計算機アーキテクチャ 78－1
（１９８９．９．１９）

高速データベースマシン HDMのアーキテクチャ

浅野拓哉、峯村治実、中村俊一郎、武藤達也

三菱電機株式会社　情報電子研究所

現在、我々はワークステーション、パーソナルコンピュータの各種端末群をＬＡＮに接続し構成された、分散作業環境に対して、リレーショナルデータベースの供給を目的とした高速データベースマシンＨＤＭを開発中である。リレーショナルデータベースシステムでは、全件サーチ・ジョインなど計算量の大きい処理が多い、ＨＤＭでは、リレーショナルデータベース用に専用マシン化することで処理のオーバーヘッドを減らし、また複数台の68020と同数台のディスクで構成したハードウェアによって、処理の並列化を行い高速な処理を実現させている。本稿では、まずＨＤＭの概要とハードウェア構成について、その上にインプリメントされた処理方式、データの分割方式を中心に述べ、最後にＨＤＭの性能評価を加えて報告する。

An Architecture of A High Speed Database Machine:HDM

Takuya ASANO, Harumi MINEMURA, Syunichirou NAKAMURA, Tatsuya MUTOU

Information Systems and Electronics Development Lab.,
Mitsubishi Electric Corporation
5-1-1, Ofuna, Kamakura 247, Japan

We are developing a Database Machine, HDM, which has a capability of very fast database operation. It is intended to be used as a database server for the local area network. There are many heavy transactions in a relational database such as exhaustive search or Join transaction etc. HDM realized the high speed processing, by means of reducing the various overheads of general purpose cpmputer and the paralel processing with micro processors(68020) which are connected with each own disks. In this paper, we mainly discribe the architecture and the benchmark result of the HDM.

1．はじめに

ワークステーションやパーソナルコンピュータの高性能化・低価格化にともなって、その普及率も向上し、各自1台づつのマシンを保有し、それぞれをローカルエリアネットワーク（ＬＡＮ）で接続して作業を行なう分散環境が、いたるところに現われてきている。各々のマシンがデータの格納場所として、各自のマシン上にリレーショナルデータベースを作成・使用し作業を行なうと、全件サーチ・ジョイン等の重い処理については、膨大な処理時間がかかることが多く、また大規模なデータベースを共用して使用しにくい。そこでこの分散作業環境の中に、リレーショナルデータベースを一括管理し供給するデータベースサーバを接続することによって、各マシンはデータベース処理を専用マシンに分担させ、高速に実行させることができる。

我々は、フロントエンドのワークステーションと1対で、このＬＡＮ上にリレーショナルデータベースサーバを構成することができる高速データベースマシン（以下ＨＤＭ：High speed Database Machineと呼ぶ。）を開発している（図1）。

ＨＤＭは、専用のマシンとして、ハードウェア・ソフトウェアのオーバーヘッドを減らし、またリレーショナルデータベースオペレーションのタプル単位に処理が分けられる性質を利用した並列処理によって、高速性を実現したマシンである。

本稿では、まずＨＤＭの概要を説明し、次にＨＤＭのハードウェア構成を示し、その上でどういう処理が行なわれているかについて述べ、最後に性能評価を行なう。

2．ＨＤＭの概要

先にも述べたが、ＨＤＭはフロントエンドのワークステーションと接続し、このワークステーションを介してネットワーク上の様々な計算機からアクセスできるようになっている。

2．1．ネットワーク構成

図2にネットワーク構成の一例を示す。ネットワークからみるとＨＤＭとフロントエンド・ワークステーションで1つのデータベースサーバを構成している。フロントエンド・ワークステーションと他の計算機との通信は、ＵＮＩＸ系の計算機ではＴＣＰ／ＩＰプロトコル、当社のオフィス・コンピュータＭ８０とは当社の通信プロトコルであるＭＮＡ－Ｐプロトコルで行なわれる。

