

待ち行列モデルによる機能分散型分散処理方式の性能評価*

池原 悟**

Abstract

Due to the advance in hardware technology, the cost of the small scale processor is rapidly decreasing. So, it seems possible to build the computer complex with many small scale computers cheaper than the large scale computer which has the same performance.

The computer complex can be classified as a multiprocessor and a distributed system. This paper investigates the performance of the distributed system comparing with the multiprocessor which is constructed with the same number of processors as the distributed system has. The performance condition of the individual processor under which the response time of the distributed system is shorter than multiprocessor is also derived.

1. ま え が き

LSI を中心とする半導体の価格の低減による小型計算機の低価格化によって、複数プロセッサで構成される計算機（コンピュータ・コンプレックス）が単一のプロセッサで構成される計算機に比べてコスト性能比において優る可能性が生じてきた^{1),2)}。

コンピュータ・コンプレックスをプロセッサの制御方式に着目して平等マルチプロセッサ方式と機能分散型の分散処理方式（ポリプロセッサとも呼ばれる）に分けると、前者は各プロセッサで実行される機能に差はなく、各プロセッサは平等に負荷を分担する。これに対して、後者は各プロセッサごとに分担する機能が固定されるので、各プロセッサハードウェアを使用目的に合わせて設計することができて、低価格化ないし高速化が期待される。しかしその反面、平等マルチプロセッサではジョブを任意の空いているプロセッサに割り当てることができるのに対して、分散処理方式ではジョブはいくつかの専用化されたプロセッサを渡り歩いて処理されるため、次に必要なプロセッサが空かない限り処理は待たされ、応答時間が長くなる可能性がある。

本論文はネットワーク型の待ち行列解析モデル³⁾⁻⁷⁾を用いて両方式の応答時間を比較し、同等価格の分散処理方式の応答時間が平等マルチプロセッサ方式に優るためには、分散処理を構成するプロセッサの価格性能比を平等マルチプロセッサのその何倍にすれば良いか（プロセッサの性能条件）を求めた。分散処理方式をマルチプロセッサ方式と比較するに当たって、先ず任意のネットワーク型の分散処理方式とマルチプロセッサ方式の比較式を求め、これを用いてリアルタイム処理モデルを例にとった定量比較をした。次に特別な場合として、各ステージのプロセッサが1台の場合は上記プロセッサの性能条件が簡単に求まることを示し、これを用いて任意のネットワーク型分散処理方式がマルチプロセッサ方式に優る性能となるための条件を求めた。

2. 評価モデル

2.1 分散処理のモデル化

計算機ソフトウェアの機能を m 個のグループに分割し、それぞれのグループごとに専用のプロセッサを1台または複数台ずつ準備するものとする。このような特定の機能を受け持つグループをステージと呼ぶ。

システムに入力されたジョブ***は各ステージを渡り歩くことによって必要な処理を受け、ある特定のステージから去っていく。ジョブのステージ間移行は各ステージごとに設けられた待ち行列による管理を受け

* Performance Evaluation of the Distributed Function System by The Queuing Models by Satoru IKEHARA (Yokosuka Electrical Communication Laboratory, N. T. T.)

** 日本電信電話公社横須賀電気通信研究所

*** ここではジョブはシステムの抜かう処理の単位であり、ジョブ同志はお互いに独立であり、閉じている（途中で端末と応答したりしない）とする。

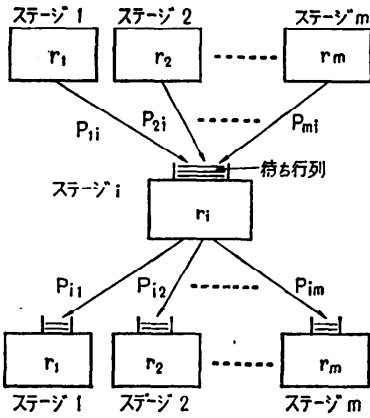


Fig. 1 The i -th stage model of the distributed system.

