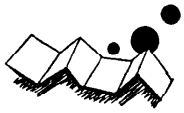
解　説



分散型データベースシステム†

植　村　俊　亮††

1. はじめに

分散型データベースシステムということばは，一見それ自身矛盾を含んでいるようにみえる．データベースシステムは，それまでばらばらに作成維持してきたデータ（ファイル）をなるべく一つに有機的にまとめて，共同利用しようという発想である．分散型システムは，それまでなるべく一つにまとめることで性能向上をはかってきた情報システムを，逆に複数の構成要素に分解していこうという発想である．どちらも1970年代の流行語であり，両者を結合した「分散型データベースシステム」もまたそうであるが，本質はどこにあるのだろうか．

本稿では，まず分散型データベースシステムの意義を検討し，データベースマシンおよび地理分散データベースシステムについて考察する．なお本学会誌ではすでにデータベースマシンに関して文献1），2），地理分散データベースシステムに関して3）が掲載されているので，本稿ではそれ以降の新しい動向や研究方向を中心に解説する．

2. 分散型データベースシステムの意義

分散型システムと総称されるものを，機能分散マシン型とネットワーク分散型とに大別するならば，データベースシステムの世界では，前者はデータベースマシン，後者は地理分散データベースシステムである．

データベースマシンということばに明確な定義はないが，増大する一方の多種多様な負荷を，単一のプロセサに一律に負担させるのではなくて，データベースシステム機能（およびデータベースそのもの）を専用のハードウェアに分担させて，情報システム全体の性能向上をはかろうとするという共通の発想をみることができる．すなわちこれはデータベースシステムという分野における機能分散マシンである．ただし専用のハードウェアといっても，単なる小型計算機から，マイクロプロセサや電子ディスク[4),5)]の組合せまで，多様な形態がある．とくに補助記憶装置まで含めた機能分散が検討されることがおおい．

データベースシステムの創始者である C. W. バックマン (Bachman) は，地理分散データベースシステムの必要性を次のように指摘している[6),7)]．

「現在では，一つの大企業のすべてのデータ処理要求を1台でまかなえるような計算機は存在しない．したがってデータ処理は複数の計算機に分散せざるをえない．現実に，フォーチュン誌に列挙される大企業500社などは，ほとんどがすでにそうなっている．さらにそのほとんどは，物理的に分散しているが，論理的に統合するにはいたっていない．地理分散データベース機能がはじめてそれを可能にする．」

この観点にたてば，いわゆるネットワーク分散型システムも，分散ということばとはうらはらに，ばらばらの計算機システムを物理的に結合するシステムであると考えることができる．地理分散データベースシステムでは，地理的に散在する計算機システム群を物理的に結合するとともに，システムが扱うデータ全体をも論理的に統合することをめざすことになる．これはデータベースシステムにおけるデータの統合を単に磁気ディスクパックの水準ではなくて，計算機システム（複数存在することがおおい）全体の水準でまとめて行おうとするものである．

3. データベースマシン

3.1 DBC

文献1）以降もいくつかのデータベースマシン提案が相つぎ，一部で実験も行われている．その中でオハイオ州立大学のシャオ教授らによる DBC (Data Base Computer) は，従来のデータベースマシン研究を統合した大規模な発想のシステムである[9)~13)]．DBC 設計にあたって，とくに検討された事項を列挙する．

† Distributed Database Systems by Syunsuke UEMURA (Computer Science Division, Electrotechnical Laboratory).

†† 電子技術総合研究所ソフトウェア部

（1） XDMS[1] に代表される後置計算機方式は，データベース機能の分散として価値があるが，そのデータベース機能が結局従来のミニコンまかせになっているという意味で十分でない．

（2） RAP[1]（次項参照）に代表される論理つきトラック方式は記憶容量面で不十分であり，価格も高くなる．

（3） 特定のデータモデルだけにしか使えないデータベースマシンでは，応用範囲がせますぎる．

以上の要請を満足すべく設計された DBC の構造を図-1 に示す．DBC は全体としては1種の後置計算機で，親計算機に接続されて動作する．DBC 自身はシャオ教授らによる属性モデルにもとづいて設計されているので，ほかのモデルたとえば関係モデルに DBC を応用したい場合には，そのソフトウェアインタフェースを親計算機上に準備する．ただしこの後置計算機は，ミニコンではなくて，RAP 型の構造記憶（structure memory）と，現行の可動ヘッド型磁気ディスク装置を改良した大記憶（mass memory）とを2層に組み合わせた新しいハードウェアである．

