# 分散処理システムにおける資源間結合方式通信方式の性能評価

飯作俊一　小野欽司
（国際電信電話株式会社　研究所）

## 1. まえがき

最近のＬＳＩ技術の進歩により、グロシュの法則から逸脱するハードウエア（メモリ・プロセッサ）の低コスト化が可能となり、それにともない分散処理が盛んに議論されている。[1]

分散処理には、地理的に各機能を分散して処理する形態（ネットワーク分散）や、ホストプロセッサ機能をいくつかに分割し、それぞれに専用化された多数の小型プロセッサを割当て密に結合したマルチプロセッサ形態など、利用形態、応用分野、システムの構成法に応じて種々の形態が考えられ、分散処理共通の問題点のみならず、システムに依存した個有の問題点も多い。

高度な分散処理システムを構築するためには、負荷分散・機能分散をより効率よく実現し、システムのコストパフォーマンスの向上と、高信頼性・高拡張性等のシステムの柔軟性への要求を満たす必要がある。マルチプロセッサ構成からなる機能分散システムの場合、方式上の基本的な問題点として、機能配分・負荷配分などを考慮した分散ＯＳの構成法と効率化、機能分割にともなうプロセッサ分割度、プロセッサ間通信オーバヘッド、共通資源へのアクセス競合、負荷の不均衡による性能劣化、障害対策などがある。

一方、最近急速に普及が進んできた分散処理プロセッサは、マルチプロセッサ構成からなり、機能としてＣＣＩＴＴのＸ.25，Ｘ.21のサポートが可能であるとともに、ローカルに使用できる言語や、ネットワーク管理ソフトの強化も行なわれている。[3] このような分散処理プロセッサを各地に配置したネットワーク分散処理の活発化にともない、ネットワークアーキテクチャ、ネットワークＯＳの構成法が重要な問題点となってきている。

本報告は、従来からマルチプロセッサ構成の分散処理方式の基本的な問題点である、プロセッサ間通信によるオーバヘッド、共通資源へのアクセス競合に焦点をあてて議論を行なう。これらの問題点は資源間の結合方式[2]にかなり依存するとともに、資源間で通信する情報の性質にも依存する問題である。そこで、プロセッサ間通信をプロセッサ相互間で行なう場合と、メモリ装置を介して行なう場合の二つの方式に分類し、結合方式を構成の簡単なバス方式で実現した場合の各種方式を比較・評価を行なう。

第２章では、分散処理方式をいくつかの角度から分類をおこない、さらに機能分散処理方式に焦点をしぼって、方式上の特徴、問題点、並びに解決法について述べる。

第３章では、ハードウェア資源間結合方式の各種方式のうちバス結合方式を取り上げ、プロセッサ間通信を通信情報の性質によって、プロセッサ間結合、プロセッサ・メモリ結合によって実現する方式について考察し、稼動プロセッサ数、送信待ち合せ時間の検討を行なう。

第４章は、前章で考察した方式のうち、プロセッサ間結合方式での競合調停方式として、先着順方式、ポーリング方式の待ち合せ時間、並びにプロセッサ・メモリ結合を複数バスによって実現した場合の、メモリアクセス競合、

稼動プロセッサ数の様子を示す数値例を与え、合せて考察を行なう。

## 2. 分散処理

### 2.1 分散処理システム

分散処理には、大きく通信対象の配置によってネットワーク分散、センタ内分散に分類できる。さらに、資源等の管理上の主従関係のある/なしによって、水平分散・垂直分散、さらに分散の対象によって機能分散・負荷分散と、いろいろな角度から分類することができる。(表1)。しかし、一般的にはプロセッサを複数台結合し各プロセッサに負荷を平等に分散させる方式(負荷分散)、機能をいくつかに分割しそれぞれに専用的にいくつかのプロセッサに割付ける方式(機能分散)、地理的に分散設置されたプロセッサを通信回線で結合したネットワーク分散処理に分類されよう。

