

磁気バブルデータベース計算機の性能評価

Performance Measurement of the Magnetic Bubble Database Machine

大表良一 国分明男 弓場敏嗣 植村俊亮 菅原保雄

Ryoichi OOMOTE Akio KOKUBU Toshitsugu YUBA Syunsuke UEMURA Yasuo SUGAWARA

電子技術総合研究所

Electrotechnical Laboratory

1 まえがき

二次記憶として磁気バブル記憶を用い、これにマイクロコンピュータを組み合わせた基本単位（データモジュール）を並列に配置したデータベース専用の計算機（EDC）を開発し、その性能を測定した。性能は、ソフトウェアモニタにより、各部の動作時間を測定し評価した。測定に当たっては、1500件の文献を関係モデルで表した実験データベースと、これに対する典型的な質問を用意した。応答時間は実験データベースを七台のデータモジュールに分散したものと、四台に分散したものの二種について測定し、並列度の効果を評価した。本文では測定結果並びに、その後の微調整による改善の程度について述べる。

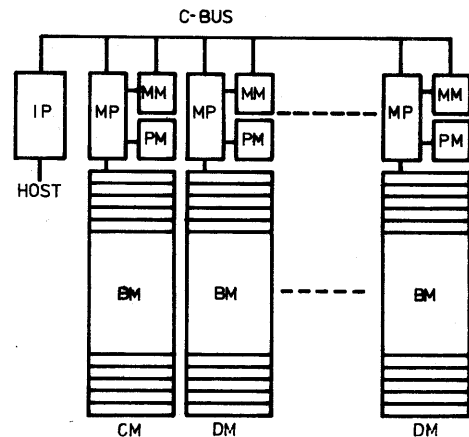
2 EDCのシステム構成

EDCは複数のマイクロプロセッサと磁気バブル記憶とによるデータベース計算機であって

(1) 強化された二次記憶による並列処理と内容呼び出しの実現 (2) 高性能な電子ディスクの採用 という、二つの設計思想のもとに、製作された実験機である⁽¹⁾

図1はEDCのアーキテクチャの概要を示したものであるが、基本単位であるデータモジュール（DM）を多数並列に動作させ、高性能なデー

タベース処理専用の計算機を実現している⁽²⁾。データモジュールは、モジュールプロセッサ（MP）、制御記憶（PM）、主記憶（MM）、磁気バブル記憶（BM）の四つの要素から構成されている。モジュールプロセッサは、マイクロプログラム制御で、400ナノ秒のマシンサイクルを持った、16ビットマイクロプロセッサであり、1) 磁気バブル記憶の制御 2) 内容呼び出し機能の実現の二つの役割を果す。データは磁気バブル記憶に



DM: データモジュール CM: 制御モジュール
MP: モジュールプロセッサ MM: 主記憶
PM: 制御記憶 BM: 磁気バブル記憶
IP: インタフェースプロセッサ

図1 EDCのアーキテクチャ

格納され、主記憶は作業用に、制御記憶はシステムプログラムの格納に用いられる。各データモジュールは共通のバス（Cバス）で互いに結合され、データベース計算機として必要な記憶容量の確保と、並列動作による高速のデータ処理を実現している。

主記憶のサイクル時間は2マイクロ秒であり、制御記憶のそれは1.6マイクロ秒である。磁気パルプ記憶のアクセス時間は平均2.2ミリ秒、データ転送時間は1.25Mバイト/秒である。

データモジュールのうちの一基が各データモジュールに指令を与えて、マシン全体の動作を制御する。これを制御モジュール（CM）とよぶ。CMのプロセッサ（CP）は、他のデータモジュールの主記憶を共有しているのので、他モジュールの主記憶にあるデータを直接操作できる。インターフェイスプロセッサがデータベース計算機と外部環境との整合を取るために有り、システムコンソールなどもこれに接続される。関係、ファイルはバンドという形で、各データモジュールに分散配置されて、並列処理の対象になる。バンドは図2に示すように、7モジュールすべてに渡っても良いし、それ以下でも良い。