るものとして、分散処理を Fig. 1 のようなネットワーク型の待ち行列で近似する。このモデルの特徴は次の通りである。

- ① ジョブのステージ間移行順序は遷移確率行列に従い確率的に定まる。
- ② 各ステージのサービス率は指数分布に従う。また解析の簡単化のため、次の前提を設ける。
- ③ 各ステージの負荷がバランスするようにプロセッサが設けられている。

2.2 オープンモデルとクローズモデル

外部からのジョブの入力を考え、システムモデルとして次の2つを設けた。

(1) クローズモデル

一つのジョブがシステムから出て行くと同時に一個の新しいジョブが入力される。従ってシステム中には常に一定数のジョブがあることになる。

(2) オープンモデル

システム外からジョブはポアソン到着する。到着したジョブはすべて待つことなくシステムに入ることができる。従って一時にシステム内にあるジョブ数には制限はない。

今、ジョブはランダムに到着するが、システム内ジョブの多重度数に一定の制限があって、その値を超えてスケジュールされないシステムを考えると、このシステムの特徴は上記2つのモデルから得られる結果の中間に位置すると思われる。

3. 応答時間とそのバラツキ

同一台数の同一性能を有するプロセッサで構成され

た分散処理とマルチプロセッサの応答時間とそのバラツキを比較する。

分散処理においてはソフトウェアを分割し各ステージに割り当てることによってステージ間のパラメータ転送オーバーヘッドが生じ、その代りにソフトウェアはステージごとに特殊化することができて動的ステップ数の一部の削減が期待されるが、ここではこれらの効果を無視し、ジョブ当りのプロセッサの負荷（動的ステップ数）が分散処理とマルチプロセッサで等しいと仮定する。すると 2.1 の ③ の条件から分散処理およびマルチプロセッサのスループットの最大値 λ_{Bmax} , λ_{Amax} の間に

$$\lambda_{Bmax} = \lambda_{Amax} \tag{3.1}$$

の関係が成り立つ。

3.1 応答時間の比較

3.1.1 クローズタイプのシステムモデル

(1) 分散処理

Fig. 1 のネットワーク型待ち行列でステージ数を m , ステージ i を構成するプロセッサ数を r_i 台、その1台当りのサービス率を μ_{Bi} とし、ジョブがステージ i による処理を受けた後ステージ j の処理を受ける確率を μ_{ij} , ステージ i に存在するジョブの個数の平均値を n_i とすると、システムを構成するプロセッサの総数 M は

$$M = \sum_{i=1}^m r_i \tag{3.2}$$

であり、 μ_{ij} はすべての i について、

$$\sum_{j=1}^m \mu_{ij} = 1 \tag{3.3}$$

またジョブの多重度 N は

$$N = \sum_{i=1}^m n_i \tag{3.4}$$

で表わされる。

このシステムの平衡状態において、ステージ m に k 個のジョブが存在する確率は文献3), 4) により、以下の通り与えられる。

$$P(n_m = k) = \frac{X_m^k}{A_m(k)} \frac{g(N-k, m-1)}{G(N)} \tag{3.5}$$

但し、 X_m , $G(N)$, $g(N-k, m-1)$ はそれぞれ、

$$\mu_{Bj} X_j = \sum_{i=1}^m \mu_{Bi} X_i P_{ij} \tag{3.6}$$

$$G(N) = \sum_{\bar{n} \in (N, m)} \prod_{i=1}^m \{X_i^{n_i} / A_i(n_i)\} = g(N, m) \tag{3.7}$$

$$g(n, l) = \sum_{k=0}^n \frac{X_i^k}{A_i(k)} g(n-k, l-1) \quad (3.8)$$

$$\left. \begin{aligned} g(n, 1) &= X_i^n / A_i(n) \quad n=0, 1, \dots, N \\ g(0, l) &= 1, \quad l=1, 2, \dots, m \end{aligned} \right\} \quad (3.9)$$

$$A_i(k) = \begin{cases} k! & k \leq r_i \\ r_i! r_i^{k-r_i} & k > r_i \end{cases} \quad (3.10)$$

さて、単位時間にステージ m を通過するジョブ数を λ_m 、ステージ m の待ち行列の長さの平均値を L_{qm} 、プロセッサ使用率を ρ_m とすると、ジョブがステージ m を通過する時間 T_m はサービス時間と待ち時間の和で

$$T_m = \frac{1}{\mu_{Bm}} + \frac{L_{qm}}{\lambda_m} \quad (3.11)$$

から求める。但し、ステージ m の待ち行列のジョブ数 L_{qm} は

$$L_{qm} = \sum_{n_m=r_m+1}^N (n_m - r_m) P(n_m) \quad (3.12)$$

$$\lambda_m = \mu_{Bm} \rho_m r_m \quad (3.13)$$

$$\rho_m = \sum_{k=1}^{r_m} \frac{k}{r_m} P(n_m=k) + \sum_{k=r_m+1}^N P(n_m=k) \quad (3.14)$$