ネットワーク分散処理システムは近年のデータ通信システムの普及にともない急速に発達してきたシステムであり、ネットワークアーキテクチャ(プロトコル)と、効率のよいネットワークの構成(資源の最適配置・配分・運用・統制)を実現することが、分散データベース、分散処理用OSの実現手法と合せて主たる問題点である。

| | |
|---|---|
| ・配置 | ネットワーク分散 |
| | 構内分散 |
| ・主従関係 | 垂直分散 |
| | 水平分散 |
| ・対象 | 機能分散 |
| | 負荷分散 |

表1　分散処理の分類

### 2.2 機能分散処理

多数の機能的に専用化したプロセッサを結合してジョブを処理する機能分散処理の方式上の特徴としては次の項目があげられる。

(1) システム構成の柔軟性・高信頼性
(2) プロセッサ専用化と並列処理による高性能化
(3) 機能分割による個々のOSの小形化と生産性向上

一方、方式上の解決すべき問題点としては、

(1) 機能分割によるプロセッサ分割損
(2) 特定プロセッサへの負荷集中によるシステムのデッドロック
(3) プロセッサ間通信オーバヘッド
(4) 共通資源へのアクセス競合による性能低下
(5) プロセッサの専用化方式
(6) フェイルソフト・フォールトトレラントなシステム構成法

である。上記項目は相互に関連をもっており、システムの利用形態等を考慮しつつ対策を講じる必要がある。

(1)については、機能の分割と各プロセッサへの機能の配分の仕方であり、共通機能と専用機能の切分けが重要である。共通機能の各プロセッサへの付与はメモリ増分と交信オーバヘッドのトレードオフにより決まる。専用機能(処理プログラム実行部、実メモリ管理部、負荷制御部、構成制御部、外部通信制御部、保守診断部など)には、規模に応じて専用のプロセッサを付与し、マイクロプログラム化を行なうとともに、OS機能の一部分はファームウェア化することが望ましい。

(2)については、特定機能と負荷分散マルチプロセッサ化する方式と、ダイナミックなプロセッサ割付け法による

方式が考えられる。

(3),(4)については、資源の結合方式と関連して、プロセッサ間通信の情報の性質にあった結合形態・通信方式を考える必要がある。

(5)はすべての基本となる項目である。プロセッサを均質形(同一)にするか非均質形にするかは、システム全体のコストパフォーマンス、制御の容易性、柔軟性、信頼性を考慮する必要があり一般的には、特定システムでより性能重視のシステムでは非均質構成、負荷特性が不明確な汎用システム、信頼性を重視したシステムでは均質構成と言えよう。項目(1)との関連から、プロセッサ専用化の一手法としてOSのファームウエア化が有効であり、さらに実現する機能対応にハードウェア、ソフトウエア、およびファームウェアでの機能分担を最適にする必要がある。

(6)については(2),(5)とも関連があり、一般的には性能的に縮退した状態でシステムの稼動を続行できるとともに、障害検出・回復のスピードアップをはかるため、保守診断プロセッサを用いたプロセッサ動的変更(再配置)による代行が可能な方式が望ましい。

以上のような分散処理システムの問題点、解決法において必要と思われるハードウェア構成技術としては、

(1) マイクロプログラム制御方式
(2) 主メモリ・制御メモリのマッピング機構
(3) 資源間結合装置
(4) 資源間制御装置

である。

マイクロプログラムは同種のプロセッサの専用化を容易にする方式であり、マイクロプログラム格納の制御メモリと主記憶メモリとのアドレス変換機構を用いて、プロセッサを動的に機能の変更を可能とする。これによりプロセッサの再配置制御が容易となり、負荷の平滑化、信頼性がはかられるとともに、システムの高性能化をもたらす。

資源間結合装置は、各プロセッサ間が通信する情報の性質にかなり依存する。コマンドタイプの場合には、一回の情報量は少ないが通信要求頻度は大きい。一方、共有データタイプの情報の場合には、一回の情報量はかなり多いが、通信要求頻度はそれほど多くない。したがって、通信の形態に応じた結合方式を考慮する必要があり、直接結合、共有メモリを介した結合等の選択を評価する必要がある。