EDCは図3に示す階層を持っている⁽³⁾⁽⁴⁾。CインターフェイスがEDCの機械語レベルに相当し、データモデルに対して独立であり、ファイルや関係に相当するバンドの内容呼び出し命令が準備されている。

E、Dインターフェイスはデータモデルに依存し、関係モデルインターフェイスが実装されている。EからCへの変換はCMに内蔵されている、言語プロセッサ（コンパイラ）が行なう。CMにあるCインタプリタがCインターフェイスの命令を解釈実行する。

バンドを操作対象とするC命令を、データモジュール対象にいかえたものがB命令で、B命令

は各データモジュールにブロードキャストされて、Bインタプリタの操作対象になる。Aインターフェイスは磁気パルプ記憶をブロック単位に読み書きする入出力命令の水準であるが、内容呼び出し機能が含まれている。

EDCシステムは、多重プログラミングをサポートする操作システムのほか、表1に示すプロセスから構成されている。⁽⁵⁾⁽⁶⁾

BM1	BM2					BM7
		B	A	N	D	1
		B	A	N	D	2
		B	A	N	D	3

図2 データの記憶方式

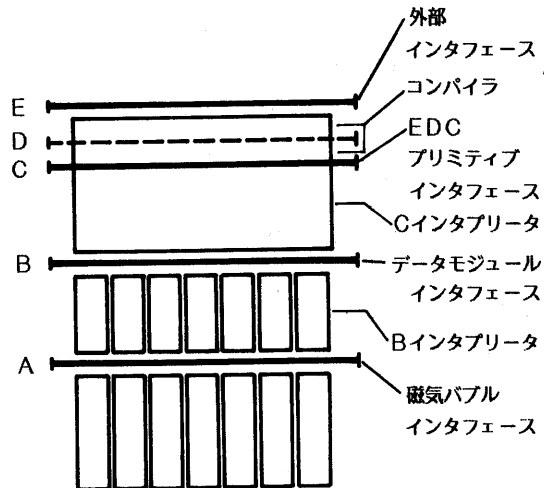


図3 EDCの抽象マシン

3 性能測定方法

性能評価は、システムを構成する各種プロセスの動作に要する時間を、ソフトウェアモニタにより測定して行なう。このためEDC本体に時間測定用のハードウェアクロックを用意した。クロックの周期は1マイクロ秒である。測定されたデータはインターフェースプロセッサ（IP）へ送られ、IPで記憶、処理される。測定データはオフラインで表示される。

3.1 動作時間の測定

動作時間の測定は、つぎの三モードで行なった。

(1) モード 0

モード0は各プロセスの累積動作時間を測定する。このモードではシステムの内部動作の詳細は測定できないが、全体の傾向、プロセスの全体に対する比重等を把握するのに有用である。又、問い合わせの性質の変化が、どのように内部に影響を与えるか等も、評価できる。

(2) モード 1

モード1はプロセスの切替えが発生した時刻を測定する。あるプロセスから別のプロセスへ制御が移される時刻、動作を開始するプロセス名、実行を開始するアドレスが記録される。

このモードでは各プロセス間の制御の流れが把握できるので、EDCソフトウェアの、より詳細な解析が可能である。また、実行開始時刻の差を取れば、あるプロセスの各実行部分の所要時間が得られる。動作開始アドレスの測定からはプロセス内のどの部分に実行所要時間が集中しているかもわかる。（このためにはプログラムリスト、リンケージリストを参照する必要がある。）このモードでの測定結果から、プロセス切替におけるオーバーヘッドが解析できる。

(3) モード 2

測定モード2は操作システムがプロセスへ制御を移す時刻、動作開始時刻、動作開始アドレスが

表1 EDCのプロセスとその大きさ

プロセス名	語数
E構文解析	2783
Eコード生成	2396
D構文解析	2759
Dコード生成	2200
Cインタプリタ	3419
Bインタプリタ *	3415
B命令管理	139
Bバンド生成	371
問い合わせ入力	201
結果出力	107
CP初期設定	219
MP初期設定	628
CP性能測定	147
MP性能測定	155
測定結果出力	195
CPOS	1809
MPOS	1811

