

多重プロセッサシステムにおける記憶階層構成の最適化†

藤原 祥隆^{††} 伊藤 陽之助^{††}

平等多重プロセッサシステムにおける主記憶系記憶階層を対象とする最適構成解析法を提案する。本解析法はプロセッサの実行するプログラムの特性をオペレーティングシステムの特性で代表させることにより記憶階層内のメモリアクセス競合現象を単純な待ち行列モデルで表現したことを特徴としており、与えられたプロセッサ・主記憶系間の実効アクセス時間の制約条件下で主記憶系の総コストを最小とする最適構成を幾何計画法を基礎とする逐次近似法により導く。また本解析法は3階層構成において数回程度の近似回数で5%以内の変動範囲に解が収束し実用的である。

さらに本解析法を数年後のリアルタイムシステムを想定した具体例に適用し多重プロセッサ化と主記憶の多層化による主記憶系の性能価格比の改善効果について解析を行い、記憶装置の価格性能特性が上記改善効果に最も影響力のあるパラメータであり、またLSIの技術動向は改善効果を増す方向にある、等の事柄を明らかにした。

1. ま え が き

性能と信頼性の一層の向上をねらいとして、最近の大型情報処理システムではプロセッサの多重化が盛んに行われている。このような多重プロセッサシステムにおいても主記憶系がシステムの性能価格化を左右しその階層構成の最適化が重要課題である。

従来単一プロセッサシステムにおける記憶階層の最適手法としてスタックプロセッシング法¹⁾を利用した試行錯誤法および非線形計画法を利用した解析法^{2),3)}が提案されている。しかしながらこれらを記憶階層内のアクセス様態が複雑な多重プロセッサシステムに直接適用することはむずかしく、またこれに替る手法もまだ提案されていない。

ここでは精度では試行錯誤法に劣るが解析の速度において優れている解析法の立場から多重プロセッサシステムにおける主記憶系階層構成の最適手法を提案する。

多重プロセッサシステムでは複数のプロセッサは主記憶を共有しながら並列にプログラムを実行するためメモリアクセス競合が生じる。メモリアクセス競合の影響は待ち行列によりプロセッサ・主記憶系間の実効アクセス時間に反映されるが通常の多重プログラミングの条件で正確にこれを表現しようとすると待ち行列は複雑となり解析はむずかしい。本論文では以下に述べる考え方によりメモリアクセス競合問題を単純化

し、プロセッサ・主記憶系間の実効アクセス時間の制約条件の下で主記憶系の総コストを最小とする最適構成を導く。

すなわち、多重プロセッサシステムにおいてはデータの入出力に伴うファイル記憶へのアクセス、プログラムの走行に伴う前処理・後処理等のために各プロセッサは共通に利用可能なオペレーティングシステムをひんばんに利用しながら個々のアプリケーションプログラムを実行している。そこでプロセッサの実行するプログラムの特性をオペレーティングシステムの特性で代表させることができることに着目し、考慮すべきメモリアクセス競合をオペレーティングシステムの存在する領域に限定して待ち行列モデルを単純化し最適構成を導く。

本論文ではさらに得られた解析法をリアルタイムシステムに適用し、主記憶系の性能価格比とプロセッサの多重化および主記憶系の性能価格比と主記憶の多層化の関係を明らかにする。

2. 解 析 法

2.1 解析モデル

今後とも多重プロセッサシステムの主流となると考えられる負荷分担形の平等多重プロセッサシステム⁴⁾を対象とする。したがって複数のプロセッサが主記憶とオペレーティングシステム(OS)を共用する。さらに図1に示すように、主記憶系は N 個のプロセッサに専用のキャッシュメモリと m 個の主記憶ユニットより構成されるものとする。プロセッサと主記憶系の各階層との情報転送方式は一般に用いられているリードス

† Optimization of Storage Hierarchies in Multiprocessor System by YOSHITAKA FUJIWARA and YOHNOSUKE ITOH (Musashino Electrical Communication Laboratory, N. T. T.)

