

## 交渉により負荷分散を行う方式の評価

小川智之 小林真也

金沢大学 工学部

マルチコンピュータ環境において処理の高速化を図るための負荷分散方式として、ノード間でタスクの依頼とその諾否からなる交渉を行い、必要ならばタスク依頼を受諾したノードがまた別のノードと交渉を繰り返すことで、タスク実行ノードを決定する自律負荷分散方式の提案を行っている。本稿では、提案方式の有効性について検証するため、他ノードに関する情報の制限、ネットワークのセグメント化、処理能力の非均一化の影響について評価を行った。その結果、高速なノードが存在する場合には低負荷での応答時間の改善効果が大きい等の有効性が認められた。

## Evaluation of Load Distribution Method Based on Internode Negotiations

Tomoyuki Ogawa Shin-ya Kobayashi

Faculty of Technology, Kanazawa University

We proposed "Autonomous Load Distribution Method" for processing tasks in high speed on a multicomputer system. This method decides which nodes should execute the task by internode negotiation. The negotiation is done between a node having the task and another node. If the negotiation reaches a settlement, the tasks will be passed from the node having the task to the other and then the other node will negotiate further in case of need. In this paper, we investigate affection of restriction of node information, segmented network or heterogeneity of computer performance to goodness of task allocation of our method by simulation. The simulation results lead to the conclusion that our method can allocate tasks effectively, especially in the heterogeneous system.

## 1. はじめに

FDDI, イーサネット等のネットワークで相互接続されたワークステーション (WS) 等の複数の計算機から構成されるシステムを, マルチコンピュータ環境と呼ぶ。マルチコンピュータ環境では, ユーザが複数の計算機を同時に利用できるため, 独立した複数の処理単位 (タスク) を別々の計算機 (ノード) で実行するという負荷分散を行うことにより, 処理の高速化を図ることができる。

最も簡単な負荷分散方法として, ランダムに選んだノードにタスクを転送し, 実行させる方法が考えられる。しかしながら, 転送先のノードが低負荷であるとは限らないし, また処理速度が遅いノードであるかもしれないため, かえって処理が遅くなる場合もある。従って, 各ノードの負荷状況の動的変化, ノードに固有な処理速度等のハードウェア的な処理能力の違いを考慮し, 最適なノードにタスクを割り当てる必要がある。これらをユーザが行なおうとすると, 変動の大きい負荷状況の把握は困難であり, 負担が大きい。そこで負荷分散処理を自動的に行う負荷分散方式が要求される。

負荷分散は, タスクの到着の仕方から, 静的負荷分散と動的負荷分散とに分けられる。静的負荷分散では, 同時に到着する一定個数のタスクを取り扱うのに対し, 動的負荷分散では, タスクの到着間隔, 所要処理量について統計的なことしか分からず, 予め全てのタスクに関する情報を得ることができない場合を取り扱う。マルチコンピュータ環境では, 様々な所要処理量のタスクが, ユーザによりランダムな時間間隔で投入されるため, 負荷分散は動的負荷分散の範疇に入る。

動的負荷分散方式は, 放送型通信を用いる放送型と, 1対1通信を用いる非放送型に分けられる。放送型としては, Smithの契約ネットプロトコル[1]を用いる方式がある。契約ネットプロトコルでは, まずタスクを持つノードが管理者となり, システム内の各ノードにタスクの要求するリソースなどの情報を含むタスク通知を放送する。放送を受けたノードは自らが持つリソースや状態を考慮し, 実行時間の予測値を含む入札メッセージを返す。管理者はそれらのうち最適な入札者と契約, すなわち両者間でのタスクの授受を行うアルゴリズムである。しかしながら, マルチコンピュータ環境においてトラヒックを増大させる放送型通信を頻繁に行うことは不可能であり, また, 放送の範囲はネットワーク・セグメントに限定される為, 複数のネットワークから構成される大規模なマルチコンピュータ環境に対して, 契約ネットプロトコルは不向きである。

我々研究グループでは, 非放送型動的負荷分散方式として, 各ノードが自律的に他ノードと交渉を行い, 負荷を分散させる自律的負荷分散方式[2]を提案し, 評

価を行ってきた。これまでのシミュレーションによる評価により, 均一なノード10台から構成されたシステムにおいて, 各ノードの平均到着率が均一な場合, 主に以下の2つが示されている。