DBC はデータベースとして 10^{10} ビット規模の容量を想定しており，これを経済的に実現する基盤には現在使われている可動ヘッド型の磁気ディスク装置が最有力であると考える．現実的な妥協点として，この種のディスク装置の各ヘッドにプロセサを付与し，磁気ディスクの1シリンダ分のデータについて並列の内容呼出しを可能にする．異なるシリンダにわたるデータ操作には，通常のディスク装置と同じだけのヘッド移動時間が必要である．このような装置は経済的で実現も比較的容易と考えられ，わが国でも同様の提案が行われたことがある[14]．

データ操作がなるべく1シリンダの範囲ですむように，データベースの登録簿をべつに用意して，これを RAP 型の構造記憶に置く．構造記憶は大記憶のほぼ1% の容量を想定し，CCD 素子の使用を提案している．

図-1 の指令制御プロセサは親計算機から DBC コマンドを受け取り，構造記憶や大記憶を操作してコマンドを実行する．DBC はこのほかにも機密保護用フィルタプロセサなどいくつかの機能要素を含んだ，かなり複雑なシステムである．1977 年に来日したシャオ教授は実験システム構築に楽観的な見通しを述べていたが，今後の進展が注目される．

3.2 RAP. 2

XDMS システム[1]とともにデータベースマシン研究流行のきっかけとなった RAP[1),15),16)]は，その後も実験システム構築を続けており，RAP. 2 システムが報告されている[17]．RAP. 2 は最初の RAP 設計にくらべていくつかの点で改善や妥協が行われて，図-2 の構造になった．おもな変更点は，制御部にミニコン PDP 11/10 を使用していること，セル間の通信機能を削除したこと，各セルに1キロ語の入出力バッファを用意したこと，補助記憶として CCD を用いていることなどである．全体として，PDP 11/10 のユニバスにセルを二つ結合した簡単な構成になっている．

RAP. 2 で注目されるのは，セルに CCD 素子（Intel 2416, 16 キロビット/チップ）を使用し，容量1メガビットのセルを二つ実現した点である．CCD がデータベース向きであるかどうかには疑問があるけれども，電子ディスクがこうした実験システムに試用されつつあることの意義は大きい．

3.3 国内における研究開発

国内でも，いくつかの研究開発計画が進行している．東芝総合研究所では，後置計算機を2台並列に配置して，地理分散型の感触を兼ねたデータベースマシン構築を行いつつある．日電中央研究所では，後置計算機の弱点をメモリ共有によって補う，いわばバッファ型のデータベースマシン構築が進行中である[18]．これに近い方式としては，アメリカでも MADMAN[19]

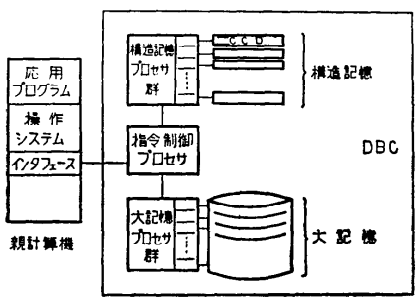


図-1 DBC の構造

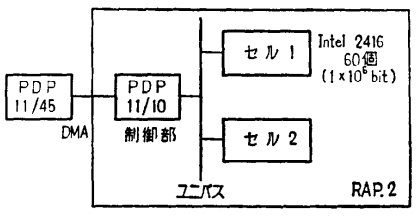


図-2 RAP. 2 の構造

システムが提唱されている.

電総研データベース計算機 EDC[20] は，電子技術総合研究所で研究開発中のデータベースマシンである(図-3). EDC は，マイクロプログラム制御マイクロプロセサ PULCE と磁気バブル素子とを組合わせたデータモジュールが複数個並列に動作する構成である. 後置計算機方式を新しいアーキテクチャで実現しようとしている点で前述の DBC に近い. 現在すでに 64 キロビットの磁気バブル素子を使った容量1メガビットのデータモジュール8基の製作調整が行われており，さらに 256 キロビットの高速磁気バブル素子によるデータモジュール製作が予定されている.

