対し、トランザクションIDを渡し、以後の通信のための識別子として利用する。

(ii) INITIATE-PATH
レコードに対するアクセス経路を開設する。GDSでは、アクセス経路として種々のものが準備されているが、どのアクセス経路で処理を行うかを宣言する。

(iii) CALL GDI
(ii)で指定したアクセス経路に従って、データベース アクセスを要求する。

(iv) RELEASE-PATH
データベース処理が終了すると、関連するアクセス パスを解放する。

(v) END-TR
GDSとの交信を終了し、START-TR時に得られたトランザクションIDを解放する。

なお、DMLEXCでは、上記の(i)(ii)の手続きを応用プログラムの開始時に、(iv)(v)の手続きを応用プログラムの終了時に行うようにしている。アクセス経路の開設は、サブスキーマで示されている全てのレコード，セットに対して出される。

(5) ハードウェア インタフェース

2.1節で述べたように、実験システムでは、ホストプロセッサとGDSの間の通信手段として、MMIとPSIインタフェースが準備されている。

DMLEXCは、GDIコマンドの発生通知，GDSからの処理の終了通知の同期をとるため，PSIインタフェースを利用している。一方，ホストプロセッサからGDSへ渡されるGDIコマンド パラメータやUWAに関する情報は、MMIインタフェースを介し、GDSが直接参照できるようにしている。MMIインタフェースでは、GDSが主記憶を絶対アドレスで参照するため、DMLEXCは関連情報を主記憶に常駐化し、その絶対アドレスをGDSに通知する。

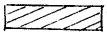
4. 性能評価

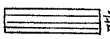
4.1 実験システムの評価

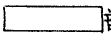
実験システム(ADBS/GDS)を性能評価するため、いくつかのプログラム モデルを作成し、ADBS/GDS とACOS4/ADBSの上で実行させた。

図3はその結果で、代表的なDMLコマンドに対するCPU処理量の比較を示している。

データベースIOの影響を小さくするため、その発生頻度の少いものから、本データを選出した。図3において、各項目の上段はACOS4/ADBS，下段はADBS/GDSの性能を示している。また、ADBS/GDS部の内訳は次のデータを示している。


部：DMLEXCの実行時間
(ホストプロセッサ)


部：GDIコマンドを起動するために要したCPU量
(ホストプロセッサ)


部：GDSでのCPU量

GDI/DMLの欄は、1回のDMLコマンドで発生したGDIコマンドの平均回数を示す。

図の最後は、選択された10個のDMLコマンドの平均実行時間(加重=1)を示す。

なお、実行時間の単位は、ACOS4/ADBSの平均実行時間を1として表わしている。

図3のデータより実験システム(ADBS/GDS)を分析する。

(1) ホストプロセッサからのCPU処理のオフロード率

ACOS4/ADBSの実行管理部の68%がGDSにオフロードされている。本データでは、意識的にデータベースIOの少いものから選出したので、データベースIOの発生も考慮すると負荷の軽減は、更に増大するものと思われる。

(2) プロセッサ間通信オーバヘッド

実験システムでは、GDIコマンドを起動するために、PSIインタフェースを介して行なったが、その通信のために、ACOS4/ADBS実行管理部の約1割のCPU量が、ホストプロセッサに付加されている。この通信量も考慮すると、ホストプロセッサからのCPU処理のオフロード率は57%になる。

