ＤＢ 31 － 3
(1976. 11. 11)

汎用ＤＢＭＳの性能評価システムについて

米田 茂、山口康隆、吉田郁三（日立 シ研）
仁平博三（日立ソフトウェア工場）

1、 緒 言

データベースを中心としたアプリケーション、システムの普及に伴ない、データベース、マネジメント、システム（ＤＢＭＳ）に対する評価において、機能的な要求ばかりでなく、適応性の技術、特に性能に関する評価が重要視されている。特に、計算機システム建設段階で、多様なデータベース利用への対応を実現するため、ＤＢＭＳ性能の評価、検討を実施することが、データベース、システム構築上の必須手続きであると見做されて来ている。ＤＢＭＳの性能には、システムを構成する多様な要素が密接に影響し合っているが、それらを整理すると、図１のようになる。

日立製作所においては、Adaptable Data Manager（ＡＤＭ）、Practical Data Manager（ＰＤＭ）の２種のＤＢＭＳを開発している。ここでは、ＰＤＭを対象として、上記データベース、システム建設に際しての基本的技術課題を解決するため開発した、性能評価システムＰＤＭ Performance Estimation and Administration Library（PEAL）について、解析モデルを中心に報告するものである。

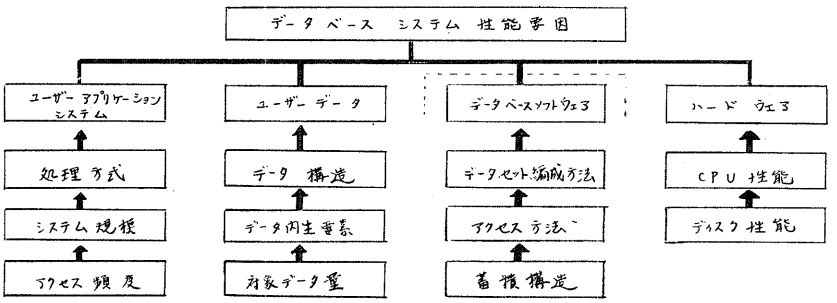


図１ データベース、システムの性能要因

2、 解析モデルの建設

2、1 ＤＢＭＳ

我々の対象としたＤＢＭＳは、マスタ、データ、セット（ＭＤＳ）、とバリアブル、データ、セット（ＶＤＳ）とゝう２つのデータ、セット編成方法を持っている。ＭＤＳは、アプリケーション、プログラムがキーを用いダイレクトにアクセスできるデータ、セットで、複数のＶＤＳと接続可能である。一方、ＶＤＳは、ＭＤＳを経由してアクセスできるデータ、セットで、複数のＭＤＳと接続可能である。

この２つの型のデータ、セットＭＤＳとＶＤＳとの接続を図ることによってＰＤＭは、データ、セット間ネットワーク構造のデータベースを形成する。

以下では、今回の解析モデル構成の主対象である、ＭＤＳについて、解析モデル構成の立場から説明を行なう。

ＭＤＳは、ランダマイズ手法を用いレコードをアクセスするデータ、セットである。ＰＤＭのランダマイズアルゴリズムは、指定されたキーの値に変換を施し、データ、セット中の個々のレコードの格納エリアを表わす相対レコード番号（RR

N）を出力する。ランダマイズの結果、異なるキーの値が等しいRRNを持つことが発生する。これをシノニムが発生したという。ランダマイズ手法のデータ・セット編成においては、シノニム・レコードの管理と、シノニム・レコード格納エリアを決定するスペース管理の方法が性能上重要な意味を持っている。

(1) シノニム・レコードの管理

シノニム・レコードの管理方法には、大別して、Open方法とChaining方法の2種がある。MDSでは、Chaining方法を採用しているが、通常のChaining方法のシノニム・レコードの管理で設定されるシノニム専用のオーバロー・エリアを持たない、というChaining方法の中にOpen方法の特長を併用した管理を行なっている。また、シノニム・レコードを結び付けるアドレス・チェーンとしては、2方向チェーンを有している。

(2) スペース管理と登録処理

MDSのスペース管理は、データ・セットの各シリンダの先頭にあるCCR（Cylinder Control Record）によって行なわれる。CCRには、どこから空エリアを探索すべきかという情報が入っており、空エリアの探索に際しては、CCRの示すアドレスより上昇順にシリアルに行なう。ここでCCRのスペース・ポインタが意味を持つのは、そのシリンダにシノニム・レコードが1つ以上存在するときである。また、その時のCCRの内容は、最も新らしく登録されたシノニム・レコードが格納されているRRNである。