3.4 データベースマシンの意義と将来

データベースマシンは，データ処理機能の分散マシンとみなすことができる. しかし計算機がデータ処理に使われる比率の大きさを考えると，データ処理機能の分散というよりは，データ処理用の新しい計算機アーキテクチャ研究がやっと始まったと考えるべきなのかもしれない. 現在のところ，後置計算機の思想を普及させたていどにとどまり，実用化は 1980 年代に持ちこしたが，今後の動向は重要である. 筆者は次の二つの方向をとくに強調したい.

(1) 信頼できるソフトウェア作りを支援するハードウェア[21]. RAP. 2, DBC, EDC その他どのデータベースマシンも，外部の親計算機とのインタフェースをかなり高水準に設定しており，データの物理的表現，編成法などはいずれもマシンの内部にかくされている. 利用者が直接使うにしても，データベースシステム作成者が使うにしても，データベースマシンの使用は従来の磁気ディスクなどにくらべていちじるしく抽象度の高いものになる. これは信頼できるソフトウェア作りをハードウェア面から積極的に支援する方向であると考えられる.

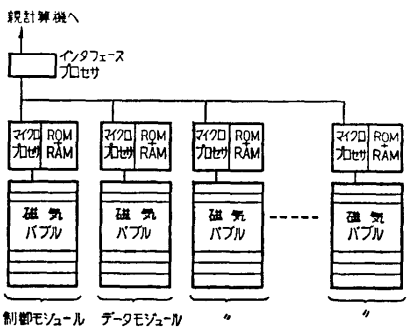


図-3 EDC の構造

(2) 電子ディスク普及のきっかけとなりうる. 計算機システムはその歴史のごく初期に，主記憶から磁気ドラムのような物理運動を伴う装置を駆逐したが，それ以降は，いわゆる呼出し時間の格差をしばしば指摘されつつも，物理運動を伴う補助記憶装置に甘んじて現在にいたっている. RAP. 2, DBC, EDC などがいずれも CCD や磁気バブルなど電子ディスクの方向を打出し，実験を行いつつあることは，きわめて注目すべきである. これらの記憶装置は物理運動をいっさい伴わないので，マイクロコンピュータ用補助記憶装置としても応用範囲が広い. とくに磁気バブルは不揮発性というデータベース向きの特性をもっている.

4. 地理分散データベースシステム

4.1 地理分散データベースシステムの意義

地理分散データベースシステムには，次に列挙する長所があることがしばしば指摘されている[3),8)].

(1) 信頼性を向上させうる. すべてを一つにまとめる方式に比べて，障害を狭い範囲にとどめることができる.

(2) システムの段階的構築，段階的増改築が可能である. 必要な部分からとりかかり，必要におうじて部分的にシステムを改造できる.

(3) 経済的である. (2)のほかにも，データの局所性を生かしたシステム作りによって，効率のよいデータ呼出しや通信経費の軽減を期待できる.

(4) 現実社会の組織構造とよく適合しているので，システムとしては受け入れられやすく，無理なく稼動させることができる.

もちろん地理分散システムには，従来の集中型システムに比べて困難な課題がいくつかある.

(1) 集中し大規模化することによって経済性を高めるという効果を期待できない.

(2) データ制御 (呼出し制御，首尾一貫性の制御)，物理データの管理体制など各面で，システムが複雑になる.

(3) 組織体にとかく不足しがちな計算機専用家を，これまでの計算センタのようなところに集中させるだけではすまなくなる.

大規模集積回路の普及によるプロセサ価格の低下が(1)の困難を緩和した. (2)は次項以降でふれるように，近年の研究の中心課題である. (3)は今後さらに重大な問題に発展する可能性がある.