と書けるので、(3.5) 式を使えば $P(n_m=k)$ が求まるから、ステージ m の通過時間 T_m を求めることができる。一般のステージ i についてはステージの番号 i が m になるように番号をつななおして以上の手順を繰り返せばよい。

応答時間 T_B はステージ i の通過時間 T_i とジョブ当りステージ i を通る回数の平均値 e_i との積を全ステージについて加えれば良く、

$$T_B = \sum_{i=1}^m e_i T_i \quad (3.15)$$

となる。以上で各ステージのプロセッサ台数が任意の場合の応答時間が求まった。

次に各ステージのプロセッサ台数を1台と限定する場合については、プロセッサの負荷をバランスさせるという条件から上の結果はさらに簡単化できる。

ステージ i のプロセッサの使用率 ρ_i は(3.5)~(3.10) 式で $r_i \equiv 1$ において、

$$\rho_i = P(n_i \geq 1) = X_i \frac{G(N-1)}{G(N)} \quad (3.16)$$

$$G(N) = \sum_{n \in S(N, M)} \prod_{i=1}^M X_i n_i \quad (3.17)$$

となる。 N は多重度、 n_i はステージ i に平衡状態で存在するジョブの数を表わし、 X_i は(3.6)式から求まる。

(3.6) 式は平衡状態で単位時間にステージ j に到着

するジョブ数はステージ j から出て行くジョブ数に等しいことを意味している。つまり、 $\mu_i X_i$ は単位時間内にステージ i を出るジョブ数に比例する量であるから、平衡状態ではジョブ当りのステージ i の通過回数 e_i に比例し、

$$\mu_i X_i = \alpha e_i \quad (3.18)$$

とかける。 α は任意に定め得る比例定数である。

ステージ i の使用率 ρ_i はスループット λ のとき

$$\rho_i = \lambda \frac{e_i}{\mu_i} \quad (3.19)$$

である。スループットの最大値を λ_{max} とおく。前提により各ステージは負荷バランスがとれていてすべての i について ρ_i は等しいので、 $\rho_i=1$ のとき $\lambda=\lambda_{max}$ であり、(3.19) 式は $\mu_i/e_i=\lambda_{max}$ となり、これを用いて(3.18) 式は

$$X_i = \alpha / \lambda_{max} \quad (3.20)$$

と書ける。ここで比例定数 α を λ_{max} と選べばすべての i について $X_i=1$ となる。(3.17) 式から $G(N)$ は $\binom{N+M-1}{N}$ となるので、これを(3.16) 式に代入して各ステージの使用率 ρ_i は

$$\rho_i = \rho = \frac{N}{N+M-1} \quad (3.21)$$

となる。

スループット λ と応答時間 T の関係は(3.21) 式とリトルの定理 $\lambda T=N$ を用いて

$$T_B = (M-1) \frac{1}{\lambda} \frac{\rho}{1-\rho} \quad (3.22)$$

となる。

この結果はジョブのプロセッサ間の遷移確率 P_{ij} に依存しない。従って1ステージ1プロセッサで構成され、負荷バランスのとれたネットワーク型の待ち行列モデルはすべて、 $P_{i,i+1}=1, P_{ij}=0 (j \neq i+1)$ なるサイクリックな待ち行列モデルに縮退できることが分かる。

(2) マルチプロセッサ

ジョブの多重度を N とすると、 $N \leq M$ では待ちではなく、 $N > M$ のときサービス時間の $(N-M)/M$ 倍の待ち時間が生じるから、応答時間 T_A は

$$T_A = \begin{cases} 1/\mu_A & (N \leq M) \\ N/\mu_A M & (N > M) \end{cases} \quad (3.23)$$

で表わされる。

3.1.2 オープンタイプのシステムモデル

(1) 分散処理

2. で述べたオープンタイプの待ち行列モデルには

Jackson の定理⁶⁾が適用できるので、各ステージを独立した待ち行列で評価して良い。ステージ数 m 、各ステージ内のプロセッサ数がすべて r であるモデルでは、応答時間 T_B は 1 ステージの通過時間を m 倍すれば良く、

$$T_B = \frac{m}{\mu_B} \left(1 + \frac{L_{qB}}{r\rho_B} \right) \quad (3.24)$$

L_{qB} はステージ m の待ち行列にあるジョブの数であり

$$L_{qB} = \frac{(r\rho_A)^{r+1}}{(r-1)! \sum_{n=0}^r \frac{(r\rho_B)^n}{n!} \{(r-n)^2 - n\}} \quad (3.25)$$