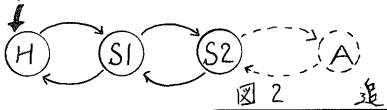
以下、MDSへの登録手順を、新レコードが入るべきエリアの状態に場合分けし、それぞれについて説明する。

a) ホーム・アドレス（ランダマイズの結果出力されるRRNのエリア）がスペース・エリアの場合。

追加すべきレコードをそのままホーム・レコードとして格納する。

b) ホーム・アドレスにホーム・レコードが格納されていた場合。

そのホーム・レコードを出発点とするチェーンの最後のレコード迄たどり、その後に追加されるレコードを結びつける。

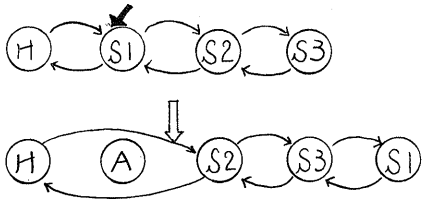


A：追加されるレコード

図2　追加処理（その1）

c) ホーム・アドレスにシノニム・レコードが格納されていた場合。

これは、追加しようとするレコードAに対して、ホーム・アドレスを求め、そこを見た時、そこをホーム・アドレスとしないレコードがすでに格納されている場合である。（図3参照）



A：追加されるレコード
＊ H：ホーム・レコード
Si：シノニム・レコード

図3　追加処理（その2）

ホーム・アドレスにあるレコードを一時主メモリ上にセイブして、追加されるレコードをそのエリアにホーム・レコードとして登録する。その後、セーブしたレコードを入れるべき空エリアを探索し、その空エリアにセー

ブした、レコードを再登録し、シノニム、チェーンを更新する。

2．2　解析モデル

チェーニング方法を用いたランダマイズ方法の解析に関し、L.R.、Johnson、Van der Pool等種々の論文が出されている。ここでは、既に出されているランダマイズ方法の解析に対し、我々の解析モデルの特徴と考えている、レコードの登録処理に関する解析モデルを、まず、ブロッキング、ファクタ（BF）が1の場合について説明し、任意のBFに拡張するに際しての要点を説明する。

2．2．1　記号の定義

N：データ、セットに収容可能な最大レコード数

n：データ、セットに現在収容されているレコード数

S：ブロッキング、ファクタ　（S＝レコード数／1ブロック）

ρ：ローディング、ファクタ　（$\rho = n/N$）

m：$m = \rho \times S$

2．2．2　BF＝1の解析モデル

解析モデルは、アプリケーション、プログラムよりレコードの追加要求が発行された時、DBMSによって行なわれる平均の入出力回数を求めることを目的とする。

(1)　確率空間の設定

(a)　P空間

N個のアドレス、スペースを持つデータ、セットにn個のレコードをランダマイズして格納するとする。このとき、1個のアドレスに着目すれば、そこにk個のレコードがランダマイズされる確率P_kは、各レコードが確率的に互いに独立に、ランダマイズされると仮定すると、

$$P_k = {}_nC_k \left(\frac{1}{N}\right)^k \left(1-\frac{1}{N}\right)^{n-k} \qquad (1)$$

という2項分布で与えられる。

よく知られているように$n/N = \rho$を一定として、$N \to \infty$、$n \to \infty$とすると、2項分布(1)は、ρをポアッソン、パラメータとするポアッソン分布(2)に漸近的に収束する。

$$P_k(\rho) = \frac{\rho^k \times e^{-\rho}}{k!} \qquad (2)$$

式(2)より明らかなことは、ランダマイズ方法によれば、シノニムの発生個数は、Nというようなデータ、セットの大きさを表わすようなものには独立で、ρによってのみ決定されるということである。

式(2)は、Xを任意のアドレス、スペースにランダマイズされているレコードの数を値としてとる確率変数とし、ポアッソン、パラメータを陽に書くと、式(3)と表現できる。

$$P_r(\rho, X=k) = \frac{\rho^k \times e^{-\rho}}{k!} \qquad (3)$$

確率分布$P_r(\rho, X=k)$の期待値$E\{X\}$はρであり、これは、1つのアドレス、スペース当たり平均ρ個のデータ、レコードがランダマイズされていることを表わす。

次に、新たな確率分布$P_r'(\rho, X=k)$を次のように定める。

$$P_r'(\rho, X=k) = P_r(X=k \mid X \neq 0) \qquad (4)$$

式(4)は、$X=0$と云う条件のもとで、$X=k$となる条件付確率を表わす。

$$P'(\rho, X=k) = \frac{P_r(\rho, X=k)}{\sum_{i=1}^{\infty} P_r(\rho, X=i)} \qquad (5)$$

確率分布$P'(\rho, X=k)$の期待値$E'[X]$が、1個のホーム・レコード当たりの平均のシノニム・レコード・リスト長となる。

$$E'[X] = \sum_{k=1}^{\infty} k P'(\rho, X=k) = \frac{\rho}{1-P_r(\rho, X=0)} = \frac{\rho}{1-e^{-\rho}} \qquad (6)$$