(3) DMLコマンドの処理時間

実験システムでは、DMLコマンド当りの平均処理時間(データベースIOが発生しない場合)は、ACOS4/ADBSの約6割である。

(4) CPUオフロード率の高いDMLコマンドには、③,⑦がある。③はFINDコマ

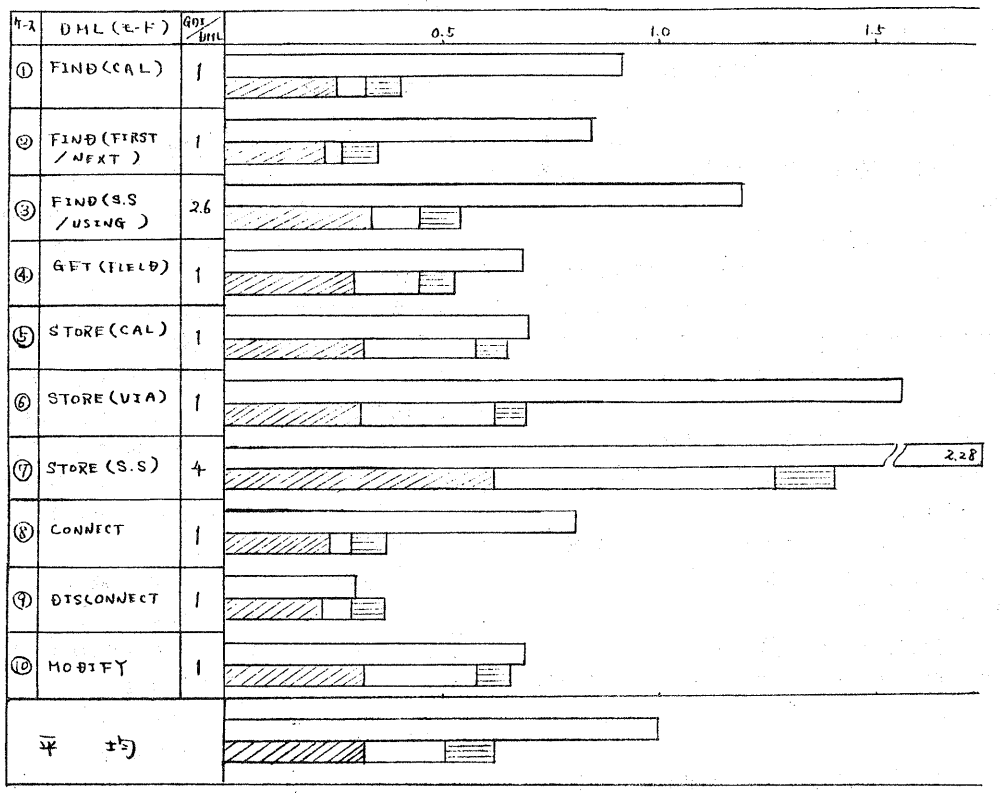


図3 ACOS4/ADBSと実験システムADBS/GDSのCPU性能

の形式7である。⑰は、AUTOMATICメンバ レコードのSTORE命令である。

これらのコマンドは、セット間の関連を自動的に追跡するコマンドで、複数のGDIコマンドを発生する。

一方、CPUオフロード率の低いコマンドには、④のGET命令，⑨のDISCONNECT命令がある。

4.2 データベース マシンの評価

データベース マシンの実現形態には、いろいろあるが、ここで、2つのタイプのデータベース マシンの性能を予測する。

〈汎用データベース マシン〉

・汎用性を重視し、種々のタイプのデータモデルを対象として、データ管理の基本機能のみ提供するデータベース マシン。

〈CODASYL従属型データベース マシン〉

・データベース マシンの機能をより高級にし、CODASYL言語仕様に密着した形のデータベース マシン。

実験システムで得られた結果から、上記のデータベース マシンの性能を予測するために、次のこと仮定する。

(i) ハードウェア環境は、実験システムと同じ環境を考える。即ち、ACOS4 システム400をホストプロセッサとし、データベース マシンは バリアン社のミニコンピュータ V-76 を使用したシステムである。

(ii) データベース マシンの性能は、実験システムと同程度のものとみなす。

(iii) 汎用データベース マシンは、GDSの設計思想に従ったものでその改造版とする。実験システムでは、コンパイラには手を加えず実行管理部のみ修正した。このため、インタフェース上の冗長性があり、その冗長性を除いたものとする。

(iv) CODASYL型データベース マシンでは、スキーマ，サブスキーマ情報をデータベース マシンの管理下に置き、DMLコマンドのメイン処理は、全てデータベース マシン側で行う。ホスト プロセッサ側に残る機能は、応用プログラムとインタフェースをとるための処理である。

また、プロセッサ間通信のためのインタフェース時間をαと仮定する。

(v) ACOS4/ADBS は、ADBS/GDSを開発する時点に発見された項目についての改造を行なったものとする。

以上のような仮定をもとにして得られたデータベース マシンの性能予測図を図4に示す。

汎用データベース マシンは、実行管理部の約77%のCPU量をホスト プロセッサからオフロードしている。

一方、CODASYL従属型データベース マシンによるCPUオフロード率は89%になっている。

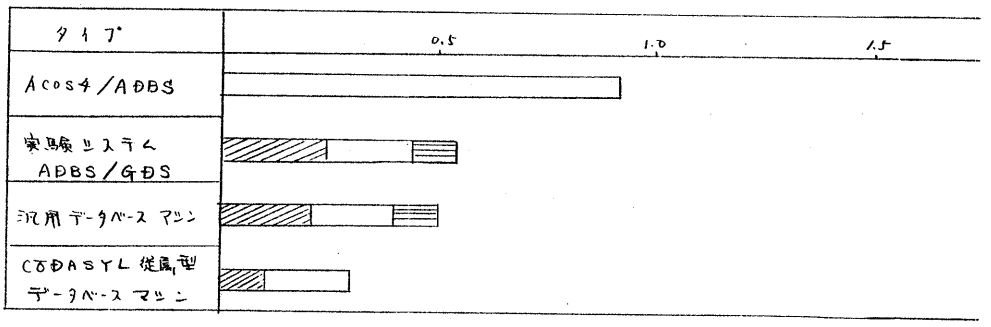


図4 データベース マシンのCPU性能予測

5. おわりに

実験データベース　マシン（GDS）を使って実現したCODASYL型DBMS（ADBS/GDS）の構成とその評価結果について報告した。

ADBS/GDSの開発，評価を通じて，次のことが確認された．

(1) ホスト　プロセッサ上でのソフトウェアの改造量

実験システムを実現する上でACOS4/ADBSのモニタ部を改造したが，その改造に関係したプログラムの　ソース　プログラム　ステップ数が13Kステップで、改造後、約7.3Kステップになった。