*: 4プロセスあり
一語32ビット

測定される。測定項目はモード1と同じであるが、測定回数が多くきめ細かい。

このモードの測定のタイミングは割込み毎である。特に操作システムへの処理を依頼するSV C割込みは、プログラムによって発生されるので、ソフトウェアの論理的動作を追跡する事が可能となる。しかし、このモードは他のモードに比較して詳細なデータが取れるが、一回に取れるデータの量が限られているため測定範囲が小さいほか、全ての割込み毎にデータが取られるので論理的に重要でないデータが大半を占め、出力結果が繁雑であるといった欠点がある。

3.2 データと制御の流れ

ソフトウェアモニタの扱うデータと制御の流れは、どの測定モードについても同じである。問い合わせ言語（EDCL）による、問い合わせ入力直後から測定を開始し、問い合わせに対する結果の出力直後に測定を終了し、測定結果をIPのディスクファイルへ転送する。この目的で、制御プ

ロセッサ (CP) に性能測定制御プロセスと性能測定出力プロセスが、モジュールプロセッサ (MP) に性能測定プロセスが、追加された。CPとMPにおける、性能測定時のデータと制御の流れを図4と図5に示す。

時刻の測定は操作システムで行なう。これはプロセス切替えのタイミングが容易に獲まえられるからである。また被測定プロセスに手を加える必要がないのも利点である。しかし、全ての測定を画一的に行なうので、測定結果があらずぎたり、こまかすぎたりする欠点もある。

測定データは一旦MMに格納され、処理終了後にIPへ転送される。問い合わせ終了タイミングは、結果出力プロセスが性能測定プロセスへ通知する。性能測定プロセスは性能測定出力プロセスへ測定結果の出力開始を指示する。

性能測定制御プロセスは、測定開始を優先しておこなう為、高い優先度を割り当てられている。したがって、測定終了時には、問い合わせによって起動された優先度の低いプロセスが動作完了しないうちに、問い合わせ終了が通知されてしまう。そこで、測定結果を出力する優先度の低いプロセスを作った。性能測定制御プロセスが出力開始を指示すると、性能測定プロセスは、問い合わせに関するプロセスが動作を終了してから、出力を開始する。

4 実験データベースと質問

4.1 実験データベース

性能測定のために、1980年開催分を除く、過去七回の世界コンピュータ会議に発表された論文に関するデータベースを準備した。データベースは関係モデルで表現されており、五つの関係からなる。関係名と、各関係のタブルの構成を図6に示す。タブルは固定長であり、各関係のタブル数は表2に示すとおりである。一つの関係は図2

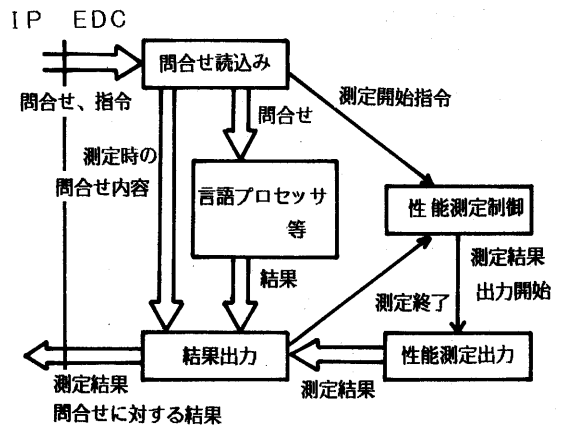


図4 性能測定システムのデータと制御の流れ (CP)