(b) Q空間

Q空間は、P空間より直接導かれる確率空間である。

今、ランダマイズされたレコードに対し、次の操作を行なうものとする。

i) ホーム・レコードには1、スペース・エリアには0のラベルをつける。

ii) シノニム・レコードには、シノニム・チェーンの逆方向から2、3……とラベルを付ける。

iii) シノニム・レコードをラベル0のついているエリアに移す。(図4参照)

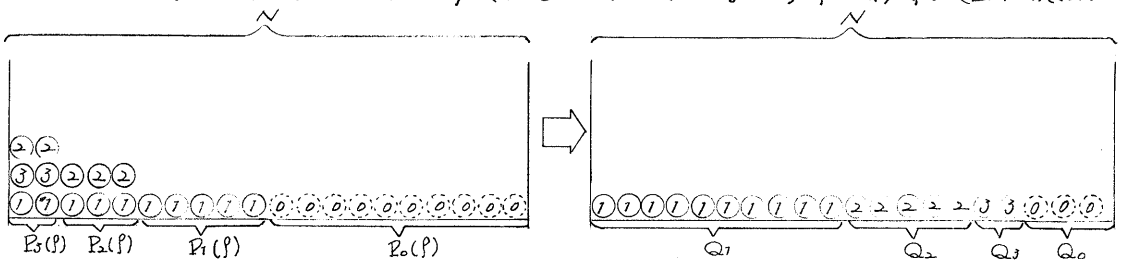


図4. Q空間の設定

前述の操作を行なった後で、Yを、任意のアドレス・スペースに割り当てられたラベルの値をとる確率変数とし、Yが値kをとる確率を、Q_kとすると、

$$Q_k = P_r(Y=k) \qquad (7)$$

となる。

このQkは、以下の確率を表現する。

$$\begin{cases} Q_0 = P_r(Y=0): \text{スペースの存在する確率} \\ Q_1 = P_r(Y=1): \text{ホーム・レコードの存在する確率} \\ \sum_{i=2}^{\infty} Q_i = P_r(Y>1): \text{シノニム・レコードの存在する確率} \\ Q_{i+1}\ (i>1): \text{シノニム・チェーンの後ろから数えてi番目に登録されたシノニム・レコードの存在する確率} \end{cases}$$

図4からも推察されるように、Q_iとP_iは次式で結ばれている。

$$\left.\begin{aligned} Q_0 &= 1-\rho \\ Q_1 &= \sum_{i=1}^{\infty} P_i(\rho) \\ &\vdots \\ Q_k &= \sum_{i=k}^{\infty} P_i(\rho) \end{aligned}\right\} \qquad (8)$$

(c) 平空間

データ・セットの先頭から、アドレスの上昇順にシリアルに空エリアを探索する時、平均何回の探索を行なえば、空エリアを見つけることができるかを考える。

Xをスペースが見つかる迄に必要な探索回数を表わす確率変数とすると、k回の探索でスペースを見つける確率 G_k は、次の幾何分布で表現される。

$$G_k = P_r(X=k) = \sigma^{k-1}(1-\sigma) \qquad (9)$$

但し $\sigma = Q_1$

式(9)の期待値が、今考えている探索回数を与えており、この期待値 $E[X]$ は、次式で表現される。

$$E[X] = \sum_{k=1}^{\infty} k\, G_k = \frac{1}{1-\sigma} = \frac{1}{1-Q_1} = \frac{1}{P_0(\rho)} \qquad (10)$$

我々のDBMSのスペース探索は、CCRのアドレスに基づいて行なわれるため、探索するエリアには、シノニム・レコードが一般的には存在せず、上記G空間が、スペース探索回数設定の確率空間として使用できることが分かる。また、前述の説明によって、DBMSのスペース探索の平均回数は、式(10)で表現できることが分かる。