但し、 μ_B は $\mu_B = \lambda_{max}/r$ である。

特に $r=1$ の場合は次のように簡単になる。

$$T_B = \frac{1}{\mu_B} \frac{m}{1 - \rho_B} \quad (3.26)$$

(2) マルチプロセッサ

プロセッサ数を M 台とすると応答時間 T_A は良く知られており⁶⁾、

$$T_A = \frac{1}{\mu_A} \left(1 + \frac{L_{qA}}{M\rho_A} \right) \quad (3.27)$$

$$L_{qA} = \frac{(M\rho_A)^{M+1}}{(M-1)! \sum_{n=0}^M \frac{(M\rho_A)^n}{n!} \{(M-n)^2 - n\}} \quad (3.28)$$

ρ_A はプロセッサ使用率、 L_{qA} は待ちの平均である。サービス率 μ_A は λ_{max}/M である。

3.1.3 各モデルの評価

前述の結果を用いてクローズタイプ、オープンタイプ共に任意のネットワーク型のシステムモデル M について性能を解析的に求めることができるが、すべての場合について分散処理とマルチプロセッサを比べることは出来ない以下の方法で評価用のモデルを定める。

(1) 評価モデルの選定

今 M 台のプロセッサがあるとす。これらのプロセッサをすべて並列に配置すればマルチプロセッサでありこれを $(1 \times M)$ 型システムと呼ぶ。逆にすべて直列に並べたとすればこれは $r_i \equiv 1$ なる分散処理システムであり、これを $(M \times 1)$ 型システムと呼ぶ。 M 台のプロセッサで構成するシステムのうち、待ち時間の最も短いのが $(1 \times M)$ 型システム、最も長いのが $(M \times 1)$ 型システムであるので、任意のネットワーク型の分散処理方式の応答時間はこの両者の中間に位置する。従って、上記 2 つの型のシステムを評価モデルとして用いる。

Table 1 The description of the simplified evaluation models.

項目	モデル種別 マルチプロセッサ (1×M)型	分散処理	
		(M×1)型	(√M×√M)型
モデルの説明	M台のプロセッサ全てを並列に並べる	M台のプロセッサ全てを直列に並べる	M台のプロセッサを√M台×√Mグループに分けグループ間直列、グループ内並列に並べる
ステージ数 m	1	M	√M
ステージ内のプロセッサ数 r _i	M	1	√M
ステージ間運移確率 P _{ij}		サイクリック**	サイクリック
サービス率*** μ _A , μ _B	μ _A	μ _B =μ _A	μ _B =μ _A /√M

* モデル記号として $(m \times r)$ を用いる。 m はステージ数、 r はステージ内プロセッサ数を示す。

** $j=i+1$ のとき $\mu_{ij}=1$ 、 $j \neq i+1$ のとき $\mu_{ij}=0$ のモデル

*** サフィックス A, B はそれぞれマルチプロセッサ、分散処理を示すために用いる。

次に上記 2 モデルの中間的なモデルとして \sqrt{M} 台ずつのプロセッサを \sqrt{M} ステージ直列に並べたシステム、 $(\sqrt{M} \times \sqrt{M})$ 型システムを評価する。

以上のモデルを Table 1 に比較して示す。

これらのモデルのうち、 $r_i=1$ となる $(M \times 1)$ 型は 3.1.2 の結果から任意のネットワーク型のモデルを表わしているが、 $r_i \neq 1$ の場合は一般にステージごとのプロセッサ台数はまちまちであり、必ずしも $(\sqrt{M} \times \sqrt{M})$ 型システムで代表できないと思われるので、ほかの 1 つの例として付録に示すネットワーク型のリアルタイム処理モデルを用いて評価する。これはバンキング処理におけるトランザクション処理をモデル化した例である。

(2) 評価結果

Table 1 の各モデルの応答時間は使用するプロセッサ数 M を決めれば 3.1.1, 3.1.2 の結果を用いて求めることができる。今 $M=9$ 台の場合の応答時間を Fig. 2 (次頁参照) に示す。 Fig. 2 は斜線を付した部分でクローズ/オープンの評価モデルの差を表わし、曲線の種類でプロセッサ配置方法の違いを表わしている。