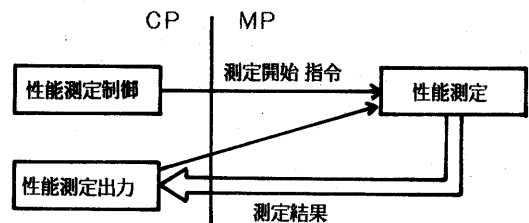


図5 性能測定システムのデータと制御の流れ (MP)

表2 関係名とタブル数

関係	タブル数	データ量
表題 (CONFR)	1500	52 (91)
キーワード (KEYR)	7994	54 (96)
著者 (AUTR)	2357	87
会議 (CONFR)	7	1
所在地 (AFFILR)	1325	57

データ量は一モジュールあたりで、Kバイト単位
()内は四モジュールに分散させた場合

に示したように、一つのバンドに記憶される。しかし、一つのタブルが二つ以上のモジュールにまたがって記憶されることはない。『キーワード』関係と、『著者』関係のタブル数が文献数より多いのは、一つの文献で多数指定されているからである。全データ量は、約1.76Mバイトである。

並列度の効果を評価するため、このデータを七モジュールに分散したものと、四モジュールに分

散したものを用意した。両者の一モジュールあたりのデータ量は、『表題』関係で52Kバイトと91Kバイトである。この二つの応答時間の比較から、並列度の効果を定量的に調べるができる。

4.2 質問とその特徴

応答時間を調べるために、表3に示す質問例を用意した。例を作成するにあたって考慮したのは処理するデータの量と出力するデータの量の、多少である。

例題1は『会議関係の全タブルを出力せよ』という意味で、処理するデータの量も出力するデータの量も少い。例題2は『キーワード関係のなかでキーワードがcomputer architectureであるものを捜して、あればそのタブルの全属性を出力せよ』というものである。条件(制限条件)にあうタブルの探索を行う。例題3は、例題2の文献の表題もあわせて出力させようとするものである。文献番号を介して二つの関係の結合(Join)をおこなわせている。例題4は、さらに著者名をも出力する。

例題5と6は、表題関係中の参考文献数の平均値と最大値を求めようとする例である。例題7は、例題2と同様であり、対象とする関係と探索条件が異なるのみである。例題3、8、9は、結合操作のデータ量依存性を調べるためのものである。

例題10は探索条件が二つの論理積となっているものである。出力データは、かなり多い。例題12と13は、出力データの量が非常に多い場合に、結合操作に要する時間を調べる例である。例題14は三重の結合操作の例、15は条件の指定が論理和の例である。

表題関係

文献番号	開始頁	頁数	参照文献数	表題
5	4	2	3	180

キーワード関係

文献番号	キーワード
5	38

数字は各フィールドの
バイト数

著者関係

文献番号	著者名	機関名	国名
5	27	149	26

会議関係

開催年度	開始月日	終了月日	開催地	国名	文献数	頁数
2	4	4	13	10	3	4

所在地関係

機関名	所在地
149	54

図6 実験データベースのタブルの構造

表3 質問例

- 1 LIST CONFR(*)
- 2 LIST KEYR(*:KEYWORD='computer architecture\')
- 3 LIST KEYR(*:KEYWORD='computer architecture\')
- 4 LIST KEYR(*:KEYWORD='computer architecture\')
- 5 LIST AVG(TITLER(NREF))
- 6 LIST MAX(TITLER(NREF))
- 7 LIST TITLER(*:NREF=364)
- 8 LIST TITLER(*:NREF=364)
- 9 LIST AUTR(*:AUTHOR='ISHII,0')
- 10 LIST AUTR(DNO,AUTHOR,AFFILIATION: NATIONALITY='JAPAN',DNO>'77000')
- 11 LIST AUTR(DNO,AUTHOR,AFFILIATION: NATIONALITY='JAPAN',DNO>'77000')
- 12 LIST AUTR(DNO,AUTHOR,AFFILIATION: NATIONALITY='JAPAN')
- 13 LIST AUTR(DNO,AUTHOR,AFFILIATION: NATIONALITY='JAPAN')
- 14 LIST KEYR(DNO:KEYWORD='DATA BASE\')
- 15 LIST KEYR(*:KEYWORD='DATA BASE\')