(2) 解析モデルの設定

P, Q, G各空間設定に際し行なった考察を適用し、データ追加処理手続きに沿って、I／O回数の推定経過を図5に示す。

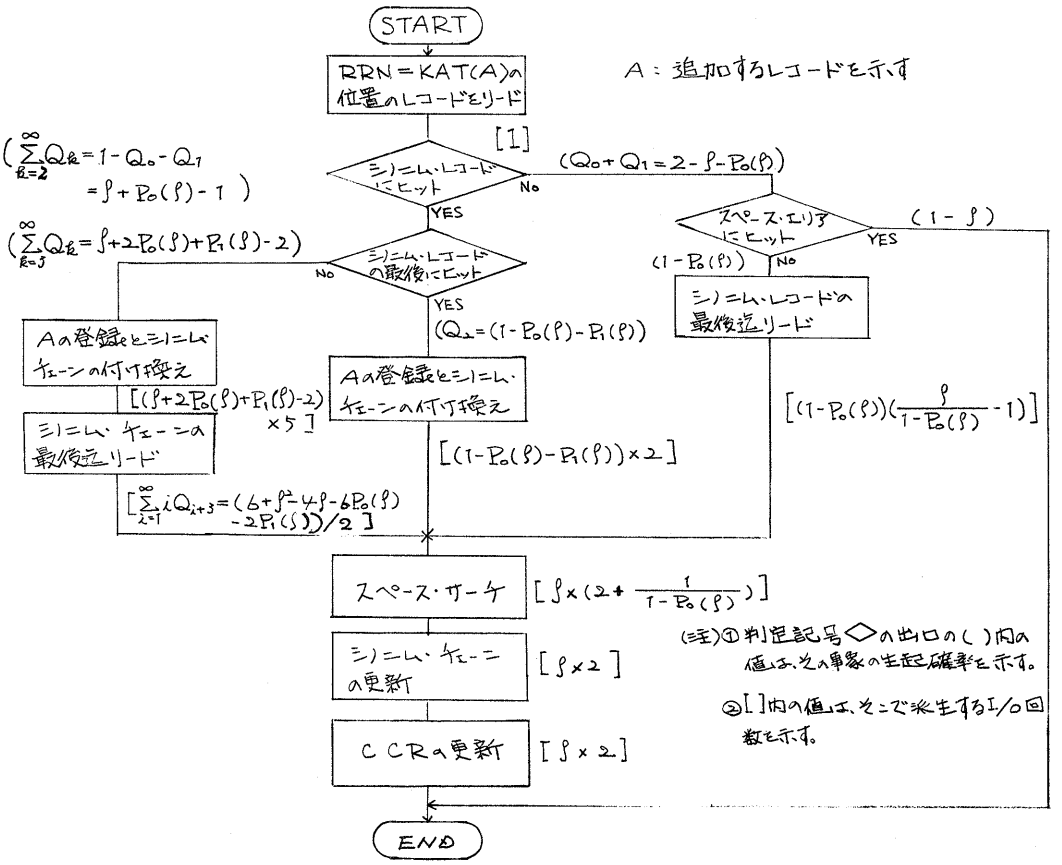


図5 MDSへのデータ追加手順とI/O回数

データ追加処理に必要なI／O回数は、図5で示した各処理区分ごとの項を足し合わせたものになるが、これは ρ のみの関数 f(ρ) として、式(11)で表現できる。

$$f(\rho) = \frac{1}{2}\rho^2 + 10\rho - 6P_0(\rho) + 2P_1(\rho)$$

$$-4 + \frac{\rho}{1-P_0(\rho)} \qquad (11)$$

式(11)を用いて、多量の追加処理を行なう場合のI／O回数の合計 T_A は、次式で求められる。

$$T_A = \sum_{i=n+1}^{a+n} f\left(\frac{i}{N}\right) \qquad (12)$$

但し、aは追加するレコード数

2、2、3 任意のBFの解析モデル

(1) フィジカルI／O回数の設定

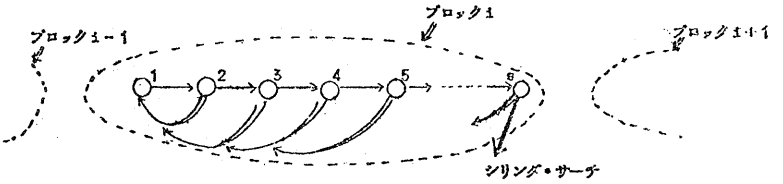
$m = \rho \times S$ をポアッソン・パラメータとするポアッソン分布を基に、I／O回数を算定する。 BFがSの場合、任意のブロックに少なくとも1レコード分以上の空エリアが存在する確率Vは、次式で表現される。

$$V = \sum_{i=0}^{S-1} P_i(m) = \sum_{i=0}^{S-1} \frac{m^i \times e^{-m}}{i!} \qquad (13)$$

フィジカルI／O回数算定のロジックは、$S=1$ の場合と殆んど同様であり、本報告では省略する。

(2) ロジカルI／O回数の設定

図6に、ブロック内スペース・サーチの手続きを示す。図6からも推測できるように、1ブロック中のデータ・エリアの位置する順序によって、空エリアの存在する確率に偏よりがある。この偏よりを反映し、ロジカルI／O回数を見積るため、ρ をポアッソン・パラメータとしたポアッソン分布を用いる。



↙シノニム最終レコードのエリアに，空きが無かった場合，当該ブロックの先頭に戻ることを示す。

→シノニム最終レコードのエリアに，空きが無かった場合のスペース・サーチの順序を示す。

#当該ブロックに空きが無かった場合，CCRの示す，エリアからスペース・サーチに入ることを示す。

図6. ブロック内スペース・サーチの手続き図

$S=1$ の場合には、ロジカルI／O回数＝フィジカルI／O回数となるが、$S \neq 1$ の場合には、この値が大きく異なっており、CPU時間算出のため、ロジカルI／O回数を算定することが、解析モデルの重要な要素となっている。