†† 日本電信電話公社武蔵野電気通信研究所

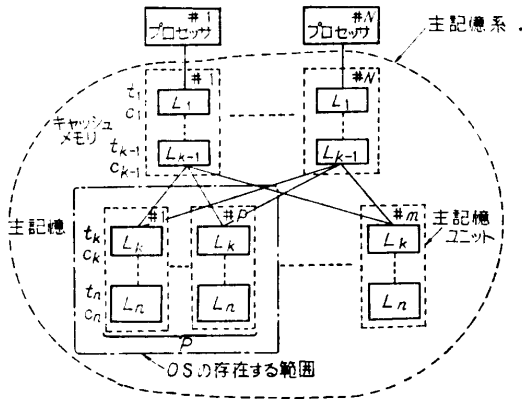


図1 多重プロセッサシステムにおける記憶階層構成
Fig. 1 Schematic of storage hierarchy in multi processor system.

ルー方式⁵⁾とする。またプロセッサと主記憶系間の実効アクセス時間はP個の主記憶ユニットにわたり配置されているOSへの実効アクセス時間で代表させ、各プロセッサはP個の主記憶ユニットを均等にアクセスするものと仮定する。

以上の前提条件の下で主記憶系に対する実効アクセス時間を与えたとき、主記憶系の総コストを最小とする構成すなわち各階層のアクセス時間と記憶容量を決定する手法を導く。

2.2 基本式の誘導

(1) 実効アクセス時間

プロセッサと主記憶系間の実効アクセス時間Tはヒット率 h_i (プロセッサが階層 L_i をアクセスする確率)とプロセッサと L_i 間の平均アクセス時間 \bar{t}_i により式(2.1)で表すことができる。

$$T = \sum_{i=1}^n h_i \bar{t}_i \quad (2.1)$$

さらにヒット率 h_i はフォルト関数、すなわちプロセッサが参照した情報が L_i に存在しない確率で以下のように表される⁶⁾。

$$h_i = f(C_{i-1}) - f(C_i) \quad (2.2)$$

フォルト関数は図2の実測例からいえるように L_i における置換えアルゴリズム、マッピング方式、ブロックサイズを与えれば、記憶容量の関数であり次式のように近似できる。

$$f(C_i) = \begin{cases} 1 & ; i=0 \\ F_i C_i^{-\alpha_i} & ; i=1, \dots, k-1 \\ F_i (P \mu_i C_i)^{-\alpha_i} & ; i=k, \dots, n-1 \\ 0 & ; i=n \end{cases} \quad (2.3)$$

ここで μ_i は実効記憶容量係数と定義するもので、

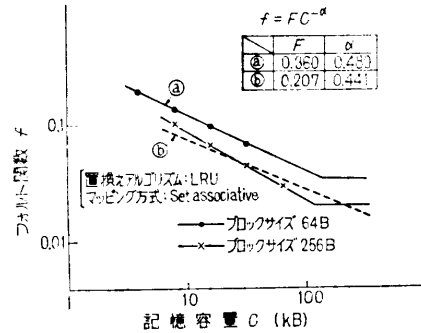


図2 リアルタイムシステムにおけるフォルト関数の例

Fig. 2 An example of fault functions in real time system.

L_i において1個のプロセッサが使用可能な記憶容量の割合を示す。

また平均アクセス時間 \bar{t}_i は装置使用率 ρ_i の関数である待ち時間比(平均待ち時間と平均サービス時間の比) $W/h(\rho_i)$ を用いて次式のように表現できる。