(1) タスク転送・結果返送に要する通信遅延  $T_c$  の増加に対しては, 平均応答時間は増加するが, 負荷分散を行わない場合の平均応答時間を越えることはない。

(2) 1回の交渉に要する交渉時間  $T_n$  の増加に対して, 平均応答時間は増加するが, 交渉時間がタスクの平均処理時間の6割程度までであれば, 負荷分散を行わない場合の平均応答時間を越えることはない。

本稿では, 上記の評価をふまえ, 更に

(1) 既知ノード数の限定

(2) ネットワークのセグメント化

(3) 処理能力が非均性にともなう影響について評価を行い, 本方式の有効性について評価・検討する。

## 2. マルチコンピュータ環境

ここでは, マルチコンピュータ環境における制約・特性について触れ, この環境に適した負荷分散方式の条件を探る。

マルチコンピュータ環境におけるネットワークは, 複数のユーザにより共有される設備であり, トラヒックを増大させる放送型通信を頻繁に行うことはできない。また, システムの規模が増大し, 複数のネットワーク同士がゲートウェイ, ルータを介して接続されている場合, 放送は単一のセグメントに限られる。このような大規模なシステムでは, ネットワークとこれに接続された全てのノードに関する情報は膨大となり, 1台のノードで集中管理することは不可能である。

一方, マルチコンピュータ環境で処理されるタスクは, 複数のユーザにより投入される。一般に, 異なるユーザにより投入されたタスク間の依存性は低いため, それぞれを独立に扱うことができる。そしてユーザにより投入されるタスクのうち, 特にCPU負荷を増大させる処理として, コンパイルやシミュレーションなどがあり, これらは繰り返して実行されることが多い。

これらの点を考慮すると, マルチコンピュータ環境における負荷分散方式は以下の制約を満足しなければならない。

- ・通信は1対1通信である。
- ・情報は各ノードが局所的に管理する。
- ・各ノードの持つ情報は, システムの一部に対するものである。

また, 本研究では負荷分散の対象となるタスクとして, 以下の仮定を設ける。

- ・それぞれが独立であり, 主にCPU資源を要求す

る。

・所要通信量（ファイルサイズ）、所要処理量（所要処理時間×演算速度）の予測が可能。

後者の仮定は分散対象となるタスクを大幅に限定するように思われるが、タスクの実行毎に、その所要処理量をファイルサイズなど同様にタスクと組にして管理することにより実現できる。

### 3. 自律負荷分散方式

提案を行っている自律負荷分散方式は、ユーザからタスクが投入されたノードが、他のノードと1対1通信を用いてタスクの依頼とこれに対する諾否からなる交渉を行い、依頼を引き受けたノードがまた別のノードと交渉を行うといったことを繰り返すことで、タスク実行ノードを決定し、タスクを渡すことにより負荷分散を行う方式である（図1）。交渉のためのアルゴリズムは、タスクを他ノードに依頼するか、自ノードで実行するかの判断（依頼判断）、及び依頼を引き受けるかどうかの判断（諾否判断）に分けられる。本章では、自律負荷分散方式の依頼・諾否判断アルゴリズム、及びこれら判断に必要な情報の管理について述べる。

#### 3.1 依頼判断

タスクを他ノードに依頼するか、自ノードで実行するかの依頼判断のアルゴリズムを図2.1に示す。これはタスクが投入されたノード（依頼元ノード）、及びタスクを依頼され、受諾したノード（依頼ノード）が行う。

（A-1）所要時間推定に必要な情報を含むノード情報が存在するノード（既知ノード）全てと、自ノードでのタスク実行の所要時間を推定し、これを昇順に並べた候補リストを作成する。ただし、自ノードが依頼ノードである場合には、依頼メッセージに含まれる経路情報から既に交渉に参与しているノードを調べ、候補リストから除く。これは依頼の連鎖が巡回することを回避し、処理時間減少にもつながる。ノード情報には、以下の情報が含まれる。

- ・ハードウェア情報（ノード名,処理速度,通信路特性）
- ・負荷情報（ノードの総プロセス数,総残余処理量）

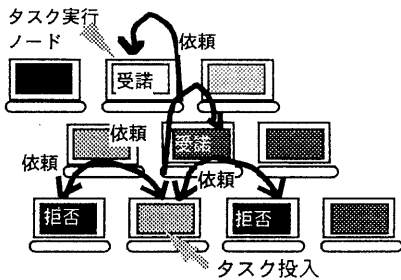


図1 自律負荷分散方式

所要時間の推定方法については本稿では省略する。

（A-2）リストの先頭が自ノードであれば実行ノードを自ノードに決定する。そうでなければそのノードに対し、依頼を行う。依頼先ノードに送る依頼メッセージには、以下の情報が含まれる。