5 測定データの例

表4に、例題2のモード0の測定データの例を示す。各プロセスの動作に要した時間がミリ秒単位で示されている。図7はモード1の測定データを図に表現したもので、例題2の結果である。横軸は時間で、やはりミリ秒単位であり、縦軸は実行中のプロセス名である。

制御モジュールの端末制御プロセスによって読み込まれたEプログラムが、E、Dコンパイラプロセスによってまず処理される。ついで、Cインタプリタプロセスに制御が移り、同プロセスによる連想読出し命令が各データモジュールにブロードキャスト(A)され、各データモジュールの並列動作が起動される。

出力結果は、制御モジュールの要求(B)により送られ(C)、その都度制御モジュールは端末に出力(D)している。最後に、制御モジュールから実行停止命令がブロードキャスト(E)され、各データモジュールは当該識別子を持つ命令および出力データについての後処理を行う。その後、E、Dコンパイラ関係のプロセスの後処理のフェーズがある(F)。

表4 測定結果の例

プロセス名	CP	MP1	MP2	MP3	MP4	MP5	MP6	MP7
休止	0	715	711	714	711	734	733	730
初期設定	0	0	0	0	0	0	0	0
メッセージ受信	11	6	6	6	6	6	5	5
メッセージ送信	24	2	3	2	3	2	2	3
E構文解析	34							
Eコード生成	64							
D構文解析	71							
Dコード生成	169							
Cインタプリタ	1253							
端末出力	160							
バンド創製	4							
性能測定	12							
B命令管理		10	10	10	11	10	10	10
Bインタプリタ		1030	1034	1031	1033	1012	1013	1015
合計	1806	1767	1767	1767	1767	1766	1767	1767

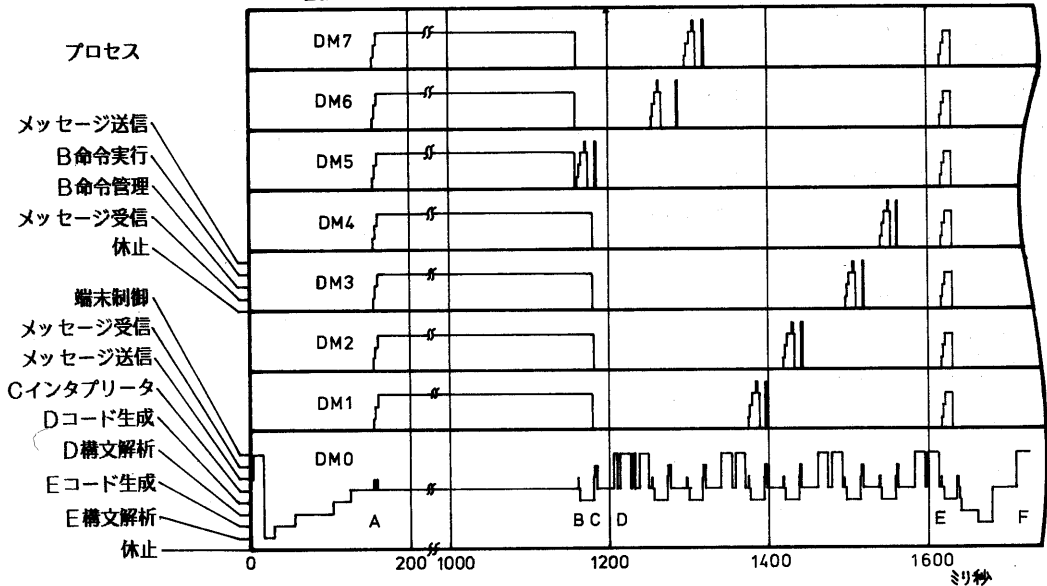


図7 EDCの実行過程

6 解析と微調整

6.1 E、Dコンパイラの動作

E、Dコンパイラは、それぞれ、構文解析とコード生成のふたつのプロセスからなっている。これらの動作時間の質問による差を調べた結果を表5に示す。コード生成プロセスの動作に要する時間は、Eコンパイラで50～70ミリ秒、Dコンパイラで170～190ミリ秒であって、大きな差は認められない。