$$\bar{t}_i = t_i \left\{ 1 + \frac{W}{h}(\rho_i) \right\} \quad (2.4)$$

ただし $i=1, \dots, k-1$ において $\frac{W}{h}(\rho_i) = 0$

待ち時間比 $W/h(\rho_i)$ は図3の例からわかるように次式より近似できる。

$$\frac{W}{h}(\rho_i) = E_i \rho_i^{\gamma_i} \quad (2.5)$$

さらに使用率 ρ_i は L_i をヒットすることによるトラヒックと L_i より下位の L_j をヒットすることによ

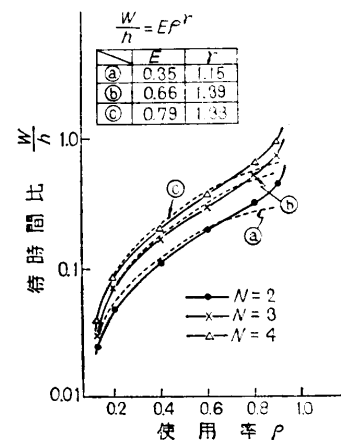


図3 M(N)/D/1モデルにおける使用率と待ち時間比の関係

Fig. 3 Relations between relative waiting time W/h and utilization factor ρ in $M(N)/D/1$.

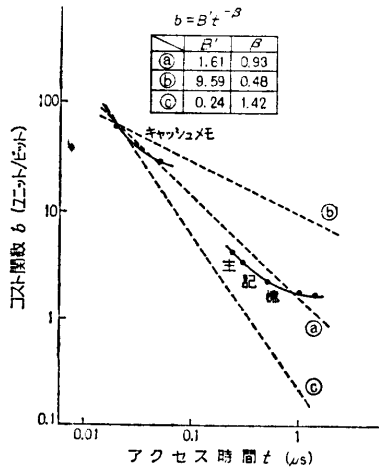


図4 コスト関数の例

Fig. 4 An example of cost functions.

り発生する L_i から L_i へのステージング動作によるトラヒックの二つの要素から次のように合成される。

$$\rho_i = NR_N f(C_{i-1}) t_i / P \quad (2.6)$$

ここで R_N は OS に対する 1 個のプロセッサ当りの平均トラヒックを表す。

(2) 総コスト

主記憶系の総コスト S は次式で表される。

$$S = N \sum_{i=1}^{k-1} b(t_i) C_i + m \sum_{i=k}^n b(t_i) C_i \quad (2.7)$$

ここで、 $b(t_i)$ はコスト関数、すなわち周辺回路および実装を考慮に入れた階層 L_i の単位記憶容量当りのコストを表すものであり、図4の容量 32 kB のキャッシュメモリと容量 1 MB の主記憶の例からも推定できるように次式で近似できる。

$$b(t_i) = B_i' t_i^{-\beta_i} \quad (2.8)$$

2.3 幾何計画法による最適解の導出

式(2.1)のプロセッサ・主記憶系間の実効アクセス時間 T の値を与えたとき、式(2.7)の主記憶系総コスト S を最小とする各階層のアクセス時間 t_i と記憶容量 C_i の値を幾何計画法により求める。

(1) 標準形式

まず式(2.1)を正項多項式 (posynomial) により近似する。すなわち最適解において各変数をとるであろう値の範囲を推定してその中から 1 組の動作点ベクトル $X^* = (t_1^*, \dots, t_n^*, c_1^*, \dots, c_{n-1}^*)$ を選ぶ。ただし最下位階層の記憶容量 C_n は既知とする。次に式(2.1)の各項を選択した動作点において関数値および 1 次偏微分係数が一致するように式(2.9)のように近似する⁷⁾。

$$H = h_i \bar{t}_i \approx K_i t_i^{a_{i,1}} C_{i-1}^{a_{i,2}} C_i^{a_{i,3}} \quad (2.9)$$

ここで

$$\begin{aligned} a_{i,1} &= \{(t_i/H) \partial H / \partial t_i\} X^* \\ &= \{1 + (t_i/1 + D_i) \partial D_i / \partial t_i\} X^* \\ a_{i,2} &= \{(C_{i-1}/H) \partial H / \partial C_{i-1}\} X^* \\ &= \{(C_{i-1}/H) \{(1 + D_i) t_i (\partial h_i / \partial C_{i-1}) \\ &\quad + h_i t_i \partial D_i / \partial C_{i-1}\}\} X^* \\ a_{i,3} &= \{(C_i/H) \partial H / \partial C_i\} X^* = \{(C_i/h_i) \partial h_i / \partial C_i\} X^* \\ K_i &= \{h_i t_i (1 + D_i) / t_i^{a_{i,1}} C_{i-1}^{a_{i,2}} C_i^{a_{i,3}}\} X^* \end{aligned} \quad (2.10)$$