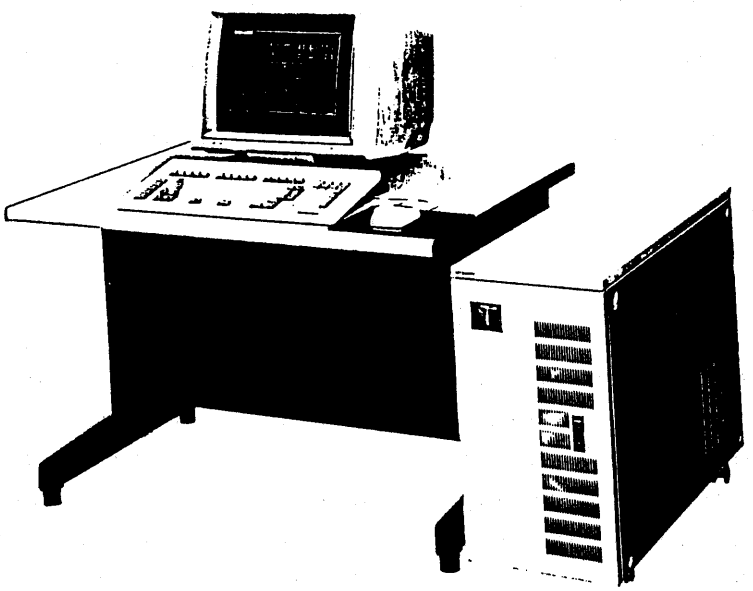

図1　高速データベースマシンＨＤＭ

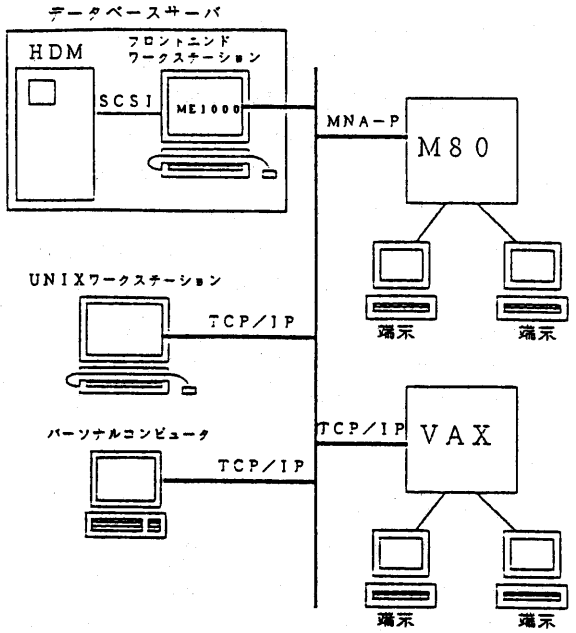

図２　ネットワーク構成例

２．２．データベースサーバ

データベースアクセスのインターフェースとして国際標準のデータベース言語であるＳＱＬ（ＩＳＯおよびＪＩＳ規格）のサブセットを採用している。ユーザはネットワークを介して、あるいはフロントエンドのワークステーションから、インタラクティブＳＱＬあるいはダイナミックＳＱＬで、データベースを操作することができる。フロントエンドのワークステーションとＨＤＭはＳＣＳＩインターフェースで接続されており、その間の通信はすべて中間言語によって行なわれる。ＳＱＬ言語と中間言語とはフロントエンドプログラムのＳＱＬパーサーで構文解析・変換される。ＨＤＭでは、中間言語の入出力によって、データベースマネージメントシステムの機能を担当している（図３）。またネットワーク機能は、フロントエンドのワークステーションのネットワークの機能をそのまま使用している。