しかし構文解析プロセスでは、かなりの差が認められる。一般に、結合を含むものや、制限条件の複雑なものが長時間を要しているが、D構文解析プロセスの、例題4、8、10、12、14に対する値が異常に長いし、10と11、12と13では、複雑な問い合わせの処理のほうが短時間となっている。

この原因を調べるため、モード1の測定データを詳細に検討したところ、各プロセスとも、磁気バブル記憶へのアクセスの前後に動作しているが、後半の後処理の動作が長時間にわたって繰り返されていることが判明した。表5には()内に前半の動作の所要時間を、いくつかの例について示した。

後処理に長時間要する理由は、多重プログラミングと、プロセスのスイッチ動作の両方が結合したためである。EDCのプロセスは、マイクロプログラムで書かれており、制御記憶の容量の関係で、常駐と非常駐のもの二種がある。E、Dコンパイラ関係のプロセスは非常駐であり、必要に応じて、MMからロードされる。ロードに要する時間はプロセスの大きさによって異なるが、おおむね20ミリ秒である。

多重プログラミングシステムでは、プロセスの実行の優先度を定めている。EDCシステムでは、C-D-Eの順、EとDの中では構文解析-コード生成の順となっている。E-D-Cと処理が進んで、磁気バブル記憶へのアクセスが開始される

表5 E、Dコンパイラの動作時間

例題	動作時間(ミリ秒)			
	E構文解析	Eコード生成	D構文解析	Dコード生成
1	30 (8.1)	63 (25)	66 (43.7)	169 (21)
2	34 (11.7)	64 (25.9)	71 (47.8)	169 (21.5)
3	62 (17.2)	48 (29.2)	100 (76.3)	187 (21.8)
4	87 (41.6)	72 (32.7)	260 (102.8)	187 (22.8)
5	50	64	68	169
6	31 (8.6)	64 (25.9)	68 (45.5)	170 (22.2)
7	34	64	71	169
8	40	68	234	187
9	81 (36.1)	48 (29.1)	145 (77.2)	187 (22.1)
10	63 (37.5)	48 (28.5)	494 (70.6)	187 (21.6)
11	224 (23.3)	51 (32)	123 (98.9)	187 (21.9)
12	57 (33.9)	67 (27.9)	1491 (65.6)	187 (21.8)
13	221	51	118	187
14	250	57	315	187
15	242	49	104	187

と、Cインタプリタはデータ待ちとなる。操作システムは、Cインタプリタが待ち状態になったので次に優先度の高い、実行可能なプロセスを選んで起動する。通常このプロセスはDコード生成プロセスであり、これがロードされ、実行される。Cインタプリタの待ちが解除されると再度Cインタプリタがロードされ実行される。このふたつの組み合わせが、データ量に対応した長さだけ繰り返される。Dコード生成プロセスの動作が終了すると、D構文解析の後処理へと移る。

こうして、E構文解析の後処理までおこなわれるが、出力データ量が少なければ、途中でCインタプリタの動作が終了し、後処理が優先的に行われ、累積動作時間が短くなる。例題10や12のようにD構文解析プロセスの実行に長時間を要しているものは、データ待ちの時間間隔と、D構文解析プロセスのロードと実行に要する時間との整合が悪く、後処理に長時間を要した。例題13は12以上の出力データ量となるが、データ待ちの時間間隔との整合性がたまたま良かったので、後処理時間が短くてすんだ。

こうした不都合を根本的に解消するためには、プロセスの優先度の変更や、待ち時間の変更のみでは不可能で、非常駐プロセスを無くする必要がある。今回のプログラム量では制御記憶の容量の