$$\partial D_i / \partial t_i = \begin{cases} 0 & ; i=1, \dots, k-1 \\ E_i \{NR_N f(C_{i-1}) / P\} \tau_i \gamma_i t_i \tau_{i-1} & ; i=k, \dots, n \end{cases}$$

$$\begin{aligned} \partial D_i / \partial C_{i-1} &= \begin{cases} 0 & ; i=1, \dots, k-1 \\ E_i \{NR_N t_i / P\} \tau_i \gamma_i f(C_{i-1}) \tau_{i-1} \\ \quad \times \partial f(C_{i-1}) / \partial C_{i-1} & ; i=k, \dots, n \end{cases} \end{aligned}$$

$$\partial h_i / \partial C_{i-1} = \begin{cases} 0 & ; i=1 \\ \partial f(C_{i-1}) / \partial C_{i-1} & ; i=2, \dots, n \end{cases}$$

$$\partial h_i / \partial C_i = \begin{cases} -\partial f(C_i) / \partial C_i & ; i=1, \dots, n-1 \\ 0 & ; i=n \end{cases}$$

$$\partial f(C_i) / \partial C_i = \begin{cases} 0 & ; i=1, n \\ -\alpha_i F_i C_i^{-\alpha_i-1} & ; i=2, \dots, k-1 \\ -\alpha_i F_i \mu_i^{-\alpha_i} P^{-\alpha_i} C_i^{-\alpha_i-1} & ; i=k, \dots, n-1 \end{cases} \quad (2.11)$$

また D_i は式(2.5)の $W/h(\rho_i)$ を示す。

式(2.1), (2.7), (2.9)より最適化問題は次のように幾何計画法の標準形式で表現することができる。

$$\text{目的関数;} S = \sum_{i=1}^n B_i t_i^{-\beta_i} C_i \quad (2.12)$$

$$\text{ただし, } B_i = \begin{cases} NB_i' & ; i=1, \dots, k-1 \\ mB_i' & ; i=k, \dots, n \end{cases}$$

$$\text{制約条件;} \sum_{i=1}^n (K_i/T_0) t_i^{a_{i,1}} C_{i-1}^{a_{i,2}} C_i^{a_{i,3}} \leq 1 \quad (2.13)$$

ここで T_0 は与えられた主記憶系の実効アクセス時間を表す。

式(2.12), (2.13)の項の総数は $2n$ 個、未知数が $2n-1$ 個であるので解を一意的に決定できる。

(2) 双対な問題

次の条件を満足する双対変数 $\delta_1, \dots, \delta_{2n}$ を求める。

$$\delta_i \geq 0 \quad i=1, \dots, 2n \quad (2.14)$$

$$\sum_{i=1}^{2n} x_{i,j} \delta_i = 0 \quad j=1, \dots, 2n-1 \quad (2.15)$$

$$\sum_{i=1}^n \delta_i = 1 \quad (2.16)$$

ここで、 $x_{i,j}$ は式(2.12), (2.13)の各項の変数の指数であり、

$$[x_{i,j}] = \begin{pmatrix} -\beta_1, 0, \dots, 0, 1, 0, \dots, 0 \\ 0, -\beta_2, 0, \dots, 0, 0, 1, 0, \dots, 0 \\ \vdots \\ 0, \dots, -\beta_{n-1}, 0, \dots, 1 \\ 0, \dots, -\beta_n, 0, \dots, 0 \\ a_{1,1}, 0, \dots, 0, a_{1,3}, 0, \dots, 0 \\ 0, a_{2,1}, 0, \dots, 0, a_{2,2}, a_{2,3}, 0, \dots, 0 \\ \vdots \\ 0, \dots, a_{n-1,1}, 0, 0, \dots, a_{n-1,2}, a_{n-1,3} \\ 0, \dots, 0, a_{n,1}, 0, \dots, a_{n,2} \end{pmatrix} \quad (2.17)$$