３．ＨＤＭのハードウェア構成

ＨＤＭのハードウェアは、１枚のマスター・プロセッサと複数台（図１の装置では４台構成になって

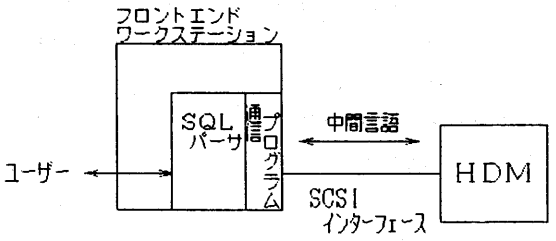

図３　データベースサーバ

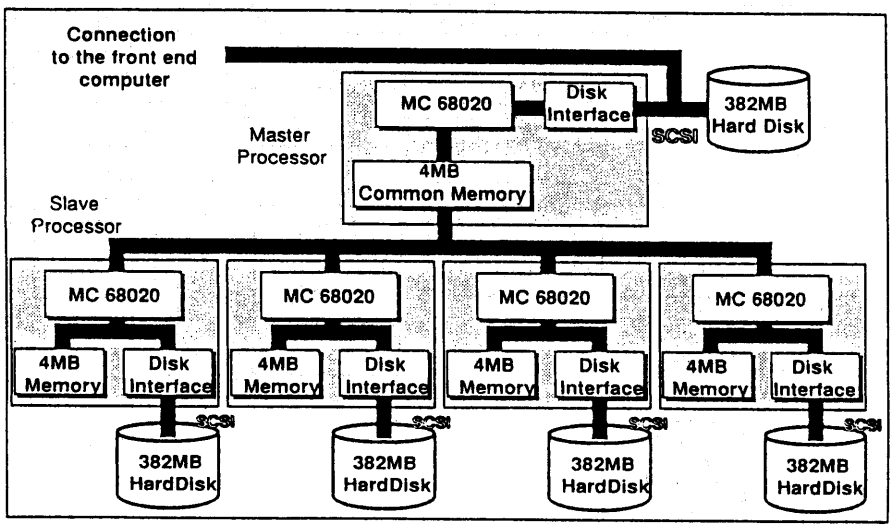

図４　ＨＤＭのハードウェア構成

いる）のスレーブ・プロセッサを１０ＭＢｐｓの高速内部バスで接続し、各々にディスク装置が付随し、これらのディスクが並列にリード・ライトできる構成になっている（図４）。各プロセッサは１枚のカード（３０㎝四方）からなり、マスター・プロセッサ、スレーブ・プロセッサともに同一のカードで実現されている。図５は１枚のプロセッサカードの構成を示している。以下に各機能と特徴を述べる。

（１）メモリ

各プロセッサは４Ｍバイトのローカルメモリを持っており、マスター・プロセッサのみ共有メモリとしても使用している。この共有メモリによってスレーブ・プロセッサとマスター・プロセッサの通信を行なっている。マスター・プロセッサは共有メモリの特定領域にデータを書き込み、送信したいスレーブ・プロセッサに割り込みを掛け読みだしてもらい、スレーブ・プロセッサも同様にして共有メモリと割り込みによってマスター・プロセッサと通信を行なう。ＨＤＭではマスター・プロセッサ上のタスクが全ての同期をとっており、スレーブ・プロセッサ同士の通信は行なっていない。

（２）プライオリティー制御

プライオリティー制御は、各々のプロセッサが共有メモリを使用する際に、調停を行なうためマスター・プロセッサのみが使用するロジックであり、１回のメモリアクセス（ロングワード）毎に共有メモリを使用するプロセッサを振り分けている。

スレーブ・プロセッサの要求が頻繁に生じた場合でも、マスター・プロセッサが共有メモリを使用できるように、スレーブ・プロセッサが何台のシステムであっても、スレーブ・プロセッサとマスター・プロセッサは必ず１回づつ交代して共有メモリを使用する。

（３）ディスクバッファ

ディスクのアクセスデータを一度バッファを通し、バースト転送することにより、ＤＭＡコントローラのバスアービトレーションの回数を減らし、ディスクのリード・ライト中でもＭＰＵ（68020）の動きが鈍くなるのをふせいでいる。