増加は非現実的とはいえないが、主記憶との記憶容量の配分を再検討する必要がある。

E, Dコンパイラの動作に関しては、後処理の問題を除けば、ほぼ予想どおりである。コード生成は、E, Dともに、質問の複雑さの影響はほとんどないが、構文解析に要する時間は、結合の多少や、制限、射影条件の複雑さと関係している。

E, Dコンパイラの動作時間の全体に対する割合は、例題2で9%、例題3で5%である、後処理の時間を差し引けば、更に半分以下になるので、ほとんど問題はない。

6.2 C, Bインタプリタの動作

C, Bインタプリタの動作時間を表6に示した。例題1, 2, 5, 6, 7, 10, 12は結合が無いので、一回の内容呼び出しで、磁気バブル記憶へのアクセスは終了する。以上の例題で、動作時間が異っているのは、データ量の差と制限、射影条件の相違による。そこで、一ブロックあたりのデータ量の処理に要する時間を算出した(表6)。この結果、制限条件の指定の長いものや、複雑なものは、処理時間が長いこと、簡単なものに比べ、50%から200%も増加することが判明した。

結合がある場合の処理時間は、最初の内容呼び出しでの出力量に関係する。これをnとすると、一般には、n回の二番目の関係への内容呼び出しが発生する。例題3の場合は、キーワード関係に一回の、表題関係に五回の内容呼び出しが行われる。最初のアクセスは例題2と同様であるから、3600ミリ秒かかり、次のアクセスには、例題7と同様の時間がかかると考えられるので、結局、 $5 \times 800 + 3600 = 7600$ (ミリ秒)と予想され、実測値と良く合っている。

例題4の場合は、結合が二回あるが、結合に使用する属性が同じなので、n回の表題関係へのアクセスと、n回の著者関係へのアクセスで済み、 $10 \times 800 + 3600 = 11600$ (ミリ秒)

表6 C, Bインタプリタの動作時間

例題	動作時間(ミリ秒)		
	C	B	B/ブロック
1	348	86	43
2	3799	3612	65.7
3	9526	8276	
4	15778	14784	
5	1417	1120	21.1
6	1400	1099	20.7
7	1122	790	14.9
8	2379	1696	
9	2929	2295	
10	2869	1096	12.5
11	31098	26343	
12	3112	3182	36.2
13	111951	91002	
14	90383	87792	
15	45546	36362	

と実測値より若干少ないが、良く一致していると思なせる。

例題8, 9は、最初の内容呼び出し出力が一件なので、 $790 + 1320 = 2110$ と計算されるが、実測では例題8がかなり短い時間で済んでいる。これは、例題8が数値の比較、9が文字列の比較であり、数値の比較が短時間で、できることによる。

以上のように、動作に要する時間は、ほぼ計算どおりと見なすことができる。しかし、表6でも明らかのように、制限条件の長い、あるいは複雑なものの動作時間が長すぎるようなので、Aインタフェースの改良を検討した。

Aインタフェースの命令は、内容呼び出し操作を実現するため強力になっている。しかし、Bインタプリタの使用するのはそのサブセットであるので、過剰な部分を除くことで、速度の向上を期待できる。そこで、(1)複数の制限条件の指定が可能であったのを単一に制限する、(2)制限条件、射影条件に関する情報の格納場所を可能なかぎりMMからプロセッサの内部レジスタに移す、(3)レジスタの保存、回復に必要な記憶領

域をPMに移す、といった改良を行なった。これらは、動作速度の遅いMMの使用をなるべく避けることで、速度の向上を図ろうとしたものである。これにより、表7に示すように、例題によってバラツキはあるが、平均して40%の速度向上を達成できた。

6.3 並列度の効果

並列度の効果を調べるために、キーワード関係と表題関係のデータを四モジュールに分散させたデータベースを作成し、例題2, 3, 5, 6について、モード0の測定を行なった。その結果のうち、Bインタプリタの動作時間と、7モジュールの値を4モジュールでの値で規格化したもの(比)を表8に示す。