4.2 課題とされてきたことと現状

地理分散データベースシステムはどうあるべきかについて，従来からいろいろの議論が行われてきた[3),6),8)]．いくつかを列挙しよう．

（1） なにを分散させるべきか，データベースそのものだけか．データベースシステム機能をも分散させるべきか．データ記述やデータ登録簿はどうするか．

（2） いかに分散させるべきか，データベースをいくつかの区分に分割して分散させるのか，同じデータをなん回も重複させて分散させるのか．

（3） 分散させる通信ネットワークの形態，性能はどうあるべきか．

（4） データモデルはなにがよいか．分散した各部分は均質なデータベースシステムにもとづくべきか．異質なものなら，いかに組み合わせるか．

（5） データ制御をどう考えるべきか．

（6） 利用者は分散をなるべく意識しないですむようにすべきであるが，それはどこまで実現可能か．

すでに述べたように，最近の地理分散データベースシステムは，最初は一つにまとまっているものを順次分散させるというよりも，散在してしまった計算機システムをネットワークに結合して，そこにデータベースシステムをのせる色合いが強い．したがってネットワークの形態，性能などは当面の研究課題として強く意識されなくなっているし，ネットワークの各節も本来独立して動くものであることが暗黙のうちに想定されている．最近の代表的な地理分散型データベースシステムと考えられる SDD-1[8),22)] の概念を図-4 に示す．DM (Data Module) は局所的なデータベースシステムである．これ自身で利用者の要求をこなせるような，独立したものを考える．SDD-1 は Datacomputer[1)] を開発した CCA 社の開発なので，DM として Datacomputer そのものを流用する計画という．TM (Transaction Module) は各 DM を論理的に結合する役目をはたす．すなわち利用者の要求を適当な

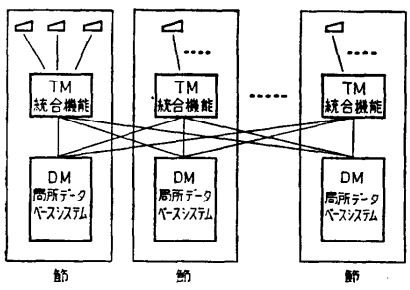


図-4 SDD-1 の構造

DM へ送り出し，結果を受けとるといった調整を行う．各 DM が異質なシステムの場合には，その相互間の変換も行うことになるかもしれない．各種の地理分散システム提案も，概要としては図-4 につきる．ネットワークの各節には，TM と DM との組合わせが一つあるいはそれ以上存在し，節は地理的に分散することになる．

この種のシステムについて，次の二つの動向をかなりはっきりうかがうことができる．

（1） データにある程度の重複を認めることが重要である．重複を認めない方式では，局所的に障害が発生した場合，システム全体はダウンしないまでも，その場所にあるデータを操作している応用プログラムは実行不可能になる．またデータの呼出し効率や通信経費などの点からみても，他の節の参照がおおくなって得策でない．高信頼性，経済性のどちらからも，**重複データを認める**方式のほうがのぞましい．

（2） 利用者はデータの分散や重複をなるべく意識しないで，あたかも一つのデータベースを使っているように分散データベースを使えるべきである．この概念を分散の不可視性 (invisibility) ということもある．これはデータベースシステムについてよく強調される**データ独立** (データベースのデータに変更があっても，応用プログラムはなるべく安定して使い続けたい) の拡張ともいえる概念であるから，筆者は**分散独立** (distribution independence) という用語を提案しておきたい．CODASYL によるデータベース記述言語 DDL が，内部スキーマ (物理データの編成法) の水準に分散の考え方を取り入れようとしているのは，この意味で正しい[25),26)]．

以上の二つの動向は，地理分散データベースシステムに特有の技術的課題を生み出しつつある．その典型的な例を次節にとり上げる．

4.3 重複データの更新

データを重複させると，こんどはデータの更新時に重複するすべてのデータを矛盾なく更新してやる必要が生じる．単一のデータベースシステムでも，更新にともなってデータをつねに首尾一貫した状態にたもつ必要があるが，ここでは**重複するデータベース相互間の首尾一貫性** (mutual consistency) が問題になる．これも分散独立と同様にシステムの責任である．

スーパーマーケットの本店と各支店との両方に支店売上げ状況を示すデータを置いて，支店側で売上げ発生のつどそれを更新するようなシステムでは，首尾一

貫性の要請はそんなに強くなくて，毎日きまった時間（たとえば開店時）にだけ保たれていれば，実用上十分であろう．銀行の預金口座データを本店や各支店に重複分散させておいて，オンライン更新するような環境では，データの首尾一貫性が常時要求される．