M を変えて同種の図を得ることができる。それらの図から以下の特性が確認できる。

- i) 同一数、同一性能のプロセッサの組合せで作られた分散処理は待ちが多く応答時間が長い。
- ii) オープン/クローズドの近似するモデルのタイ

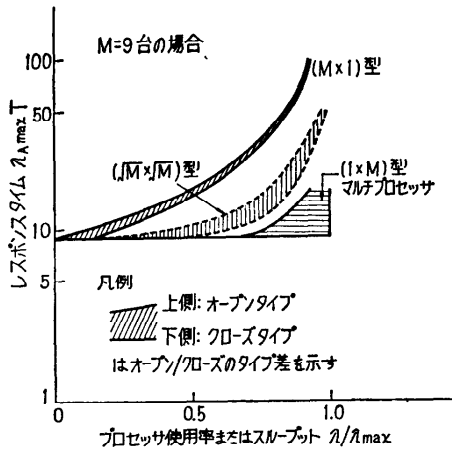


Fig. 2 The comparison of the response time characteristics.

の差は直列型 ($M \times 1$) 型では $\rho \rightarrow$ 大となるにつれて小さくなるが、 $(\sqrt{M} \times \sqrt{M})$ 型は $\rho = 0.6 \sim 0.8$ で差が最も大きくなる (30~40%) ので注意を要する。

なお付録 1 のモデルの解析結果は Fig. 2 の $(\sqrt{M} \times \sqrt{M})$ 型に近い。

3.2 応答時間のバラツキ

ポアソン到着、指数サーバのネットワークモデルの応答時間は指数分布になる。ここでは応答時間のバラツキを応答時間の標準偏差で評価する。

各ステージでの待ち時間とサービス時間が独立で、かつステージ間のサービス時間もお互に独立であるとすると、応答時間の分散 σ_T^2 は各ステージの通過時間の分散 $\sigma_{T_i}^2$ の和であり、各ステージの通過時間の分散は待ち時間の分散 σ_w^2 とサービス時間の分散 σ_s^2 の和である。 σ_w^2 と σ_s^2 は良く知られており⁷⁾、 σ_T^2 が求まる*。

例として 3.1 と同じく $M=9$ の場合について計算すれば Fig. 3 を得る。Fig. 3 の結果から以下の結論を得る。

i) 分散処理では応答時間の分散が小さくなる。

これは 1 つのジョブがいくつもの互に独立なステージで分割処理され、処理の遅れと進みが平均化されるためであり、応答時間のバラツキはほぼステージ数の平方根 \sqrt{m} に逆比例して小さくなる。

ii) $(\sqrt{M} \times \sqrt{M})$ 型、 $(M \times 1)$ 型の分散処理を比

* 特に $r=1$ のときの分散処理では $\sigma_T^2 = m / (\mu_B^2 (1 - \rho_B))$ となる。

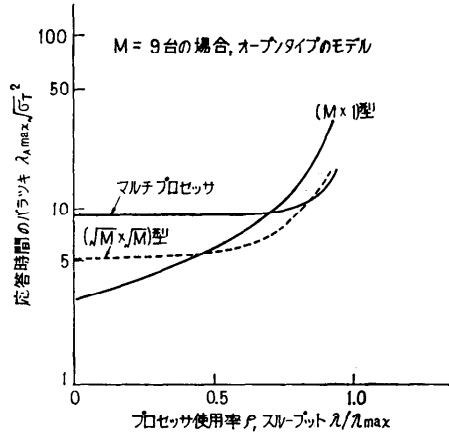


Fig. 3 Standard response time deviation of the evaluation models.

べると ρ が小のとき ($M \times 1$) 型が有利、 ρ が大のとき $(\sqrt{M} \times \sqrt{M})$ 型が有利である。

これは $(M \times 1)$ 型分散処理は応答時間のバラツキが使用率に敏感であることを意味する。

実際のシステムではジョブの処理時間はステージ間で何らかの相関を持つと考えられるので i) の傾向は薄まると思われる。

4. 分散処理がマルチプロセッサに優る性能となるための条件

同一台数のプロセッサを用いて構成された分散処理とマルチプロセッサを比較し、分散処理の性能 (応答時間又はそのバラツキ) がマルチプロセッサよりも良くなるための分散処理を構成するプロセッサの性能に関する条件を以下の方法で求める。