データベース　マシンの実用化を考えると、この他、サービス　ユーティリティなどのプログラムの改造が必要になる。

(2) ホストプロセッサのCPUオフロード率

GDSのように、データベース管理の基本機能から成る汎用データベース　マシンでは、実行管理部の約8割が、ホストプロセッサからオフロードされる。一方、CODASYL言語仕様に密着したCODASYL従属型データベース　マシンの場合、実行管理部の9割程度のCPU量が、データベース　マシン側にオフロードされる。

単一レコードアクセスを基本としたDBMSを対象にした場合、汎用データベース　マシン方式でも十分な効果があることが判る。

(3) コマンド　インタラクション

ホスト　プロセッサとデータベース　マシン間でのコマンドの発生頻度は、CODASYL従属型データベース　マシンの場合，1DMLコマンドにつき1回発生する。

汎用データベース　マシンの場合には、スキーマの構造，DMLコマンドの種類に依存するが、簡単なDMLコマンドでは、1DMLコマンドにつき1回発生する。

(4) プロセッサ間通信

実験システムでは、GDIコマンドを起動するための制御インタフェースとして、PSエイントフェース（4チャンネル結合）を利用したが、この通信のために、実行管理部の約1割のCPU量がホスト　プロセッサに付加された。

CODASYL言語仕様のように、単一レコード　アクセスを基本としたDBMS用データベース　マシンでは、プロセッサ間通信の発生頻度が非常に高いので、チャンネルを介さず、直接プロセッサに割込みを起こす方式を採るべきであろう。

一方、リレーショナル　データベースのように、データベース　マシンの機能がより高級になれば、チャンネル結合方式でも適用するであろう。

(5) データベース　マシンの機能レベル

データベース　マシンのような機能分散型プロセッサでは、プロセッサ間の負荷のバランスが問題になる。

CODASYL　DBMSのように単一レコード　アクセスを基本としているDBMSでは、ユーザ　プログラム　レベルでの処理量が比較的多いと云われている．このようなシステムを対象としたデータベース　マシンでは、その性能はそれ程要求されず、低コストなものが要求されよう。一方、リレーショナル　データ　ベース　を対象にした場合、ユーザ　プログラム　レベルでの処理量は、かなり減少することが予想される。このため、高性能なデータベース　マシンを必要とするであろう。

<謝辞>

本研究を進めるにあたり，数多くの方々の御支援と御協力を得た。

日本電気　情報処理事業部　基本ソフトウェア開発本部　登家部長，片山主任をはじめ、そのグループの方々には、ACOS4/ADBSプログラムの提供とその改造に対する種々の助言を頂いた。同中央研究所　コンピュータシステム研究部　松津部長には、研究開発の機会と御指導を頂いた。㈱日本システム　アプリケーション　富林氏，小林氏および高井氏には、ソフトウェア開発を担当して頂いた。

ここに、感謝の意を表します。

<参考文献>

1. K.Hakozaki et al : A Conceptual Design of A Generalized Database Subsystem, 3rd International conf. on VLDB, Oct'77
2. 牧野他 : データベース マシン実験システム, 電子通信学会計算機研究会(情報処理学会と合同) 昭和55年1月 発表予定
3. CODASYL Data Description Language Journal of Development, Jun'73, NBS Hand Book
4. ACOS-4 データ管理 ADBS 概説書.

<附 録>

主なGDIコマンド

(1) DB検索コマンド

a) Direct Access

□	·GET-PRIM-DIRECT	一次キーによる検索
	·GET-IMAGE-DIRECT	二次キーによる検索
□	·GET-CURRENT	カレンシインジケータによる検索

b) Relative Access

□	·GET-PRIM-NEXT/PRIOR	一次キーのオーダリングによる検索
	·GET-IMAGE-NEXT/PRIOR	二次キーのオーダリングによる検索
	·GET-IMAGE-FIRST/LAST	〃
□	·GET-LINK-NEXT/PRIOR	リンクのオーダリングによる検索
□	·GET-LINK-FIRST/LAST	〃
□	·GET-LINK-OWNER	リンクの親子関係による検索
	·GET-SCAN-FIRST/LAST	格納順による検索
	·GET-SCAN-NEXT/PRIOR	〃
	·GET-STAT	システム統計情報の検索

(2) DB更新コマンド

□	·STORE	格 納
□	·UPDATE	内容変更
□	·DELETE	削 除
□	·CONNECT	リンクの関係付け
□	·DISCONNECT	リンク関係の解除
□	·INITIATE-PATH	アクセスパス生成
□	·RELEASE-PATH	アクセスパス解放

注)·□印はADBS/GDSが使用するGDIコマンドを示す。

盛光印刷所