（４）ＳＣＳＩプロトコル制御ＬＳＩ

各プロセッサは、ＳＣＳＩバスを使って、ディス

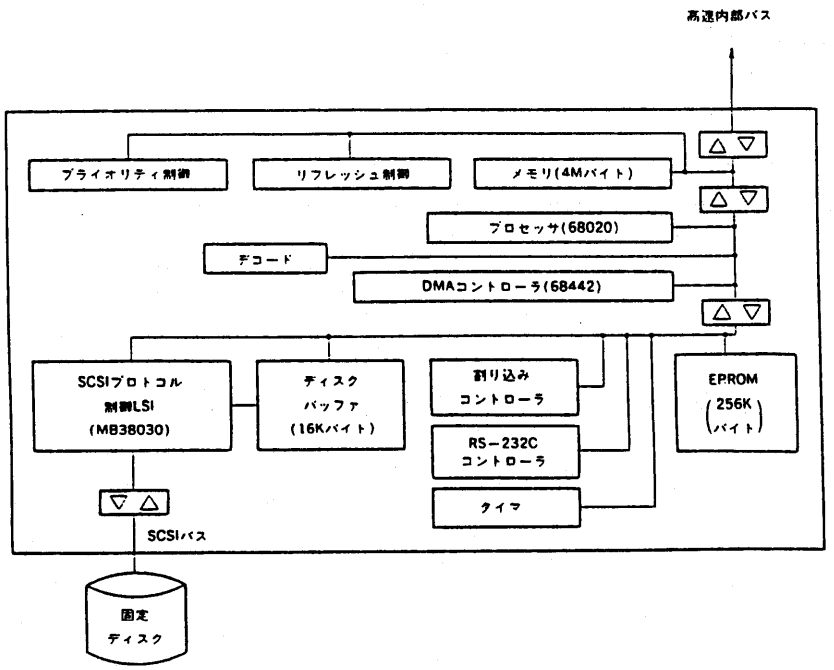

図５　プロセッサカード内ブロック図

ク装置を動作する。またマスター・プロセッサは、その他にフロントエンドのワークステーションとも同一ＳＣＳＩバスに接続され通信にも使用する。すなわちマスター・プロセッサにおいては、このＳＣＳＩプロトコル制御ＬＳＩは、ＳＣＳＩバス上でフロントエンドのワークステーションからのターゲットとして使用され、自分のディスクに対しては、イニシエータの形で使用されている。

（５）ＥＰＲＯＭ

マスター・プロセッサとスレーブ・プロセッサのカード上の違いは、このＲＯＭに書き込まれた内容だけで、あとはカードを差し込むバスケットの位置で、自らの役割を認識する。

４．データベース処理の並列化

ＨＤＭでは、トランザクションを処理するために必要なリレーションのスキーマ情報はマスター・プロセッサとスレーブ・プロセッサの両方が持っており、クラスタード・インデックスおよび非クラスタード・インデックスは、マスター・プロセッサが、リレーションのデータは各スレーブ・プロセッサが持っている。また、データベースに対する処理を並列に行なうために、１つのリレーションのタプルを各スレーブ・プロセッサに均等に配分して持っている。これを水平分割と呼ぶ（図６）。

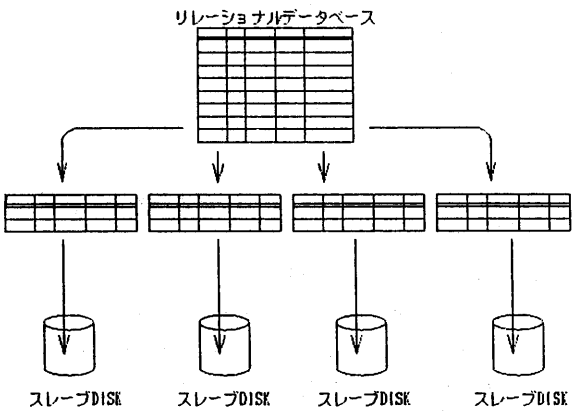

図６　データの水平分割

４．１．データベースの水平分割

ＨＤＭでは、リレーションが均等に分割された状態を保ちながらタプルを格納していくためにデータ挿入時以下のような方法が採られる。

（１）もし、リレーションがクラスタードインデックスを持っていなければ、マスター・プロセッサは、自分のディスクに入っているスキーマ情報を調べ、単にタプルをそのリレーションのタプルの数が少ないスレーブ・プロセッサのディスクに送り、格納する。