- ・投入タスクに関する情報
- ・依頼元ノードの推定所要時間
- ・交渉に参与したノードのノード情報（経路情報）

これらは依頼先ノードが諾否判断を行い、依頼先ノードの局所的情報データベース（DB）を更新するために必要な情報である。

（A-3）依頼に対する諾否メッセージには、以下の情報が含まれる。

- ・受諾/拒否
- ・依頼先ノードのノード情報

これにより、自ノードの局所的情報DBにある依頼先ノードのノード情報を更新する。依頼拒否の場合にはリストの先頭を削除し、（A-2）を繰り返す。依頼受諾の場合には、交渉は引き続き依頼ノードが行う。

#### 3.2 諾否判断

タスクを依頼されたノードが行う、タスクを引き受けるかどうかの諾否判断のアルゴリズムを図2.2に示す。

（B-1）経路情報を用いて自ノードの局所的情報DBにおける経路上のノードのノード情報を更新し、自ノードのノード情報を経路情報に追加する。依頼されたタスクを自ノードで実行した場合の所要時間を推定し、依頼メッセージに含まれる依頼元ノードの所要時間と比較して短ければ依頼を受諾し、引き続き受諾したタスクに対して依頼判断（A-2）を行う。この諾否判断では、依頼元ノード或いは依頼ノードによる依頼に対し、自ノードについての直接的な返答を行うの

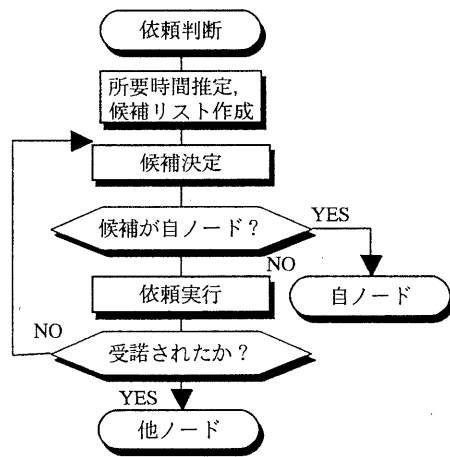


図2.1 依頼判断

で、直接的諾否判断と呼ぶ。

(B-2)(B-1)で受諾しなかった場合、依頼判断(A-1)を行い、候補リスト中、依頼元ノードの推定所要時間より短い所要時間となるノードが存在すれば、依頼を受諾し、受諾したタスクについて依頼判断(A-2, 3)を行う。存在しない場合には依頼を拒否する。自ノードではなく、他ノードのために間接的な諾否判断を行うという意味で、この判断を間接的諾否判断と呼ぶ。

### 3.3 ノード情報の管理

自律負荷分散方式では、交渉に関与したノードのノード情報を含む経路情報を、交渉を行うノード間で相互に回覧することにより、他ノードに関するハードウェア情報、負荷情報を得る。交渉で得た情報は、局所的情報DBに保存する。従って、初期的にはシステム内の全ノードの情報を持つ必要はなく、交渉中に情報の未知なノードの情報を間接的に得ることが可能である。例えば図3に示すように、互いのノード情報を持たないノードAとCが、A, Cいずれの情報も持つノードBを介して交渉を行った場合、ノードAとCの局所的情報DBには互いの情報が追加される。特に大規模なマルチコンピュータシステムではノード情報は膨大となるため、この間接的に他ノードの情報を得る機能は重要であると考えられる。

しかしながら、交渉を繰り返すことにより情報DBの情報は増え続け、DBがあふれてしまう。従って不要になった情報を削除する機能が必要となる。自律負荷分散方式では、情報の信頼性の保たれる期間とし

て、ハードウェア情報には長期的な、ノード情報には短期的な信頼性有効期間を定め、これを過ぎて古くなった情報は削除することでこの機能を実現する。

## 4. 評価

本章では、前章で述べた自律負荷分散方式の評価について述べる。本稿では、以下の条件が本方式の性能に及ぼす影響について評価し、考察を行うことで自律負荷分散方式の有効性を示す。