資源間制御装置は、結合装置の使用・調停等を制御する装置であり、バス等の高速な結合装置に対して有効である。したがって、これらの各種制御方式について性能評価する必要がある。なお、低速な資源間に対してはソフトウエア上の制御(テストアンドセット、セマフォ機構)が主体となろう。

## 3. 資源間結合方式

### 3.1 結合方式と通信方式

システムに存在する各種資源をどのような形態で結合するかは、プロセッサ通信方式、資源間情報転送効率、資源の増設の容易性、システムの信頼性の確保などシステム構成上の観点から非常に重要である。表2に従来から提案されている各種結合方式を示す。[2]

- 結合方式
  - ネットワーク型
    - 階層構造(ツリー, スター)
    - 非階層構造(リング, ループ)
  - 共有メモリ型
    - バス方式(単一バス, 複数バス, パススイッチ)
    - メモリスイッチ方式(クロスバースイッチ, スイッチ網)
    - マルチポートメモリ

表 2　プロセッサ結合方式の分類

本章では、アクティブな資源としてプロセッサ、パッシブな資源としてメモリを取り上げ、結合方式としてバス方式に対象をしぼり、機能分散形マルチプロセッサ構成を主体に考察する。

結合形態を大きく密結合と疎結合に分類する。ここで言う結合が密とは各プロセッサ間の情報転送量が多いことを意味する。すると前者については、主記憶共有形のマルチプロセッサ、後者については独立に動作するプロセッサをバス結合で接続するマルチプロセッサが考えられる。同一のシステム内に両者が混合される場合もありうる。上記二方式をそれぞれプロセッサ・メモリ間結合方式、プロセッサ間結合方式と呼び、バス結合の場合の種々の調停方式、並びにバス・メモリアクセス競合にともなう性能低下について定性的・定量的な検討を行なう。

## 3.2 プロセッサ・メモリ結合方式

主記憶共有形のマルチプロセッサシステム（密結合マルチプロセッサ）の結合方式にバス結合を用い、プロセッサ、プライベートメモリ、共有メモリ、バスをそれぞれ複数用いた一般的なプロセッサ・メモリ間結合方式を図1に示す。[4][5] プロセッサ数を $p$、共有メモリ数を $m$、バス本数を $b$ とすると、$m = b$ の場合にはメモリに対してバスは専用的に使用され、バス上の競合とメモリへの競合は同一のものと考えられる。一般的にバスの有効利用の観点から、$p \geq m \geq b$ の形態である。この場合、各プロセッサの状態は次の3つの場合に分類できる。

(1) プライベートメモリアクセス状態
(2) 共有メモリアクセス状態
(3) 共有メモリアクセス待ち状態

機能分散形のマルチプロセッサ構成においては、状態(1)は内部処理であり、実際稼動している状態である。状態(2)は共有メモリを介した通信であり、一種のオーバヘッドとみなすことができる。状態(3)には、アクセスしたいメモリが他のプロセッサによって占有されていないにもかかわらず、バス開そくによる待ち状態と、バスが空きにもかかわらずメモリ開そくによる待ち状態の二通り考えられる。上記のモデルで一般的には、状態(1),(2)にあるプロセッサの平均を求めることにより、複数バスマルチプロセッサ方式のスループットが求まる。

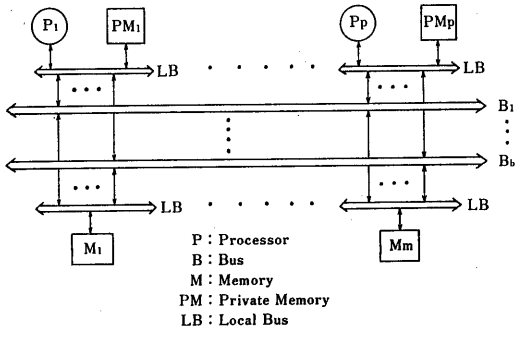

図1 プロセッサ・メモリ間結合方式（複数バス方式）

図1の複数バス方式の性能は、プロセッサの状態遷移を状態(1)にいるプロセッサ数と、状態(2),(3)にいるプロセッサ数の2状態からなる出生死滅過程に縮退した形で近似的に求められており[5]、状態(1)にいるプロセッサ数の平均は以下の式で与えられる。