ここでは、ロジカルI／O回数推定解析モデルの特徴を最もよく表現している、スペース・サーチに際し発生する、ロジカルI／O回数推定方法の要点について記す。

ブロックの先頭から数えて j ($j=1, 2, \cdots, S$) 番目のデータ・エリアが空である確率 V_j を求める。

i) $j=1$ が空である確率（∇_1）

∇_1は、次の条件が満足される確率である。当該エリアに、レコードがランダマイズされている数が0で、かつ、本ブロック内の他のエリアには、高々1個しかレコードがランダマイズされていない。

$$\therefore \quad \nabla_1 = P_0(\rho)\left\{P_0(\rho)^{S-1} + {}_{S-1}C_1 P_0(\rho)^{S-2} P_1(\rho) + \cdots + {}_{S-1}C_{S-2} P_0(\rho) P_1(\rho)^{S-2} + P_1(\rho)^{S-1}\right\} \qquad (14)$$

ii) $j=2$ が空である確率（∇_2）

∇_2は、次の条件が満足される確率である。当該エリアに、レコードがランダマイズされている数が0で、かつ、本ブロック内の他のエリアに高々1個しかレコードがランダマイズされていない場合、および、$j=1$ のエリアが $P_0(\rho)$ の状態で、かつ、$j \geq 3$ のエリアのどれか1つに2個のレコードがランダマイズされており他のエリアには高々1個しかレコードがランダマイズされていない。

$$\therefore \quad \nabla_2 = \nabla_1 + P_0(\rho)^2 \times P_2(\rho) \times (S-2)\left\{P_0(\rho)^{S-3} + {}_{S-3}C_1 P_0(\rho)^{S-4} P_1(\rho) + {}_{S-3}C_{S-4} P_0(\rho) P_1(\rho)^{S-4} + P_1(\rho)^{S-3}\right\} \qquad (15)$$

iii) $j=3$、4、……、Sもと同様の考察によって算出可能であるが、本稿では省略する。

スペース・サーチによって、空エリアを探し出すために必要なロジカルI／O回数をLとする。CCRのアドレス・ポインタが、ブロック中のi番目のデータ・エリアを指し示す確率をT_iとした時、Lは式(16)で表わされる。

$$L = 1 + \sum_{i=1}^{S} T_i \left[\left\{\sum_{j=1}^{i} \frac{\nabla_j}{\prod_{k=j}^{i}(1-\nabla_k)}\left(\frac{S\alpha}{(1-\alpha)} + \frac{1-i+j}{(1-\alpha)}\alpha\right)\right\} + \nabla_{i+1}\left(\frac{S\alpha}{(1-\alpha)^2} + \frac{2}{(1-\alpha)}\right) + \left\{\sum_{j=i+2}^{S} \nabla_j \prod_{k=i+1}^{j-1}(1-\nabla_i)\left(\frac{S\alpha}{(1-\alpha)^2} + \frac{1-i+j}{(1-\alpha)}\right)\right\}\right] \qquad (16)$$

但し、$\alpha = (1-\nabla_1)(1-\nabla_2)\cdots\cdots(1-\nabla_S) = \prod_{i=1}^{S}(1-\nabla_i)$, $\sum_{i=1}^{S} T_i = 1$

3. 性能評価システムPEAL

3.1 開発の目標

2章で述べた解析モデルを利用し、処理能力を推定するシステムPEALを開発している。図7に、PEAL開発に際して設定した基本目標について示す。

3.2 概略仕様

図8にPEALの全体概念図、図9に判御の流れの例を示す。

以下PEAL開発の内容について、入出力情報を中心に説明する。

(1) PEALのインプット

図8に示したように、インプットとしては、データベース定義情報、業務プログラム（UAP）モデルなどがあり、図10、図11に、定義例、UAPモデル記述例を示す。この中で、特に重要と考えられるUAPモデルに関し、それを記述す