（２）もし、リレーションがクラスタード・インデックスを持っているならば、マスター・プロセッサは、自分のディスクに入っているインデックス情報を調べ、挿入されるタプルを、そのインデックスのキー値に従って決定されるスレーブ・プロセッサの所定のページに格納する。このとき、そのページが一杯であったならば、そのページの内容は２つのページに分割され、そのうち一方のページは、その時点でそのリレーションのタプルの数が最も少ないスレーブ・プロセッサのディスクに送られ、格納される。

４．２．トランザクションの処理方法

以上のように格納されたデータに対するＨＤＭでのトランザクション処理方法について、その代表的なものを述べる。

（１）サーチ（クラスタード・インデックスなし）

マスター・プロセッサは、トランザクションをスキーマ情報によってチェックし、必要ならばインデックスを調べた後、スレーブ・プロセッサに制御コマンドを送る。このとき、データの処理範囲をインデックスを用いることによって限定できるなら、そのインデックスが持つページ番号も渡される。インデックスが使えないときは、スレーブ・プロセッサは全件サーチを行なう。ＨＤＭでは、データを水平分割して格納しているので、そのような全てのタプルをサーチしなければならないような全件サーチにおいても、サーチ処理をスレーブ・プロセッサの数に比例して並列に処理することができる。

（2）サーチ（クラスタード・インデックス有り）

並列に処理するとはいっても、全件サーチ自体は非常に高負荷な処理である。しかし、多くの場合においてクラスタード・インデックスを用いることによってサーチも範囲が限定されるので、全件サーチはリレーションのある限られた範囲に対して行なうだけでよくなる。ＨＤＭでは、クラスタード・インデックスを用いると、その範囲に含まれるタプルは物理的ないくつかのページにまとまって格納され、そのようなページは、さきに述べたリレーションの水平分割化格納方式を用いているために、全てのスレーブ・プロセッサ間に散らばっている。そこで、このような範囲が限定された全件サーチの場合でもスレーブ・プロセッサに均等に負荷分散された並列処理ができる。

（3）ジョイン

ＨＤＭのジョイン処理は、インデックスを用いており、ジョインを行なうアトリビュートがインデックスを持っていない場合は、ジョイン処理を行なう前に一時的にインデックスを生成する。マスター・プロセッサは各リレーションからジョインを行なうアトリビュートのインデックスをディスクから取り出し、各要素を順に比較していくことによってインデックスのジョインを行ない、次にマスター・プロセッサは、ジョインされたインデックスからタプルの位置情報を取り出し、各スレーブ・プロセッサへタプルを取り出すためのコマンドを発行する。各スレーブ・プロセッサはディスクからタプルを取り出し、それにプロジェクション処理が必要であれば施したのち、それを共有メモリ上のキャッシュファイルに置く。最後にマスター・プロセッサはキャッシュファイル上にあるタプルをジョインする。

（4）ソート

ＨＤＭでは各スレーブ・プロセッサが並列にデータをディスクからローカルメモリに読み込み、クイック・ソートを施す。ソートされた結果は共有メモリを通してマスター・プロセッサに渡され、マスター・プロセッサがこれらのデータをマージする。

（5）多重トランザクション

データベースでは、定型業務として通常１タプルに対する更新、追加、削除かどの処理が頻繁に行なわれる。これらの処理はいずれか１つのスレーブ・プロセッサだけで処理することができる。また、このような処理が同時に要求されたとき、それらは確率的にみて各々のスレーブ・プロセッサにばらつくと考えられる。そこで、複数のスレーブ間でこれらの処理が並列に実行されることになる。