分散処理、マルチプロセッサ共にプロセッサ台数は等しく M とする。但しここでは分散処理の各プロセッサの性能はマルチプロセッサの性能の γ 倍であるとする。すると分散処理、マルチプロセッサのスループットの最大値 λ_{Bmax} 、 λ_{Amax} の関係は (3.1) 式と異なり、

$$\lambda_{Bmax} = \gamma \lambda_{Amax} \tag{4.1}$$

となる。

以下 3 つの場合に分けて性能条件 γ_0 を求める。

4.1 応答時間が全てのスループット領域で短くなる条件

全てのスループット λ に対して分散処理の応答時間 T_B がマルチプロセッサの応答時間 T_A より常に小さくなるための性能条件 γ_0 を求める。

(1) 一般の場合

任意のネットワーク型の分散処理モデルについて3.の方法で応答時間を求めることができる。クローズタイプでは(3.15)(3.23)式から、オープンタイプでは(3.24)(3.27)式から T_B, T_A が求まる。(4.1)式の条件から分散処理の各プロセッサのサービス率を γ 倍するとし、 γ を変えて T_B を求め、 $\lambda \leq \lambda_{Amax}$ なる全てのスループットに対して $T_B \leq T_A$ なる γ_0 を求める。 $(\sqrt{M} \times \sqrt{M})$ 型モデル及び付録のリアルタイム処理モデルについてはこの方法によって γ_0 を求める。

(2) $r_i \equiv 1$ の場合 ($(M \times 1)$ 型)

この場合の γ_0 は以下の通り容易に求まる。

まずクローズタイプのモデルは(3.22)式で $\rho = \lambda/\mu_B, \lambda_{Bmax} = \gamma \lambda_{Amax}$ とおいて T_B を求め、これと(3.23)式の T_A との間で $T_B \leq T_A$ なる条件を求めると、 γ_0 は

$$\gamma_0 = 2 - \frac{1}{M} \tag{4.2}$$

次にオープンタイプのモデル*では(3.26)(3.27)式を用い、同様にして γ_0 は

$$\gamma_0 = \frac{1}{\left(1 + \frac{\mu_A L_{QA}}{M \lambda}\right) + \frac{1}{\mu_A}} \tag{4.3}$$

以上で得られた γ_0 を Fig. 4 に示す。 γ_0 の最大値と最小値を Table 2 に示す。

4.2 応答時間がほぼ同等となる条件

前の結果はすべてのスループット領域で分散処理の応答時間がマルチプロセッサより短くなる条件で求めたものであり、かなり厳しい条件である。ここでは

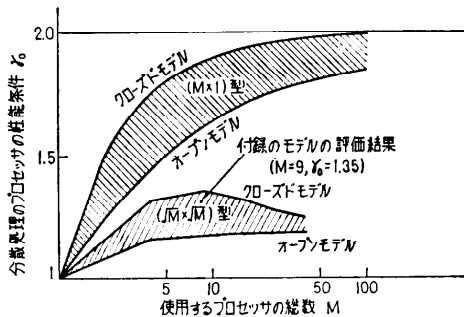


Fig. 4 Performance condition of individual processors under which the response time of the distributed system is shorter than the multiprocessors.

* オープンタイプの $(M \times 1)$ 型については文献8)で評価されている。

Table 2 Min/Max of performance condition.

モデル	オープンタイプ		クローズタイプ	
	最大 γ_0	最小 γ_0	最大 γ_0	最小 γ_0
$(M \times 1)$ 型	2.0 ($M \rightarrow \infty$)	1.25 ($M=2$)	2.0 ($M \rightarrow \infty$)	1.5 ($M=2$)
$(\sqrt{M} \times \sqrt{M})$ 型	約 1.2 ($M=4 \sim ?$)	1 (推定) ($M \rightarrow \infty$)	約 1.36 ($M=4 \sim 16$)	1 (推定) ($M \rightarrow \infty$)

T_B, T_A がほぼ同等となる条件として

$$\int_0^{\lambda_{Amax}} (T_A - T_B) = 0 \tag{4.4}$$

を採用し、この条件の成立するときの値 γ_0 を求める。

(1) 一般の場合

4.1の(1)と同様にして γ を変えて T_B を求めて得られた λ と T_B の関係を図に描き、 T_A の曲線との間で(4.5)式の成立つ γ_0 を読み取ることによって γ_0 を求めることができる。

(2) $r_i \equiv 1$ の場合 ($(M \times 1)$ 型)