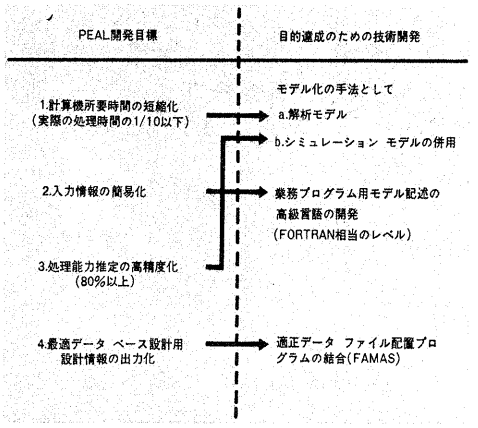


図7　PEAL開発の目標　　PEAL開発に際しての開発目標と，目標を実現するために使用した技術を表わす。

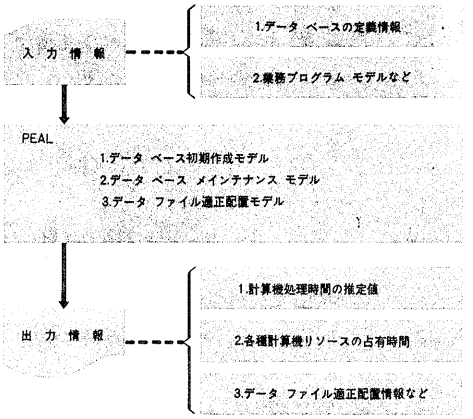


図8　PEALの概要　　データ ベースの定義情報，業務プログラム モデルなどの記述を入力すると，PEAL処理の結果として処理能力に関するデータが出力される。

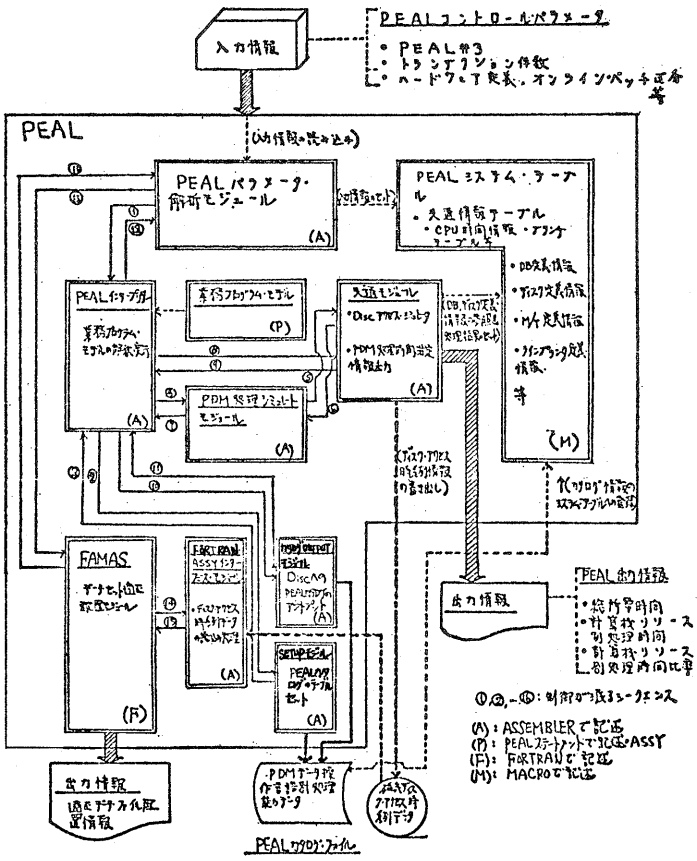


図9.　PEALの制御の流れ
（DB検索・更新の場合）

```
PLGEN FILE=H-8578,TIMER=5
DSGEN IOBUF=PRIV,DTYPE=VOS,DEVICE=H-8589,KEYLEN=18,
      BLKSIZE=3150,RECSIZE=19,BLKMEN=250
ALLOC VSN1=ETSDB1,START(1,DATA1=403
DSGEN IOBUF=PRIV,DTYPE=VOS,DEVICE=H-8589,
      BLKSIZE=11179,RECSIZE=3939,ELEMENT=10,CLL=80
ALLOC VSN1=ETSDB4,START(1,DATA1=19
LPGEN BLIN=MHEDLKP,BREC=002
DSGEN IOBUF=PRIV,DTYPE=VOS,DEVICE=H-8589,
      BLKSIZE=1745,RECSIZE=1745,ELEMEN=10,CLL=80
ALLOC VSN1=ETSDB2,START(1,DATA1=19
LPGEN BLIN=MHEDLKON,BREC=002
DSGEN IOBUF=SBGEN,DTYPE=VOS,DEVICE=H-8589,
      BLKSIZE=6300,RECSIZE=3150,ELEMEN =250,CLL=80
ALLOC VSN1=ETSDB2,START(1,DATA1=119
LPGEN BLIN=MHEDLKOD,BREC=002
DSGEN IOBUF=VDTO,DTYPE=VOS,DEVICE=H-8589,
      BLKSIZE=6300,RECSIZE=3150,ELEMEN =250,CLL=80
ALLOC VSN1=ETSDB3,START(1,DATA1=329
LPGEN BLIN=MHEDLKOK,BREC=002
DSGEN IOBUF=VDTO,DTYPE=VOS,DEVICE=H-8589,
      BLKSIZE=25200,RECSIZE=2600,ELEMEN =250,CLL=80
ALLOC VSN1=ETSDB4,START(1,150,DATA1=804
LPGEN BLIN=MHEDLKUS,BREC=002
DSGEN IOBUF=PRIV,DTYPE=VOS,DEVICE=H-8589,
      BLKSIZE=6444,RECSIZE=1719,ELEMENT=10,CLL=80
ALLOC VSN1=ETSDB4,START(1,119,DATA1=127
LPGEN BLIN=MHEDLKES,BREC=003
VLGEN VSN1=ETSDB1,DEV1=H-8589,
      VSN2=ETSDB2,DEV2=H-8589,
      VSN3=ETSDB3,DEV3=H-8589,
      VSN4=ETSDB4,DEV4=H-8589
MTGEN TYPE=INPUT,RECSIZE=1719,BLKSIZE=3444,DEVICE=H-8589,
      EPI=NO
PRGEN DEVICE=H4612
PLEND
END
```