### (1) 既知ノード数の限定

大規模なシステムでは、2章で述べたように、全てのノードの情報が膨大となるため、各ノードが持ちうるノード情報の量は制限される。そこでこれを限定した場合について評価を行う。

### (2) ネットワークのセグメント化

複数のネットワークからなるシステムに対し問題となる通信路特性の違いについて評価する。

### (3) 処理能力の非均一性

システム内の各ノードの処理能力が異なる場合、これが応答時間に及ぼす影響を調べる。

## 4.1 評価尺度

負荷分散方式を評価する場合、何らかの評価尺度が必要である。以下では、本方式の評価で用いた評価尺度について述べる。

・応答時間：ユーザがタスクを投入してから、結果を取得するまでに要する時間である応答時間の平均。

・利用率：ノードがアイドルでない確率。どのノードにも等しい負荷がユーザから投入される場合、処理能力の均一なシステムであれば、負荷分散によりどのノードの利用率も等しくなれば、偏りないタスク割り当てが行われていることになる。一方、非均一システムの場合、低速なノードよりも高速なノードの方が利用率が上げられ、高速なノードが有効に利用されていることになる。

・自ノード実行率：ノードに投入されたタスク総個数のうち、自ノードで実行されたタスク個数の割合。これが低ければ、投入されたタスクの殆どが他ノードで実行されたことになる。従って、投入されたタスクの分散度を示す尺度である。また、処理能力が均一なシステムでは、タスクが自ノードで実行される最も明

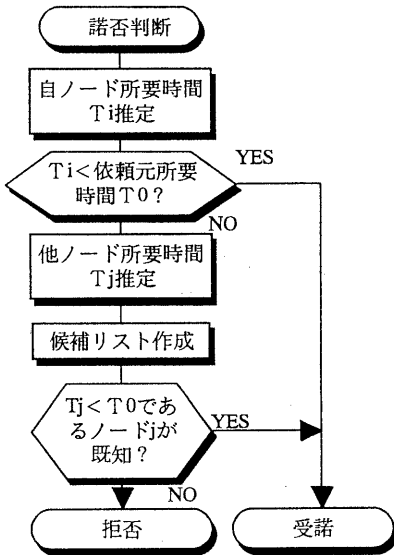


図2.2 諾否判断

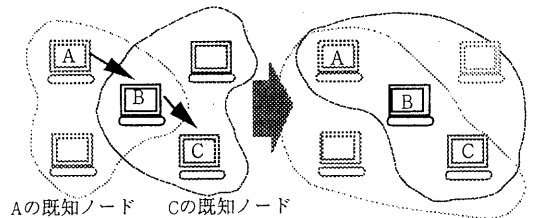


図3 交渉による既知ノード情報の追加

らかな条件は2つ考えられ、自ノードがアイドルである場合と、どのノードもアイドルでない場合である。これらは互いに排反事象であるため、自ノード実行率はこれらいずれかの場合であれば、100%に近くなる。

・依頼回数：個々のタスクにつき、実行ノード決定に要した依頼の回数。交渉に関与したノードの数、或いは応答時間に対する交渉のためのオーバーヘッド量とも考えられる。

#### 4. 2 シミュレーション条件

本方式のシミュレーションによる評価を行う。シミュレーション方法はイベントドリブンとし、CPUスケジューリングはタイムスライスが所要処理量に比べ十分小さいラウンドロビン方式とする。また、簡単化のため、CPU資源を消費するイベントはタスク実行のみとする。シミュレーションは、特定ノードのタスク実行完了のイベント数6万個で打ち切った。

まず基準となるモデルを考え、この一部を変更したモデルと比較することで評価を行う。基準モデルのシミュレーション条件は以下の通りである。

・システム構成：システムは各ノードの演算速度が等しく1[unit/sec.]であるノード10台から構成され、各ノード間で等しい通信路特性を持つ単一のネットワークに接続されていると仮定する。

・投入タスク：タスクの所要処理量は平均1[unit/sec.]の指数分布とする。また、ユーザによるタスク投入の時間間隔(投入間隔)は、指数分布と仮定し、ノードNiの平均到着率 $\lambda_i$ [tasks/sec.]として、以下の2つの場合を想定する。

均一負荷： $\lambda_i = \lambda$  ( $\Lambda = \lambda$ )

非均一負荷： $\lambda_i = \lambda$ ,  $\lambda_i = 0.3$  ( $i=2,3,\dots,10$ )