式(2.15), (2.17)より、

$$\beta_i \delta_i = a_{i,1} \delta_{n+1} \quad i=1, \dots, n \quad (2.18)$$

$$\delta_i = -a_{i,3} \delta_{n+i} - a_{i+1,2} \delta_{n+i+1} \quad i=1, \dots, n-1 \quad (2.19)$$

また式(2.18), (2.19)より、

$$(1 + a_{i,3} \beta_i / a_{i,1}) \delta_i = -a_{i+1,2} \beta_{i+1} \delta_{i+1} / a_{i+1,1} \quad i=1, \dots, n-1 \quad (2.20)$$

さらに式(2.20)より次の関係が導かれる。

$$\delta_i = \delta_n (-1)^{n-i} \prod_{j=i}^{n-1} \frac{1}{(1 + a_{j,3} \beta_j / a_{j,1})} \cdot \frac{a_{j+1,2} \beta_{j+1}}{a_{j+1,1}} \quad i=1, \dots, n-1 \quad (2.21)$$

すると式(2.16), (2.21)より δ_n は式(2.22)により求められる。

$$\delta_n = \left[1 + \sum_{i=1}^{n-1} (-1)^{n-i} \cdot \prod_{j=i}^{n-1} \frac{1}{(1 + a_{j,3} \beta_j / a_{j,1})} \times \frac{a_{j+1,2} \beta_{j+1}}{a_{j+1,1}} \right]^{-1} \quad (2.22)$$

また式(2.18)より、

$$\delta_{n+i} = \beta_i \delta_i / a_{i,1} \quad i=1, \dots, n \quad (2.23)$$

一方、下記により式(2.14)が満足される。すなわち $\gamma_i > 0$, $E_i > 0$, $\alpha_i > 0$, $F_i > 0$, $C_{i-1}^* < C_i^*$ であるので式(2.10), (2.11)の関係から $a_{i,1} > 1$ ($1 \leq i \leq n$), $a_{i,2} < 0$ ($2 \leq i \leq n$), $a_{i,3} \geq 0$ ($1 \leq i < n$) となる。また $\beta_i > 0$ であるので式(2.20), (2.23)より $\delta_1, \dots, \delta_{2n}$ はすべて同符号となる。さらに式(2.16)を満足するように $\delta_1, \dots, \delta_n$ の値を決定するので、これらの双対変数の符号はすべて正となる。以上式(2.21)から(2.23)によりすべての双対変数を決定することができる。

また本来の問題 S の最小値 S^* は式(2.24)で与えら

れる。

$$S^* = \lambda^n C_n \delta_n T_0^{-\lambda} \left(\prod_{i=1}^n B_i \delta_i \right) \left(\prod_{i=1}^n K_i \delta_{n+i} \right) \prod_{i=1}^{2n} \left(\frac{1}{\delta_i} \right)^{\delta_i} \quad (2.24)$$

ここで、 $\lambda = \sum_{i=1}^n \delta_{n+i}$

(3) 最適アクセス時間, 最適記憶容量

最適解において本来の問題と双対な問題の間に次の関係が成り立つ。

$$B_i t_i - \beta_i C_i = \delta_i S^* \quad i=1, \dots, n \quad (2.25)$$

$$K_i t_i^{a_{i,1}} C_{i-1}^{a_{i,2}} C_i^{a_{i,3}} / T_0 = \delta_{n+i} / \lambda \quad i=1, \dots, n \quad (2.26)$$

式(2.25), (2.26)より最適アクセス時間 t_i と最適記憶容量 C_i を求める式が得られる。

$$t_n = (B_n C_n / \delta_n S^*)^{1/\beta_n} \left\{ \begin{aligned} & (-1)^{n-i} l_n t_n \prod_{j=i+1}^n \frac{(a_{j,1} + a_{j,3} \beta_j)}{a_{j,2} \beta_{j-1}} \\ & + \frac{m_{i+1}}{a_{i+1,2} \beta_i} \quad ; i=n-1 \\ & (-1)^{n-i} l_n t_n \prod_{j=i+1}^n \frac{(a_{j,1} + a_{j,3} \beta_j)}{a_{j,2} \beta_{j-1}} \\ & + \frac{m_{i+1}}{a_{i+1,2} \beta_i} + \sum_{j=i+2}^n (-1)^{j-i-1} \cdot \frac{m_j}{a_{j,2} \beta_{j-1}} \\ & \times \prod_{k=i+1}^{j-1} \frac{(a_{k,1} + a_{k,3} \beta_k)}{a_{k,2} \beta_{k-1}} \quad ; i=1, \dots, n-2 \end{aligned} \right. \quad (2.27)$$