図10.　PEAL 定義の例

るＰＥＡＬ言語を説明する。

ＰＥＡＬ言語は、ＵＡＰのモデルを記述するに際して必要な次の機能を持つコマンドを保持している。

(a) データ処理の流れを表現する、CASE, LOOP, IF, GOTOのコマンド

(b) 任意の段階で、処理時間推定情報を取り出し、出力するＳＴＩＭＥ、ＢＲＥＡＫＥのコマンド

(c) ＵＡＰのプロセスを記述するPROCESS, ＰＤＭのコマンド

```
PLSTATE  START    INPUT=MDS1,MDS2,VDS1,                    X
                  OUTPUT=TEMP,TEMP,DB01
         STIME    T1,T2
         PROCESS  C=300,GET=MT00
         CASE     (1000 P=20),(2000 P=20),(3000 P=10),     X
                  (3000 P=50)

1000     PROCESS  C=100,GET=MT00
         BREAKE   T=T1,OPT=A,PCHANG=N
         Go To    5000
         PROCESS  C=200,GET=MT00

2000     PROCESS  C=200,GET=MT00
         PDM      PROC=ADD-M,DS=MDS1,I=1,M=R
         Go To    5000

3000     PROCESS  C=300,GET=MT00
         PDM      PROC=READM,DS=MDS2,I=1,M=R
         PDM      PROC=READV,DS=VDS1,LP=MDS1LK00,I=1, X
                  M=S,V=L

5000     CONTINUE
         BREAKE   T=T2,OPT=A,PCHANG=N
         END
```

図11　ＵＡＰモデルの記述例

図11のＵＡＰモデル記述例からも分るように、ＰＥＡＬ言語は、FORTRANと同等の高級言語で構成されており、処理時間推定に際しては、図9で示されるように、ＰＥＡＬ言語を中間言語に変換し、それを解釈・実行し処理を進めている。

(2) アウトプット

図12のＰＥＡＬアウトプット・リストに示すように、ＰＥＡＬは、①キャラクタ・モード、②グラフィック・モードの２種類の情報をアウトプットすることにより、種々のＤＢＭＳ利用状況に応じて

(a) データベース処理に必要な計算機所要時間.

(b) ＣＰＵ、磁気ディスクなどの、各種計算機リソースの利用時間.

(c) 各種計算機リソースの使用比率

など、処理能力を評価するに際し重要な情報を提供する。

また、ＰＥＡＬで収集する、磁気ディスク・アクセスの時系列情報を、既に開発が完了している適正データ・セット配置プログラム Effective File Allocation Method by Automatic Synthesis (ＦＡＭＡＳ) にＡ