( $\Lambda = 0.1\lambda + 0.27$ )

ここで、 $\Lambda$ はシステム内のノードのサービス率の総和に対するタスク到着率の総和(システム負荷)であり、次式で表わされる。

$$\Lambda = S \sum_{i=1}^m \lambda_i / \sum_{i=1}^m C_i$$

mはノード数、Sは平均タスクサイズ、 $\lambda_i$ はノードNiに投入されるタスクの平均到着率(他からの依頼により実行されるタスクは含まない)、 $C_i$ はノードNiの処理速度である。システムが安定に動作するためには、システム負荷 $\Lambda$ は1未満でなければならない。

均一負荷は、各ノードに対し、短期的なタスク投入間隔の変動はあるが、長期的にはどのノードも等しいタスク投入間隔である状況を表わす。一方、非均一負荷は、N1以外の周囲のノードが等しく低負荷である状況を表わす。

・通信遅延：タスク転送に要するタスク転送時間Tcはタスクの所要処理量の1割程度、1回の依頼・諾否の交渉に要する交渉時間Tnは所要処理量の1パーセ

ント程度であると考え、Tc=0.1[sec.]、Tn=0.01[sec.]の時間を要すると仮定する。ここで、Tc、Tnどちらも送信側、受信側で等しく1/2ずつ要するとする。

・ノード情報の管理：システム内の全てのノードは、局所的情報DBに全ての他ノードの情報を持つとする。ハードウェア情報は永久に消えず、負荷情報は、負荷情報の一部である総残余処理時間を信頼性有効期間として、これを過ぎた情報は削除する。

#### 4. 3 シミュレーション結果・考察

前節で述べた基準モデルに対し、(1)既知ノード数の制限、(2)ネットワークのセグメント化、(3)ノード処理能力の非均一化を行った場合を評価し、考察する。

(1)既知ノード数の制限

既知ノードが同一であるノードグループを、既知ノードグループと呼ぶことにする。処理能力の均一な10台のノードからなる基準モデルに対し、既知ノード情報を表1のように制限した場合を想定する。すなわち、グループAとCが互いのノード情報をもたない場合である。非均一負荷でN1の到着率 $\lambda$ を変化させた場合、システム負荷に対する各ノードグループの平均応答時間を図4.1に示す。比較のため、

- ・基準モデルでのN1とN2,...,N10(単にN2と表わす)の平均応答時間
- ・負荷分散を行わない場合の理論値(M/M/1)
- ・分散のためのオーバーヘッドが全く無い場合の理論値(M/M/10)

を併せて示す。ただし、M/M/1、M/M/10の理論値は、スケジューリング方式が先着順(FCFS)の場合である。

低負荷では既知ノード数制限の影響は少なく、高負荷ではNA、NBの平均応答時間はそれぞれN1、N2に比べ約4%程度の増加、NCの平均応答時間はN2に比べ約2%程度の減少となった。

図4.2は同じ条件での利用率を示している。シミュレーション条件では非均一負荷、つまりN1の負荷を変化させているため、中程度のシステム負荷でN1の利用率は他のノードに比べ高くなっている。既知ノード数制限による利用率の増減は、平均応答時間についての場合と同じく、NA、NBの利用率はN1に比べ増加、NCの利用率はN2に比べ減少している。

既知ノード数制限により、特にNBのN2に対する利

表1 既知ノードグループ

グループ	所属ノード	既知ノード
A	N1	A,B
B	N2,N3,N4	A,B,C
C	N5,N6,N7,N8,N9,N10	B,C

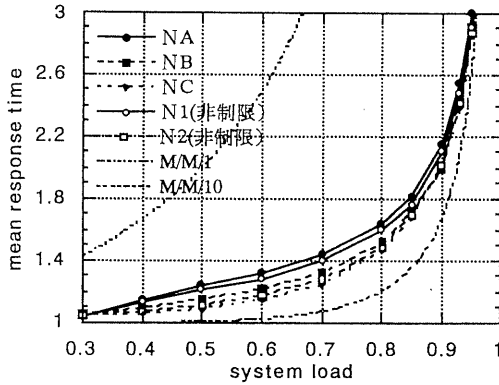


図4.1 既知ノード制限の平均応答時間への影響

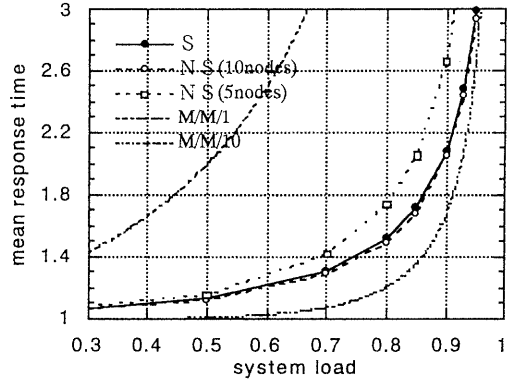


図5.1 セグメント化の平均応答時間への影響 (均一負荷)