ここで、

$$\left. \begin{aligned} m_1 &= l_n \delta_{n+1} - l_n \lambda - a_{1,3} l_n \delta_1 - l_n (K_1 / T_0) \\ & + a_{1,3} l_n B_1 - a_{1,3} l_n S^* \\ m_i &= l_n \delta_{n+i} - l_n \lambda - a_{i,2} l_n \delta_{i-1} - a_{i,3} l_n \delta_{i-1} \\ & - l_n (K_i / T_0) + a_{i,2} l_n B_{i-1} + a_{i,3} l_n B_i \\ & - (a_{i,2} + a_{i,3}) l_n S^* \quad ; i=2, \dots, n-1 \\ m_n &= l_n \delta_{2n} - l_n \lambda - a_{n,2} l_n \delta_{n-1} - l_n (K_n / T_0) \\ & + a_{n,2} l_n B_{n-1} - a_{n,2} l_n S^* \end{aligned} \right\} \quad (2.28)$$

$$C_i = t_i \beta_i \delta_i S^* / B_i \quad i=1, \dots, n-1 \quad (2.29)$$

また解の精度を上げるには式(2.27), (2.29)より得られた近似解を次の動作点ベクトルとして選び、所定の変動範囲に解が収束するまで解析を繰返せばよい。

(4) 数値例

上記の解析法を数年後のリアルタイムシステムを想定した数値例に適用して収束速度を調べた。図5に初期動作点の代表的な組合せとそれぞれ第1回目の解析結果を示す。これらの結果から初期動作点の選び方が

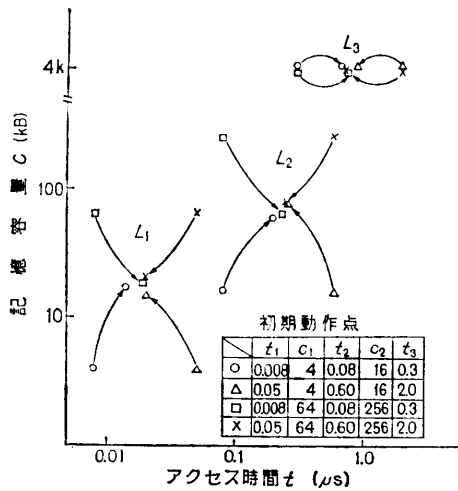


図5 4組の初期動作点と得られた第1次近似解
Fig. 5 Four sets of initial operating points and their first derived results.

収束速度に与える影響は小さく、3階層構成 ($n=3$) の場合には数回程度の逐次近似回数で5%以内の変動範囲に収束することがわかった。

3. 多重プロセッサシステムにおける性能価格比の改善効果

プロセッサ数の増加および主記憶の多層化による主記憶系の性能価格比の改善効果を前章で導いた解析法を用いシステム条件、コスト関数、フォルト関数の種々の組合せの下で明らかにする。ここでプロセッサ・主記憶系間の実効アクセス時間を T 、プロセッサ数を N 、主記憶系の総コストを S とした場合、主記憶系の性能価格比を $N/T \cdot S$ と定義する。

3.1 パラメータの設定

表1のシステム条件に関するパラメータのうち、

表1 パラメータ値
Table 1 Parameter values.