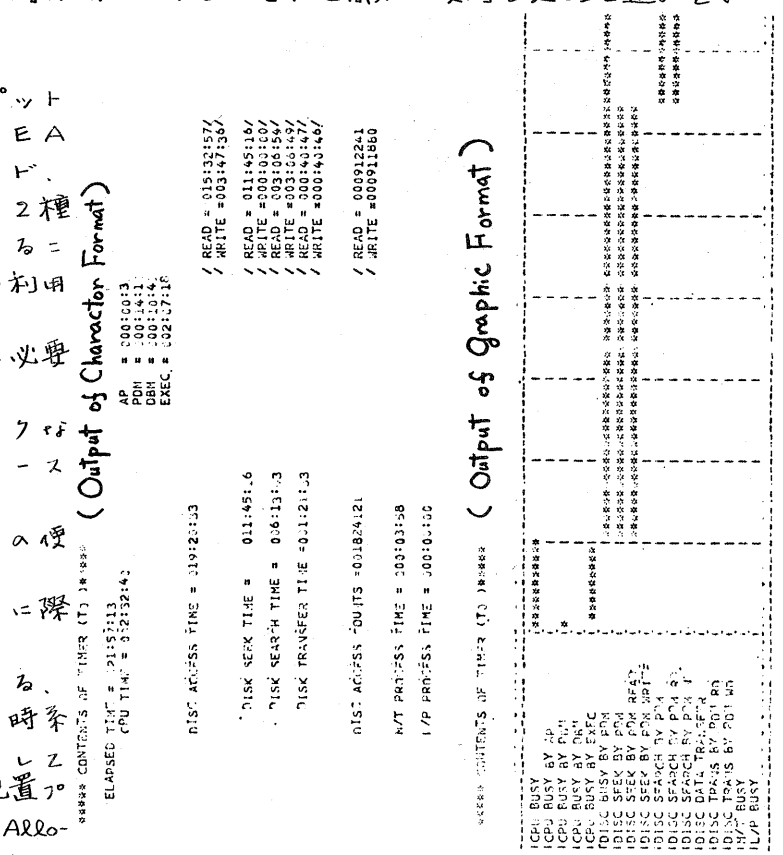


図12　ＰＥＡＬの出力例

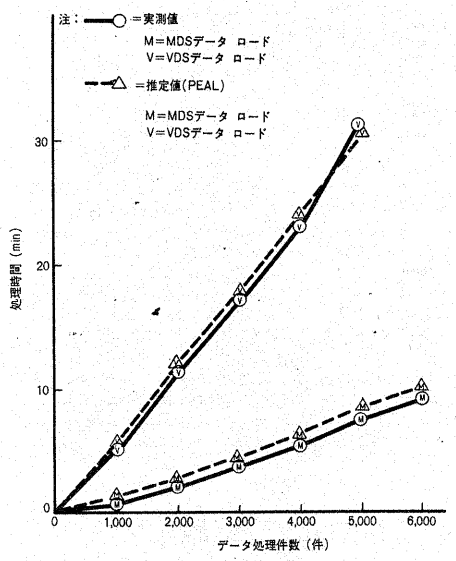


図13 PDM処理時間の推定精度　マスタ及びバリアブル各データ セットの初期作成(データ ロード)を行なうに際しての実測値データとPEALの出力結果の推定時間の比較である。

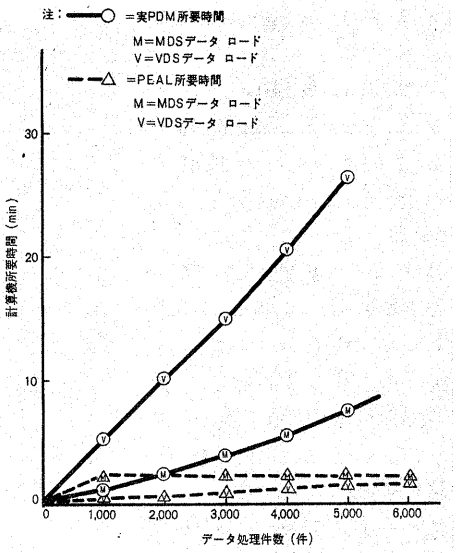


図14 PEALの計算機所要時間　PDM実データ処理の計算機所要時間と，PEALでそのモデルにおける計算機所要時間を推定する際に必要なPEAL計算機所要時間との比較を表わす。

力することによって

(d) データ・セットの適正配置

に関する情報が得られる。

(3) PEAL出力結果の推定精度、および、計算機所要時間.

図13、図14にPEALと実測データとの比較結果を示す。図13では、PEALの推定精度は95～105%になっていることが分かる。また、他の種々の利用状況でも、右図と同等の結果を得ており、精度として95%～105%である、と考えられる。

図14によって、PEAL計算機所要時間として、以下の点が把握できる。

(a) MDSの初期作成では、データ処理件数が多い程PEALの効果が大きい。
(高々1000点の代表点によって推定処理を行なっている理由による。)

(b) VDSの初期作成では、PEALは実PDMのデータ処理時間の 1/10 程度の計算機所要時間である。

また、データベースの検索・更新処理では、PEALの計算機所要時間が、実PDMの 1/20～1/30 であることを確認している。

4. 結言

本PEALの開発によってHITAC-8000シリーズ用PDMの処理能力評価に関し、次の結果を得た。

(a) オンライン、バッチ両環境における、処理能力の推定が行なえる。

(b) 適正データ・セット配置プログラムFAMASとの結合によってデータ・セットの適正配置情報が得られる。

(c) 以上の結果、データベース設計の妥当性の検証が可能となる。

今回説明した解析モデル、性能評価ツールPEALの開発でみられるように、データベース・システムの建設、運用段階でのシステムの性能、および、操作性の向上に重点をおいて、その改善を続けている。その他の性能評価の面では、シミュレータ、ソフトウェアモニタ、解析ツールなどの評価技法が蓄積されつつあるので、これらを通じ汎用的なシステム設計技法の確立を着実に推進していきたい。