$r_i \equiv 1$ なるクローズタイプのモデルでは γ_0 は以下の通り簡単に求まる。即ち、(3.22)式、(3.23)式を(4.4)式に代入して、 $\rho_B = \lambda_B/\lambda_{Bmax}, \lambda_{Bmax} = \gamma \lambda_{Amax}$ とし、 $m=M$ とおいて γ_0 は、

$$\gamma_0 = \frac{1}{1 - e^{-M/(M-1)}} \tag{4.5}$$

γ_0 は M に対して単調に増大し、

$$\lim_{M \rightarrow \infty} \gamma_0 = \frac{e}{e-1} \approx 1.582 \tag{4.6}$$

となる。

Fig. 5 に(4.5)式の値を示す。

4.3 応答時間のバラツキが小さくなる条件

応答時間のバラツキ $\sqrt{\sigma_{TA}^2}, \sqrt{\sigma_{TB}^2}$ について4.1と同様の比較を行い、

$$\sqrt{\sigma_{TA}^2} \geq \sqrt{\sigma_{TB}^2} \tag{4.7}$$

なる条件を満たす γ_0 を求める。

オープンタイプのモデルの $\sigma_{TA}^2, \sigma_{TB}^2$ は3.2の

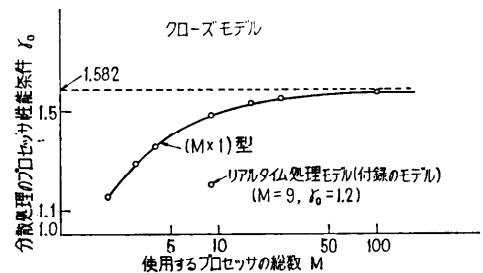


Fig. 5 Performance condition of individual processors under which the response time of the distributed system is equal to the multiprocessor.

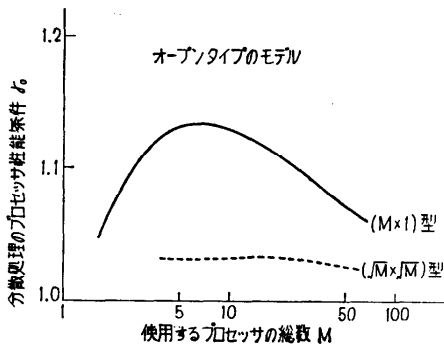


Fig. 6 Performance condition of individual processors under which the standard response time deviation of distributed system is small than the multiprocessor.

方法で求められる。 μ_B を γ 倍して σ_{TB}^2 を求め、これを σ_{TA}^2 と図的に比較して(4.7)式の満たされる γ_0 を求めるとFig. 6を得る。

4.4 まとめ

4.1の条件はかなり厳しい条件であり、実効上4.2の条件で比較してもさしつかえない場合が多いと思われる。以上4.1~4.3の結果をまとめると以下のとおりである。

- i) $(M \times 1)$ 型の分散処理の場合が最も厳しい条件となり、 γ_0 の値は
 - 4.1の条件では1~2の範囲
 - 4.2の条件では $1 \sim e/(e-1) (\approx 1.582)$ の範囲となる。
- ii) これに対して各ステージのプロセッサ数を複数化し、ステージ数を減少させれば γ_0 をかなり小さくできる。 $(\sqrt{M} \times \sqrt{M})$ 型で4.1の条件では γ_0 は高々1.3~1.4である。

iii) 応答時間に関する性能条件4.1, 4.2のいずれかが成り立っているときは、応答時間のバラツキに関する条件4.3は満足されている。

5. あとがき

分散処理とマルチプロセッサの応答時間、応答時間のバラツキ、負荷変動の影響等を待ち行列モデルを用いて比較評価し、分散処理がマルチプロセッサと同等又はそれ以上の性能となる条件を求めた。その結果、分散処理ではプロセッサの性能を最悪でもマルチプロセッサの2倍($\gamma_0=2$)にすれば、分散処理はあらゆるスループット領域で応答性に優り、約1.58倍にすれば同等になることが分かった。この条件は各ステージ

のプロセッサを複数化すれば緩和され、ステージ数とステージ当りのプロセッサ数の等しいモデルでは $\gamma_0=1.3 \sim 1.4$ で分散処理が優る性能となる。このように分散処理のプロセッサ性能を上げれば、応答時間のバラツキは十分小さくできる。