パラメータ種別	設定値
システム条件	$N=2, k=2, n=3, m=4, P=1, C_3=4\text{ MB}$ $T_0=60\text{ ns}, R_N=7 \times 10^6\text{ 回/sec}, \mu_1=\mu_3=1.0, \mu_2=0.7$
フォルト関数	$L_1: F_1=0.36, \alpha_1=0.48$ $L_2: F_2=0.24, \alpha_2=0.44$
コスト関数	$L_1: B_1'=0.09, \beta_1=1.69$ $L_2: B_2'=1.23, \beta_2=1.36$ $L_3: B_3'=0.67, \beta_3=0.59$
待ち時間比	$L_2: E_2=0.35, \gamma_2=1.15$ $L_3: E_3=0.35, \gamma_3=1.15$

C_n, k, μ_i の値を固定し他の m, P, T_0, R_N 等の値を表1の値を中心に $\pm 100\%$ 程度変化させる。またフォルト関数は表1の値を中心に、コスト関数は図4の曲線③を中心に係数部または指数部の値を $\pm 50\%$ 程度同じ割合だけ変化させる。ここで式(2.12)よりコスト関数の係数部の変化は最適解に影響を与えないため、またフォルト関数の係数部の変化は式(2.1), (2.2), (2.3)より等価的に T の変化としてみるため指数部の変化のみを考慮する。また利用率と待ち時間比の関係は主記憶には回転待ちのないランダムアクセスメモリを用いることとプロセッサと各階層間の情報転送量が一定のブロックであることを考慮して、到着分布が指数分布である有限入力源(N)をもつサービス時間が一定分布である単一窓口モデル($M(N)/D/1$ モデル)の場合の近似曲線(図3の③, ④, ⑤, etc.)を用いる。

3.2 プロセッサ多重化の効果

主記憶系の性能 N/T およびオペレーティングシステムに対する総トラヒック $N \cdot R_N$ を一定に保ちながらプロセッサ数 N を変化させて性能価格比 $N/T \cdot S$ を算出した結果、プロセッサ数が増加するに従い主記憶系の性能価格比は改善され、また改善の程度は以下のように各種のパラメータの値に依存することが明らかとなった。主要パラメータの感度係数、すなわち特定のパラメータの値が100%変化したときの $n=2, N=1$ の基準構成に対する相対性能価格比の変化の割合の一例を表2に示す。また相対性能価格比に与える影響の最も大きいコスト関数をパラメータとして、プロセッサ数と相対性能価格比の関係を図6に示す。コスト関数の指数部の変化が多重プロセッサの相対性能価格比に最も影響を与え、勾配が急なほどプロセッサ数を増やすことによる効果が大きい。例えば、勾配が100%変化すると相対性能価格比がプロセッサ数 N が2の

表2 主要パラメータの感度係数
Table 2 Sensitivity of main parameters.

パラメータ	N			
	2	3	4	
システム条件	P	0.04	0.08	0.11
	m	0.00	0.00	0.00
	T_0	-0.05	-0.10	-0.12
	R_N	-0.05	-0.12	-0.16
コスト関数	β	0.40	0.70	0.92
フォルト関数	α	-0.12	-0.18	-0.22

ただし $n=2, k=2$ とする。

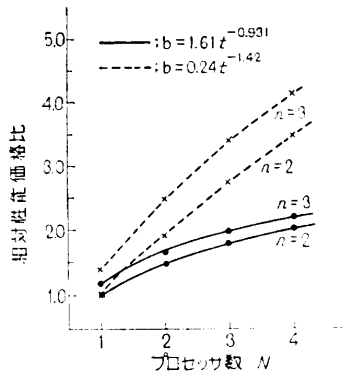


図 6 プロセッサ数と相対性能価格比の関係
Fig. 6 Relation between relative performance cost and number of processor.