いま地理分散データベースシステムの三つの節に，残高＝10 000という預金口座データが存在するとしよう．一つの節では，この口座に50 000円入金し，その直後に別の節では，この口座から30 000円引き出し，さらに別の節ではこの口座の利息計算を行うとする．

```
    ⋮
残高:=残高+50000;
残高:=残高-30000;
残高:=残高*1.02;
    ⋮
```

この三つの更新は，どちらの節でも同じ順序で行われ，しかもその中間に他の節でみだりに残高を更新しないという保証がないと，更新後の各節の残高が矛盾した値になってしまう．

この種の更新は集中制御して，たとえば更新の期間中各節のすべての単価を施錠することにすれば，問題は単純になるが，分散の長所が生かせない．

R. H. トーマス (Thomas) の多数決方式は，この種の研究の初期のものであり，問題の所在を浮きぼりにするよい例である．多数決方式では，利用者の更新要求は次のように処理される．

（1） 応用プログラムが更新要求を出すと，システムはそれを直ちに受け入れるかわりに，各節に次の情報を送って，更新可否の投票を求める．

更新したいデータのもとの値とそれぞれの刻時印
更新後の値

ここで刻時印 (time stamp) は，このデータが一番最近更新されたときの時刻を示す情報で，システムが各節の各データに付与しておくものである．

（2） 各節は，更新要求のあったデータの，自分の節における刻時印を調べて，更新要求中のもとの値が意味をもっている（その後更新されていない）かどうかを判定し，次のいずれかに投票する．

（a） 意味のある更新であり，かつ判定待ちの更新と競合しないのであれば「賛成」と投票する．

（b） 一つでも無意味なデータが含まれていると，「反対」と投票する．拒否権発動である．

（c） 意味のある更新であるが，判定待ちで優先度のより高い要求の中に，この更新を無意味にする可能性が見出されると，「棄権」と投票する．

（d） これ以外の場合には，判定待ちにして投票をあとにのばし，要求があったことだけを記録しておく．

更新要求はなんらかの通信経路にそって，各節に回覧される．賛成投票が過半数に達すると，その時点で投票は終了して，各節がこの更新要求を受け入れる．反対票が1票でも投じられるか，棄権がおおくて賛成が過半数に達しないことが明らかになると，この更新要求は拒否され，応用プログラムにもそのむね通告される．

この民主的な算法は，面白いことに節相互間の首尾一貫性を保証するのみならず，判定に要する通信量もすくなく（全員の意志を確認しなくてもよい），すくみの可能性もなくて，部分的な障害にも強い．短所は投票に時間がかかること，刻時印が必要なことなどである．

多数決方式のほか，あらかじめ更新の種類を判定して影響範囲を予測する方法[22]，論理時刻による方法[24]などが提案されている．24）はこの分野をわが国ではじめて取り上げた論文である．

重複データの同時更新における首尾一貫性を保証する技術は，データそのものだけではなくて，今後データ登録簿などへの応用や，障害回復後の首尾一貫性の問題にも応用されよう．

5. おわりに

データベースシステムが内部でデータベースマシンを使用しているとか，地理的に分散しているとかの情報は，一般利用者の目に直接ふれるべきことがらではない．その意味で，利用者言語に新しい動向があまりないのは，むしろ当然かもしれない．システム内部では，データベースマシンのコマンド[1),13)~15),18),20)]，地理分散を意識した内部スキーマ記述言語[25)]などが今後のシステム構築にだんだん影響を及ぼしてくると考えられる．

地理分散データベースシステムは，これからも計算機ネットワークと結び付いた形で普及していくであろう．ネットワークの節は，現在の中小型計算機指向から，しだいに超小型（マイクロ）計算機へ移行していくであろう．そのためにハードウェア面で，超小型計算機に適した補助記憶装置開発が急務である．さらに次の段階では，ネットワークの各節が真にデータ処理を指向するデータベースマシン中心に構築されていく

ことが期待される．ここではじめて機能分散マシン型とネットワーク分散型とが結合した分散型データベースシステムが完成する．シャオ教授らは，これを情報ユティリティ (information utility)[9),26)] とよんでいる．