$$P = \sum_{i=1}^{p} \frac{\rho^{p-i} \frac{p!}{(i-1)!} \prod_{k=1}^{p-i} \beta_k^{-1}}{1 + \sum_{j=0}^{p-1} \left[ \rho^{p-j} \frac{p!}{j!} \prod_{k=1}^{p-j} \beta_k^{-1} \right]} \quad (1)$$

$$\rho = \frac{\lambda}{\mu}$$

ここで、$\lambda$ は各プロセッサの共有メモリへの平均アクセス率（ポアソン分布）、$1/\mu$ は共有メモリへの平均アクセス時間（指数分布）、プロセッサ並

びにメモリを同一とした結果である。したがってm台の共有メモリへの各プロセッサの平均アクセス率は$\lambda/m$である。$\beta_k$の導出については文献(5)参照。式(1)を利用すると、共有メモリアクセス待ちプロセッサ数$N_g$は、

$$N_g = k - (1+\rho)P \qquad (2)$$

共有メモリアクセス中の平均プロセッサ数$N_a$は

$$N_a = \rho P \qquad (3)$$

で求められる。

式(1)～(3)からプロセッサと共有メモリモジュールの台数、性能とバス結合方式の関係が得られるとともに、単一バス方式の限界点についても、共有メモリへのアクセス頻度に応じて考察することができる。また、機能分散マルチプロセッサ構成の場合は、式(1)のプロセッサ数が本来の意味で稼動プロセッサ数とみなすことができよう。

## 3.3 プロセッサ間結合方式

図2に示すような単一バス結合方式の場合のプロセッサ間結合方式について検討する。同図において通信時には送信側、受信側のプロセッサが共に保留される形をとり、厳密には受信側のプロセッサの状態を考慮する必要があるが本節では送信側からの通信要求を受信側は拒否することはないと仮定する。図の結合形態において問題となるのは、バス使用権の決定方式である。決定方式には、先着順、優先順位による方式、ラウンドロビン方式、ポーリング方式などが考えられる。図3に先着順処理方式、図4にポーリング処理方式のモデルを示す。ここでの評価として送信要求が発生してから送信開始となるまでの時間である待ち合せ時間について検討を行なう。

各プロセッサからの通信要求は平均$\lambda_i$ ($i=1,\cdots N$)のポアソン分布に従って生起する。したがって、全通信要求呼の生起率は$\Sigma\lambda_i$である。通信転送時間の平均を$h$、2次モーメントを$h^{(2)}$とする。ポーリング方式の場合、ポーリングメッセージ通信時間の平均を$\mu$、2次モーメントを$\mu^{(2)}$とする。以上の仮定で各プロセッサの送信待ち合せ時間の平均を求める。

先着順の場合、M/G/1[6]で近似することができ

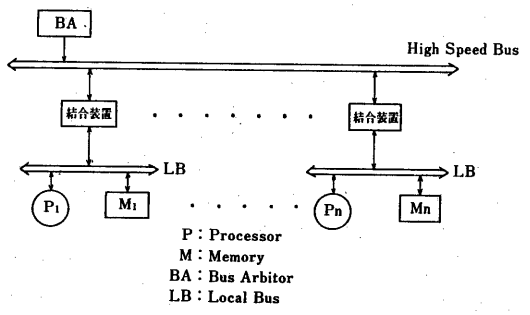


図2 プロセッサ間結合方式（単一バス方式）

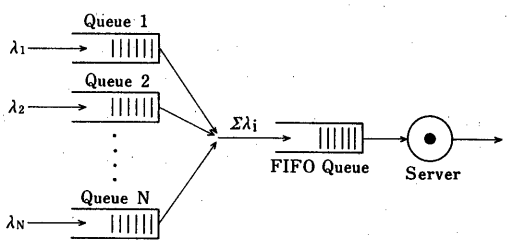


図3 先着順処理方式のモデル

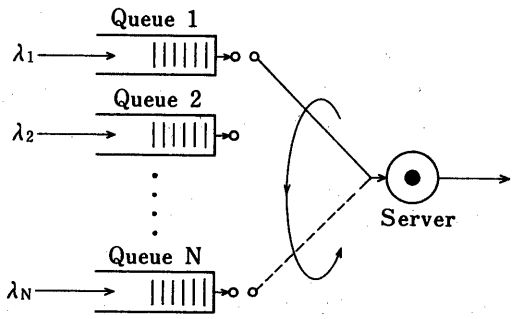


図4 ポーリング方式のモデル

$$W_R = \frac{\sum_{i=1}^{N} \lambda_i h^{(2)}}{2\left(1-\sum_{i=1}^{N}\lambda_i h\right)} \qquad (4)$$