5．性能評価

5．1．評価方法

性能評価はwisconsin Benchmarkを採用し、フロントエンドとして接続された当社のエンジニアリングワークステーションME1100から、ＳＱＬ言語を用いてＨＤＭに対する問合せを発行、ＨＤＭの内部処理時間を測定することによって行なう。測定値は、全ディスクキャッシュを無効にした直後の１回目の処理と２回目の処理の２つの場合についてとる。また評価に用いたデータベースは１０００～１００万タプルの８つのリレーション（t10k～t100k）を用い、各タプルは13項目の２バイト整数、３項目の５２バイト長文字列からなる。また各項目の値は乱数によって決定される。

5．2．性能評価結果１

以下の問い合わせによって、１万タプル以下のデータベースに対して性能評価をおこなった。なおこのデータは[7]から抜きだしたものであり、比較のため、他のデータベースシステムの結果[3]～[5]も併せて示す。結果は表１に示す。

（1）インデックスなしの選択

次の２つの問い合わせによって選択演算の性能評価を行なった。

(a)選択率１％の選択演算

```
insert into tmp select * from t10k
  where unique1 between 100 and 199 ;
```

(b)選択率１０％の選択演算

```
insert into tmp select * from t10k
  where unique1 between 1000 and 1999 ;
```

ここでt10kは１万タプルのリレーション、unique1は0から9999の一意の値を持つ2バイトの整数項目である。

（2）射影

次の2つの問い合わせによって射影演算の性能評価を行なった。

```
(a)insert into tmp
      select distinct hundred from t10k ;
(b)insert into tmp
      select distinct * from t1k ;
```

ここで、hundredは、0～99の値を持つ2バイトの整数項目であり、t1kは1000タプルのリレーションである。(a)の問い合せでは、1万件のタプルがソートされ、重複除去されて100件のタプルが生成される。また(b)の問い合わせでは、実際に除去されるタプルはないが、それでも重複除去のための１０００タプルのソートは行なわれる。

（3）インデックスなしの結合

以下の3つの問い合わせによって、結合演算の性能評価を行なった。

```
(a)insert into tmp select t1.*, t2.*
     from t10k t1, t10ka t2
    where t1.uniquel = t2.uniquel
      and t2.uniquel < 1000 ;
(b)insert into tmp select t1.*, t2.*
     from t10k t1, t10ka t2
    where t1.uniquel = t2.unique2 ;
(c)insert into tmp select t1.* t2.*
     from t1k t1, t10k t2, t10ka t3
    where t1.uniquel = t2.uniquel
      and t2.uniquel = t3.uniquel
      and t2.uniquel < 1000
      and t3.uniquel < 1000 ;
```

ここでt10kaは1万タプルのリレーションである。

表1　性能評価結果1

(1) インデックスなしの選択

システム	(a)	(b)
商用INGRES	38.4 秒	53.9 秒
ORACLE	53.2 秒	72.5 秒
IDM(DAC*なし)	20.3 秒	27.2 秒
IDM(DACあり)	19.9 秒	23.4 秒
HDM(1回目)	1.79 秒	1.81 秒
HDM(2回目)	0.65 秒	0.81 秒

*DAC = Database ACcelerator

(2) 射影（重複除去）

システム	(a)	(b)
商用INGRES	26.4 秒	132.0 秒
ORACLE	29.5 秒	117.3 秒
IDM(DACなし)	58.9 秒	31.5 秒
IDM(DACあり)	33.0 秒	22.0 秒
HDM(1回目)	2.64 秒	1.73 秒
HDM(2回目)	1.50 秒	1.10 秒

(3) インデックスなしの結合

システム	(a)	(b)	(c)
商用INGRES	1.8 分	2.6 分	2.1 分
ORACLE	)300 分	)300 分	)300 分
IDM(DACなし)	9.5 分	9.2 分	2.1 分
IDM(DACあり)	1.4 分	1.9 分	0.6 分
HDM(1回目)	5.86 秒	5.88 秒	7.20 秒
HDM(2回目)	4.38 秒	4.39 秒	5.60 秒