表 3 主記憶階層数と相対性能価格比の関係
Table 3 Relations between relative performance cost and hierarchy levels.

n	β	0.5	1.0	1.5
2		1.4	2.0	3.4
3		1.3	2.2	4.2

ただし $N=4$, $k=2$, $m=6$, $P=1$ とする。

場合に約 40%, 4 の場合に約 90% 変化する。

3.3 主記憶多層化の効果

主記憶の階層数が 1 の場合と 2 の場合について主記憶系の性能価格比を比較した結果、多層化による性能価格比の改善効果は各種パラメータの値に依存し、かつ影響を与えるパラメータの順位はプロセッサの多重化による性能価格比の改善効果に影響を与えるパラメータの大小の順位に一致した。表 3 に影響力の最も大であるコスト関数をパラメータとして階層数増加と相対性能価格比の改善効果の関係を示す。これよりコスト関数の勾配が急なほど多層化が有利となる傾向にあることがわかる。

なお現状の装置構成技術の延長上で構成した記憶装置の性能対コスト特性 (図 4 の④) のコスト要因を分析すると記憶素子に比べ制御回路・実装部分の占める割合が大きい。またこの傾向は低コスト、低アクセス時間の記憶素子を用いる記憶装置ほど具体的にはキャッシュメモリより主記憶で顕著である。しかし一方、論理と記憶を同一チップ (あるいはウエハ) 上で実現しようとする大規模 LSI の研究、あるいは LSI 向きの新しい実装技術の研究が急速に発展しており⁸⁾、将来、制御回路・実装部分のコストが大幅に低下するものと考えられる。これは図 4 の④と比較すると、コ

スト関数の勾配が急峻になり④に近づくことと等価である。また図 4 の④は主記憶ユニットの容量を 1 MB として算出したが、将来記憶ユニットの容量がさらに増加すれば制御回路の共通化等により低コスト化し、図 4 の④、⑤と比べるとさらに勾配が急峻となると考えられる。したがって今後の技術動向を考慮するとプロセッサの多重化と主記憶の多重化がますます有利であるといえよう。

4. むすび

平等多重プロセッサシステムにおける主記憶系記憶階層を対象とする最適構成解析法を新たに導いた。本解析法は幾何計画法を利用する逐次近似解法であり、3 階層構成の数値例において数回程度の近似回数で 5% 以内に解が収束し実用的である。

また本解析法を数年後のリアルタイムシステムを想定した具体例に適用し、多重プロセッサ化と主記憶の多層化による主記憶系の性能価格比の改善効果を明らかにした。すなわち多重プロセッサ化による性能価格比との改善効果と主記憶多層化による性能価格比の改善効果に影響を与えるパラメータの大小の順位は一致し影響力の最も大きいパラメータは装置のコスト関数である。また今後の LSI 技術、実装技術の動向を考慮すると、主記憶系に関してはプロセッサの多重化、主記憶の多層化が性能価格比の点で有利であることも明らかにした。

本解析法により多重プロセッサシステムの主記憶系の概略設計を効率的に行うことができる。さらに最適化の精度を上げるためには OS 以外のプログラムの走行条件を考慮に入れて解析することが必要であろう。

参 考 文 献

- 1) Mattson, R. L. and Gecsei, J.: Evaluation Techniques for Storage Hierarchies, IBM Syst. J., Vol. 9, No. 2, pp. 78-117 (1970).
- 2) Chow, C. K.: On Optimization of Storage Hierarchies, IBMJ., Vol. 18, No. 3, pp. 194-203 (1974).
- 3) MacDonald, J. E. and Sigworth, K. L.: Storage Hierarchy Optimization Procedure, IBMJ., Vol. 19, No. 2, pp. 133-140 (1975).
- 4) 村岡洋一: マルチプロセッサと並列処理, p. 354, 近代科学社 (1970).
- 5) Madnick, S. E.: Design of General Hierarchical Storage System, 1975 IEEE Intercon Conference Record, Session 20, pp. 1-7 (1975).

- 6) Gecsei, J. : Determining Hit Ratios for Multi Level Hierarchies, IBMJ., Vol. 18, No. 4, pp. 316-327 (1974).
- 7) Duffin, R. J., Peterson, E.L., and Zener, C. : Geometric Programming, P. 278, John Wiley & Sons, Inc. (1967).
- 8) 柱木久雄, 寺島 諒: 高集積化への新しい技術, 電子通信学会誌, Vol. 60, No. 11, pp. 1257-1262 (1977).
- (昭和53年12月25日受付)
(昭和54年4月19日採録)