優先順位のとき、クラスrの通信要求呼の待ち合せ時間は[6]、割込み継続形の場合においては、

$$W_{pr} = \frac{\sigma_r^{(2)}}{2(1-\sigma_{r-1})(1-\sigma_r)} \qquad (5)$$

非割込み形の場合においては、

$$W_{npr} = \frac{\sigma_p^{(2)}}{2(1-\sigma_{r-1})(1-\sigma_r)} \qquad (6)$$

である。ただし、pは優先クラス数であり、優先権番号r(=1,…p)が小さいほどプロセッサの優先順位が高いとする。また式(5),(6)の$\sigma_r$, $\sigma_r^{(2)}$はそれぞれ、

$$\left.\begin{aligned}\sigma_r &= \sum_{i=1}^{r}\lambda_i h \\ \sigma_r^{(2)} &= \sum_{i=1}^{r}\lambda_i h^{(2)}\end{aligned}\right\} \qquad (7)$$

である。

ポーリング方式の場合、ポーリング周期毎での通信要求呼の処理の仕方によって種々の方式が考えられる。ポーリングメッセージ到着時点で待ち行列に並んでいた通信要求呼をすべて処理するゲート式での待ち合せ時間は[7]

$$W_g = \left[\frac{N\xi h^{(2)}}{1-N\rho} + \mu^{(2)} + \left\{\frac{N(1+\rho)}{1-N\rho}-1\right\}\mu^2\right] \cdot \Big/ 2\mu \qquad (8)$$

で与えられる。

待ち行列内のすべての通信要求呼を処理する全処理式の場合には[7]

$$W_e = \frac{\mu^{(2)}}{2\mu} + \frac{(N-1)\mu + N\lambda h^{(2)}}{2(1-N\rho)} \qquad (9)$$

待ち行列の先頭の通信要求呼だけを処理する制限式での待ち合せ時間は[8]、

$$W_l = \frac{m-(1-a)}{\lambda(1-a)}$$

$$m = \frac{1}{2\{1-N(\rho+\xi)\}}\Big[N(N-1)(\rho+\xi)^2 + N(\rho^{(2)}+2\rho\xi+\xi^{(2)}) - \frac{N(N-1)\rho\xi a}{1-N(\rho+\xi)} + a\{-N\rho^{(2)}+2\xi(1-N\rho)-2(N-1)\rho + 2(N-1)(\rho+\xi)-N(N-1)\rho(\rho+\xi)\}\Big]$$

$$a = \frac{1-N(\rho+\xi)}{1-N\rho} \qquad (10)$$

ただし、$\rho=\lambda h$、$\rho^{(2)}=\lambda^2 h^{(2)}$, $\xi=\lambda\mu$, $\xi^{(2)}=\lambda^2\mu^{(2)}$ である。

一つのプロセッサに注目したとき、ポーリングメッセージの受信間隔であるサイクルタイムは重要なファクターである。サイクルタイムの平均はすべての方式で等しく

$$C = \frac{N\mu}{1-N\rho} \qquad (11)$$

で与えられる。

全処理式・ゲート式での各プロセッサからの通信要求呼の最大スループットは、

$$\lambda_{max} = \frac{1}{Nh} \qquad (12)$$

制限式の場合は、

$$\lambda_{max} = \frac{1}{N(h+\mu)} \qquad (13)$$

である。

## 4. 各モデルの数値例と考察

図5に、各種ポーリング方式の場合の待ち合せ時間の計算結果を示す。ポーリングメッセージ通信時間、通信転送時間の平均はそれぞれ1, 10の一定分布で計算を行なった。図5はプロセッサ台数(N)、プロセッサの通信要求頻度(λ)、をパラメータとした場合であり、全処理式・ゲート式の最大スループットを破線で、制限式の最大ス