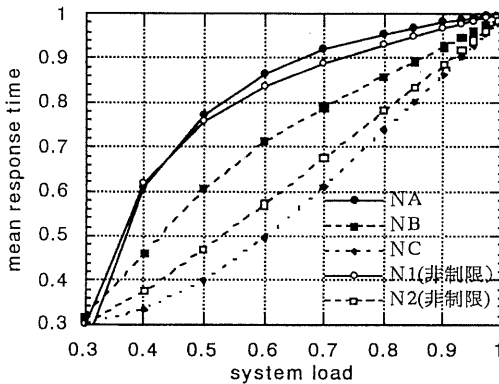


図4.2 利用率

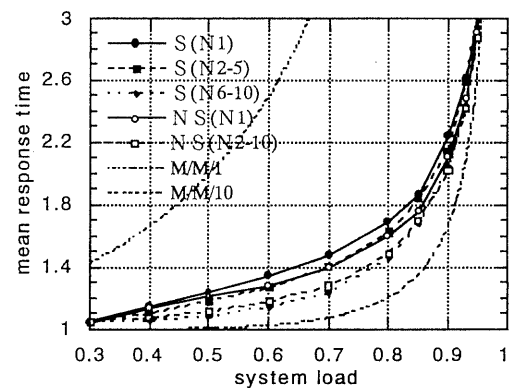


図5.2 平均応答時間 (非均一負荷)

用率の増加が約20%程度と大きい、これは、NAがNCの情報を持たないため、NCよりもNBにより多くのタスクが割り当てられていることを裏付けている。

なお、依頼回数についても調べたところ、既知ノード数制限による増減は、低負荷時を除いて、NA、NBに投入されたタスクの平均依頼回数はN1に比べ増加し、NCに投入されたタスクの平均依頼回数はN2に比べ減少している結果となった。既知ノード数を制限した場合、NAからNCに対する依頼はNBを必ず介して行うため、NAに投入されたタスクの依頼回数は基準モデルでのN1に投入されたタスクに比べ増加する。高負荷時には特にシステム中のアイドルノードが少なく、NCの6台にもまわされるタスクが増えるため、NAの依頼回数は増加する。しかしながら、その増加は最大で0.2回程度 ( $\Lambda=0.9$ で3.3→3.5回)であり、従って応答時間にもさほど影響していないと考えられる。

## (2) ネットワークのセグメント化

基準モデルのネットワーク構成を変更し、2つのセグメントS1、S2に分け、それぞれ5台ずつのノード

が接続されているシステムを考える。すなわち、N1、N2、...、N5がS1に、N6、N7、...、N10がS2に接続されているとする。セグメント内の通信路特性は基準モデルと同じく、タスク転送時間  $T_c=0.1[\text{sec}]$ 、交渉時間  $T_n=0.01[\text{sec}]$ とする。一方、セグメント外の通信路特性は、セグメント内の2倍要するとし、 $T_c=0.2[\text{sec}]$ 、 $T_n=0.02[\text{sec}]$ とする。

タスクの到着率は、全てのノードの到着率を変化させる均一負荷の場合と、N1だけの到着率を変化させる非均一負荷の場合を想定する。以下、それぞれの評価結果について述べる。

### A. 均一負荷の場合

均一負荷の条件で、上述のセグメント化されたシステムと、基準モデル、及び基準モデルでノード数を5台に変更したモデルとの比較を行った。図5.1に平均応答時間を示す。S、NSはそれぞれセグメント化を行ったモデルと行わないモデルを示す。基準モデルにくらべ、セグメント化されたモデルの平均応答時間の増加は約2%と少なく、セグメント化の影響はほとんど無視できることがわかる。