一モジュール当たりのデータ量は4モジュールの場合を1とすると、7モジュールでは、表題関係が0.57、キーワード関係が0.56であるから(表2)、結合の有無に関係なく、ほぼデータ量の減少率と等しい動作時間の減少となっている。比が完全に等しくないのは、オーバーヘッドか、測定誤差のためであるが、この程度の差では、どちらに原因があるのか正確な評価はできない。

7 おわりに

試作したデータベース計算機の性能を評価するため、応答時間をソフトウェアモニタによって測定した。測定に当って、総容量1.78Mバイトの実験データベースを作成し、典型的な問い合わせの例題を作成した。測定結果の解析から、

(1) CおよびBインタプリタの応答速度がほぼ計算どおりであること

(2) プロセスのスワップを行う必要から、予測しなかった動作が見られたが、スケジューリング方法の変更もしくは、ハードウェアの改良により解決できること

(3) 少なくとも8モジュールまでは、オーバーヘッドなしに、並列動作による動作速度の改善を達成したこと

表7 微調整後のC, Bインタプリタの動作時間

例題	動作時間(ミリ秒)		改善率(%)
	C	B	
1	339	76	11.6
2	1253	1030	71.5
3	5017	4646	43.9
4	9492	9007	39.1
5	795	600	46.4
6	736	579	47.3
7	736	521	34.1
8	1515	1139	32.8
9	1625	1334	41.9
10	1308	744	32.1
11	21860	20745	21.3
12	2844	960	69.8
13	73976	68484	24.7
14	45467	44287	49.6
15	28617	26775	26.4

改善率 (旧B-新B) / 旧B

表8 並列度の効果

例題	動作時間(ミリ秒)		比
	7モジュール	4モジュール	
2	1030	1781	0.58
3	4646	7764	0.60
5	600	1038	0.58
6	579	998	0.58

(4) 一ブロックのデータの処理時間は結合がない場合、磁気バブル記憶からの物理的な読出し時間の3倍から10倍である。これはプロセッサの処理能力が多少劣っていることに原因があるが、最近の高性能なプロセッサを用いれば、これを同程度にすることは容易であることなどが分った。応答速度の絶対値については、比較の対象がないので評価できないが、百倍のデータ量でも並列度を十倍にすれば、現在の十倍の応答時間となり、例題2や3程度の問い合わせでは、10秒程度であるから、許容しうる応答速度を期待できる。通常の装置でもこの程度の性能は達成

できるであろうが、例題14や15のような高水準の問合せに対して強味を発揮すると考えられる。

Cバスへのアクセスの競合の様子、プロセッサ間通信方式の効率ならびに、アーキテクチャの妥当性などが今後評価しなければならない点として残された。

謝辞 本研究遂行に御支援いただいた、石井治ソフトウェア部長、西野博二前パターン情報部長、新田松雄記憶システム研究室長、またEDCフォームウェアの製作に御協力いただいた(株)ソーシャルサイエンスラボラトリーの中山照章、榊隆広の両氏に深く感謝致します。

参考文献

- (1) Uemura, S., Yuba, T., Kokubu, A., Oomote, R. and Sugawara, Y.: "The design and implementation of a magnetic-bubble database machine", Proc. I F I P Congress 80, 433 - 438 (1980)
- (2) 国分、大表、弓場、植村："磁気バブルデータベース計算機EDCのアーキテクチャ"、信学技報EC78-46(1978)。
- (3) 弓場、古川、植村："磁気バブルデータベース計算機EDCの言語インタフェース"、信学技法EC78-47(1978)。
- (4) 弓場、植村、国分、大表、菅原："磁気バブルデータベース計算機EDCの階層的抽象マシン系"、信学論(D)、投稿中。
- (5) 大表、弓場、宮川、菅原、国分、植村："磁気バブルデータベース計算機EDCの制御システム"、信学技法EC78-48(1978)。
- (6) 弓場、大表、植村、国分、菅原："磁気バブルデータベース計算機EDCの制御系とその動作"、信学論(D)、投稿中。