参 考 文 献

1) 関野，植村：データベースマシン，情報処理，Vol. 17, No. 10, pp. 940-946 (1976).
2) 前川 守：ファイルプロセサとデータベースマシン，情報処理，Vol. 18, No. 4, pp. - (1977).
3) 土井，関口：分散型データベース，情報処理，Vol. 17, No. 10, pp. 934-939 (1976).
4) 石井 治：電子ディスク，情報処理，Vol. 17, No. 12, pp. 1160-1168 (1976).
5) Chang, H.: On Bubble Memories and Relational Data Base, Proc. 4 th Int. Conf. on VLDB, pp. 207-229 (1978).
6) Distributed Data Base Technology-An Interim Report of the CODASYL Systems Committee, Proc. AFIPS NCC 1978, pp. 909-917 (1978).
7) Bachman, C. W.: Commentary on the COD ASYL Systems Committee Interim Report, ibid, pp. 919-921 (1978).
8) Rothnie, J. B., Goodman, N.: A Survey of Research and Development in Distributed Database Management, Proc. 3 rd Int. Conf. on VLDB, pp. 48-62 (1977).
9) Baum, R. I., Hsiao, D. K.: Data Base Computer A Step Towards Data Utilities, IEEE-TC, Vol. C-25, No. 12, pp. 1254-1259 (1976).
10) Hsiao, D. K. et al.: Structure Memory Design for a Database Computer, Proc. ACM 77 Annual Conference, pp. 343-350 (1977).
11) Kannan, K.: The Design of a Mass Memory for a Database Computer, Proc. 5 th Annual Symp. on Computer Architecture, pp. 44-51 (1978).
12) Banerjee, J., Hsiao, D. K.: The Use of a Database Machine for Supporting Relational Databases, ACM SIGMOD, Vol. 10, No. 1, pp. 91-98 (1978).
13) Banerjee, J., Baum, R. I., Hsiao, D. K.: Concepts and Capabilities of a Database Computer ACM TODS, Vol. 3, No. 4, pp. 347-384 (1978).
14) 武末 勝：連想ファイル記憶装置とそのデータベースシステムへの応用，情報処理，Vol. 19, No. 2, pp. 158-164 (1978).
15) Ozkarahan, E. A., et al.: Performance Evaluation of a Relational Associative Processor, ACM TODS, Vol. 2, No. 2, pp. 175-195 (1997).
16) Ozkarahan, E. A., Sevcik, K. C.: Analysis of Architectual Features for Enhancing the Performance of a Data Base Machine, ACM TODS, Vol. 2, No. 4, pp. 297-316 (1977).
17) Schuster, S. A., et al.: RAP. 2-An Associative Processor for Data Bases, Proc. 5 th Annual Symp. on Computer Architecture, pp. 52-59 (1978).
18) Hakozaki, K., Makino, T., Mizuma, M., Umemura, M. and Hiyoshi, S.: A Conceptual Design of a Generalized Database Subsystem, Proc. 3 rd Int. Conf. on VLDB, pp. 246-253 (1977).
19) Hutchison, J. S., Roman, W. G.: MADMAN MACHINE, ACM SIGMOD, Vol. 10, No. 1, pp. 85-90 (1978).
20) 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 EC 78-46 (1978)，関連して信学技報 EC 78-47, EC 78-48.
21) Myers, G. J.: Advances in Computer Architecture, John Wiley (1978).
22) Bernstein, P. A., et al.: The Concurrency Control Mechanism of SDD-1: System for Distributed Databases (The fully redundant Case), IEEE-SE, Vol. SE-4, No. 3, pp. 154-168 (1978).
23) Thomas, R. H.: A Solution to the Concurrency Control Problem for Multiple Copy Data Bases, COMPCON '78 Spring, pp. 56-62 (1978).
24) 西原，金子，鶴岡，服部：分散型データベースシステムにおける重複データ制御方式，信学技報 EC 78-6 (1978).
25) CODASYL Data Description Language Committee Journal of Development 1978, Secretariat of the Canadian Government (1978).
26) 植村俊亮：データベースシステムの基礎，オーム社 (1979).

(昭和 54 年 1 月 26 日受付)