ループットを一点鎖線で示した。この値より小さい通信要求であればシステムに定常状態が存在する。

図6はポーリング方式のサイクルタイムを示したものであり、条件は図5と同一である。

図7は先着順処理方式の場合の待ち合せ時間を、通信転送時間をパラメータとして計算した例である。横軸はすべてのプロセッサからの通信要求率であることに注意を要する。

図8，9は複数バス結合のときのプロセッサ・メモリ結合方式の性能評価の数値例である。pはプロセッサ数，mは共有メモリ数，bはバス本数であり、横軸はプロセッサの共有メモリへの平均アクセス率と平均アクセス時間の積である。縦軸は、プライベートメモリをアクセスしながら内部処理を実行中の平均プロセッサ数であり、これを稼動プロセッサ数としている。図8は、バス本数が2本のときであり、共有メモリへのアクセス頻度が大きくなるにつれて、アクセス競合が発生する

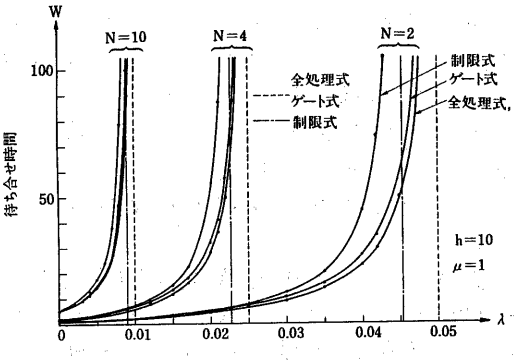

図5 ポーリング方式の待合せ時間

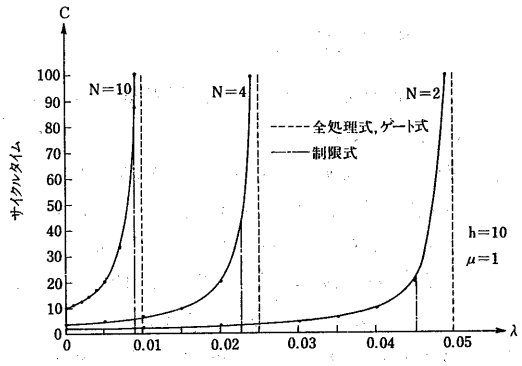

図6 ポーリング方式のサイクルタイム

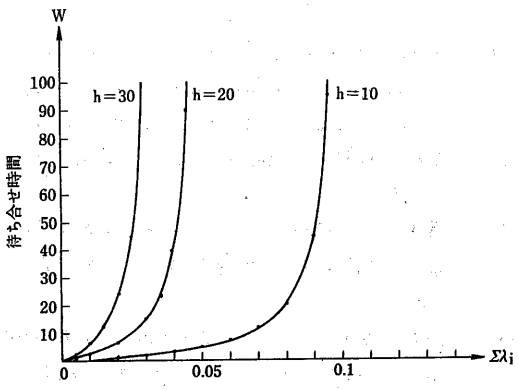

図7 先着順処理の待ち合せ時間

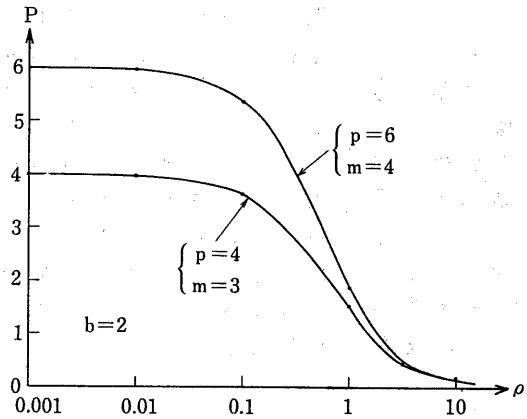

図8 平均稼動プロセッサ数

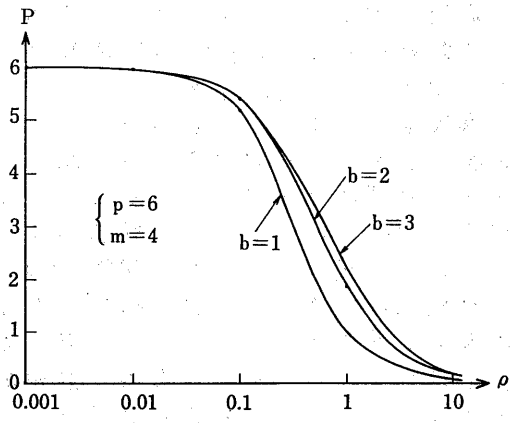

図9 平均稼動プロセッサ数

確率が高くなり、内部処理実行中のプロセッサ数が減少していく様子がわかる。

図9は、プロセッサ数とメモリ数をそれぞれ6，4と固定し、バス本数をパラメータとした場合の稼動プロセッサ数を示したものである。バス本数がメモリ数に近づくにつれてそれほど効率が上がらなくなる点には注意が必要である。

図8，9においては、プライベートメモリをアクセスしつつ内部処理を実行している平均プロセッサ数を稼動プロセッサ数と呼んだが、この意味は機能分散形マルチプロセッサ構成の場合、内部処理が本来のプロセッサの処理であり、共有メモリへのアクセスは他プロセッサとの通信を目的とした場合が多く、いわゆる通信オーバヘッドに相当すると考えられるからである。別の観点として、共有メモリアクセスによる処理をオーバヘッドとみなさない場合には、平均稼動プロセッサ数は、$P(1+p)$で与えられる。上記のようにシステムの構成法によっては評価の判断基準が、かなり変なってくる点には十分注意を要する。