なお分散処理方式のメモリアバヘッド、ソフトウェア、ハードウェア資源競合の評価については別途報告する予定である。

おわりに本研究を通じて御討論戴いた本研究所村岡調査員に深謝する。

参考文献

- 1) E.C. Joseph: Future Computer Architecture -POLYSYSTEMS, COMPCON '72, pp. 149~153.
- 2) R. Turn: Computers In the 1980s, Columbia U.P. (1974)
- 3) W.J. Gordon, G.F. Newell: Closed Queueing Systems with Exponential Servers, Oper. Res. 15, 2, '67, pp. 254~265.
- 4) J. P. Buzen: Computational Algorithms for Closed Queueing Networks with Exponential Servers, CACM, Vol. 16, No. 9, pp. 527~531.
- 5) R.W. Conway, et al.: Theory of Scheduling, Addison-Wesley (1967)
- 6) T.L. Saaty: Résumé of Useful Formulas in Queueing Theory, Oper. Res. Vol. 5, pp. 161~200 (1957)
- 7) 雁部: 電話交換トラヒック, 電気通信協会, S 41.
- 8) 牧野, 箱崎: 分散機能型多重プロセッサシステムの構成と制御, 情報学会, アーキテクチャ研究会資料 19, '76.

〈付録1〉 リアルタイム業務処理モデル

Fig. A-1, Fig. A-2に示す処理モデルをTable A-1に示したパラメータ値を設定して動作させるものとする。Fig. A-3にジョブの多重度と周辺系の通過時間を除く応答時間を示す。

(Fig. A-1, A-2, A-3. Table A-1は次頁参照)

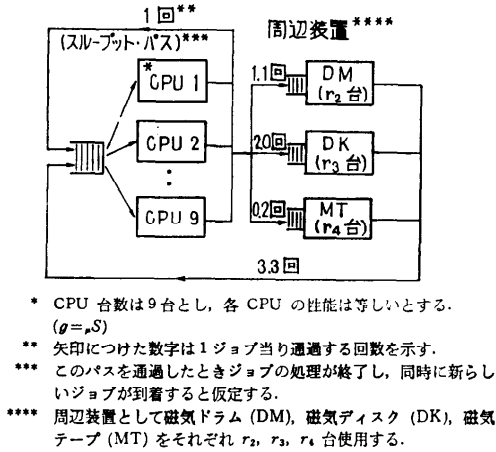


Fig. A-1 Multiprocessor model for real time job.

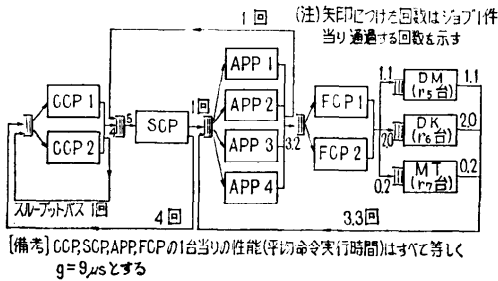


Fig. A-2 Distributed system model for real time job.

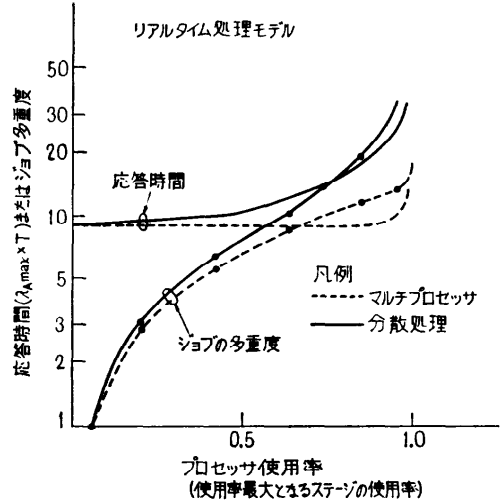


Fig. A-3 Response time and number of job of the real time model.

Table A-1 Model description parameter of the real time mode.

ステージ番号	ステージ名	動的ステップ数 (%)	ステージ通過回数	サービス率 (件/秒)	装置台数
①	CPU	100%	4.3回	23	9
1	CCP	20%	5	133	2
2	SCP	12%	5	236	1
3	APP	46%	4.3	50	4
4	FCP	22%	3.3	80	2
⑤	DM	—	1.1	100	(1)*
⑥	DK	—	2.0	26	(4)*
⑦	MT	—	0.2	56	(1)*
	1~4の合計	100%	—	—	

CCP: Communication Control Processor

SCP: Scheduling Processor

APP: Application Processor

FCP: File Control Processor

○: マルチプロセッサのステージを示す。

* プロセッサ系だけの評価では台数とする。

(昭和51年11月26日受付)

(昭和52年3月1日再受付)