5．3．性能評価結果2

以下の(a)～(f)までの6つの問い合わせによって1万～１００万タプルの大規模リレーション（t10k～t1000k）に対する性能評価を行なった。結果は表2に示す。表において、／の上側はディスク・キャッシュ内の全データを無効にした直後の処理時間、下側は2回目の処理時間を表している。

```
(a)インデックスなし
   select uniquel from ...
    where uniquel = 1000 ;
(b) (a)の非クラスタード・インデックス付き
(c)select stringl from ...
    where stringl like '%Xg%'
      and uniquel )9950 ;
(d)select stringl from ...
    where uniquel )9950
      and stringl like '%Xg%' ;
(e)select distinct four from ... ;
(f)select distinct string4 from  ... ;
```

ここで...はリレーション名（t10k～t1000k）を表している。

項目stringl, string4は、以下のフォーマットを持つ52バイト長の文字列である。

“①ＸＸＸ･･･ＸＸ②ＸＸＸ･･･ＸＸ③”

← 25個 → ← 24個 →

①,②,③は、それぞれａ～ｖの１文字を表している。

項目four, uniquelは、2バイト整数である。

表2　性能評価結果2

（Ｓｅｃ）

	(a)	(b)	(c)	(d)	(e)	(f)
t10k	1.19／0.30	0.41／0.15	3.17／2.53	1.17／0.32	2.48／1.18	3.67／2.75
t20k	1.81／0.50	0.42／0.15	5.73／4.95	1.80／0.53	3.81／2.38	7.09／5.71
t40k	3.12／0.91	0.57／0.16	11.01／9.82	3.12／0.95	7.36／4.84	14.36／11.96
t60k	4.16／1.30	0.60／0.15	16.04／14.68	4.19／1.39	10.65／7.29	23.23／22.90
t80k	5.41／5.11	0.68／0.16	21.31／20.98	5.43／5.12	14.26／13.96	34.36／33.98
t100k	6.59／6.29	0.75／0.15	26.43／26.15	6.61／6.31	17.71／17.41	45.36／44.96
t1000k	42.07／42.02	4.88／3.06	243.6／243.6	43.14／43.09	144.9／144.8	- ／ -

結果より、リレーションが６万タプル以下の場合（t60k以下）ディスクキャッシュの効果が非常に大きいことがわかる。また(c),(d)でわかるようにＨＤＭでは、このケースに対する最適化を行なっておらず、先に文字列の比較を行なう(c)は、先に整数の比較を行なう(d)よりも約４倍の処理時間を要している。

6．おわりに

本稿では、ＬＡＮ上に構成された分散環境に対して、リレーショナルデータベースを供給するデータベース専用マシンＨＤＭについて、そのアーキテクチャと性能について述べた。

ＨＤＭの特徴は、リレーショナルデータベースの性質を利用した負荷分散方式による並列処理と、専用マシン化によるオーバーヘッドを減らした処理によって、製品化レベルで高速性を実現した点にある。

現在ＨＤＭは、基本部の開発を終了し、ユーザアプリケーションレベルでのソフトウェアの開発を行なっている。今後、当社内において、アプリケーションプログラムを通した実使用の面から評価を行なっていく予定である

[参考文献]

[1]P.Hawthorn他 "A Database Machime for Local Area Networks", COMPCON Spring '86, (1986)

[2]J.Shemer他 "The Genesis of a Database Computer", IEEE Computer 17(1984)

[3]Bitton他 "Benchmarking Database Systems", Proc. of the 18th VLDB Conference(1983)

[4]Simon "Update to Decenber 1983 'DeWitt' benchmark", Britton Lee, Inc.(1985)

[5]ORACLE CORPORATION RESPONSE to the BENCHMARK"

[6]中村他 "高速データベースマシンＨＤＭのアーキテクチャ", 情報処理学会35回全国大会, 4Cc-6(1987)

[7]盛村他 "高速データベースマシンＨＤＭの性能評価", 情報処理学会35回全国大会, 4Cc-7 (1987)

[8]笠原他 "高速データベースマシンＨＤＭのアルゴリズム", 情報処理学会35回全国大会, 4Cc-8(1987)