## 5. むすび

本報告では、マルチプロセッサ構成の分散処理システムにおける問題点であるプロセッサ間通信オーバヘッド、共通資源へのアクセス競合について、結合方式としてバス結合を取り上げ、プロセッサ間通信方式を、プロセッサ間結合とプロセッサ・メモリ結合によって実現する場合について、稼動プロセッサ数、平均送信待ち合せ時間を評価基準として各種方式について比較・検討を行なった。さらに、機能分散処理方式について、より詳細に方式上の問題点並びに解決法について考察した。

現在、機能分散形マルチプロセッサ構成を用いたシステムが安価なマイクロプロセッサを用いて急速に立上がってきている状況にあり、システム形態により即した方式上の問題点をさらに深く検討する必要がある。

今後の課題としては、バス構成以外の結合方式（ループ，リング，スイッチマトリクス）を用いた場合の評価が必要である。さらに、分散処理の別の形態、すなわちネットワーク分散処理の場合（処理機能とデータベース機能を分散させた形態）についても最近の分散プロセッサの動向、ネットワークOSの構成法等、検討する必要があろう。

最後に、日頃御指導頂く鍛治所長、寺村副所長、深田次長、並びに御討論頂いた情報処理研究室諸氏に感謝いたします。

## 参考文献

(1) “特集分散処理”，情報処理 vol 20，No 3，(昭54)．

(2) 島橋：“並列処理のためのプロセッサ結合方式”情報処理 vol 23，No 3，p 201 (昭57)．

(3) Kiely，S.C.：“An Operating System for distributed Processing－DPPX”，IBM．Syst．J vol 18，No 4，p 507，(1979)．

(4) Bhandarkar，D.P.：“Analysis of memory interference in multiprocessors” IEEE Trans．Comput，C-24 p 897 (1975)．

(5) Marsan，M.A & Gerla，M：“Markov models for multiple bus multiprocessor system” IEEE Trans．Comput，C－31，p 239 (1982)

(6) 藤木，雁部 “通信トラヒック理論”，丸善，(昭55)

(7) 橋田：“情報処理システムにおける待ち行列理論の応用(2)” 情報処理 vol 18，No 3 (昭52)．

(8) 野村，塚本：“ポーリングシステムのトラヒック解析”信学論(B) J 61-B，7，p 600 (昭53)．