また、ノード数5台のセグメント化されていないモ

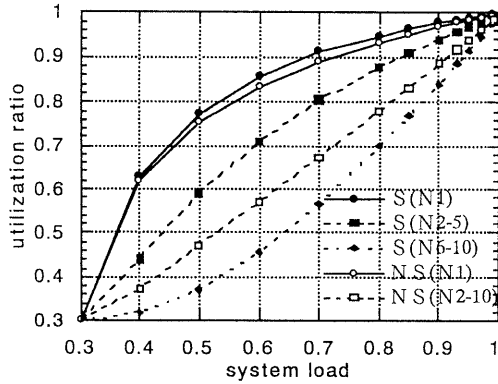


図5.3 利用率 (非均一負荷)

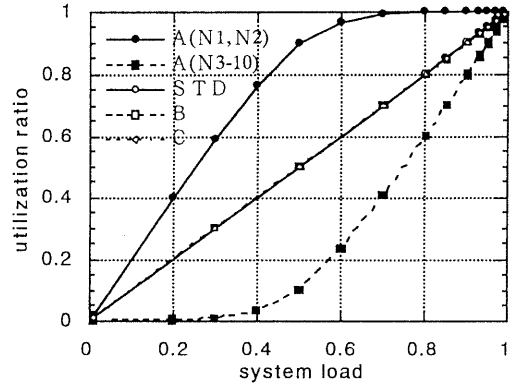


図6.2 利用率

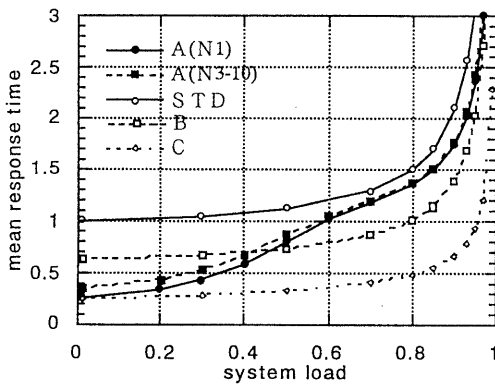


図6.1 処理速度の非均一性が応答時間に及ぼす影響

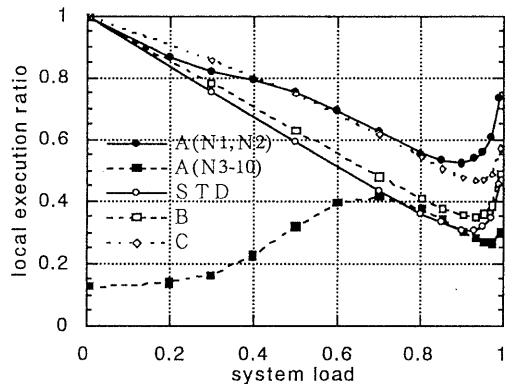


図6.3 自ノード実行率

デルと比べると、セグメント化された10台のノードの平均応答時間の改善は大きく、システム負荷 $\lambda = 0.9$ で約30%である。従って、5台のノードからなる2つのシステムでそれぞれ独立に負荷分散を行った場合よりも、5台ずつにセグメント化された10台のノードからなるシステムの方が応答時間の改善効果は大きいといえる。

#### B. 非均一負荷の場合

N1だけの到着率を変化させる非均一負荷の条件で、セグメント化されたモデルと、基準モデルとの比較を行った。平均応答時間を図5.2に示す。S(N1)、NS(N1)等のSはセグメント化されたモデル、NSは基準モデルを示し、括弧内はノードを示す。

低負荷( $\lambda = 0.3$ )ではセグメント化の影響はほとんど無いといえる。中～高負荷では、S(N1)はNS(N1)に比べ7%程増加、S(N2-5)はNS(N2-10)に比べ10%程増加している。一方、S(N6-10)の平均応答時間はNS(N2-10)に比べ3%程減少していることがわかる。

図5.2にこの場合の利用率を示す。S(N1)、NS(N1)等の表記は前述と同様である。セグメント化によりN1、N2-5の利用率は増加し、N6-10の利用率は減

少している。従って、基準モデルではN6-10に割り当てられていたタスクの多くが、セグメント化によりN2-5に割り当てられるようになったと言える。

#### (3) 処理能力の非均一性

システムを構成するノードの処理能力が不均一な場合を考える。ノードの処理能力以外は基準モデルと等しいとし、ノードが表2に示す処理能力であるシステムを評価する。システムAが不均一システムであり、これとシステムB、システムC、および基準ノード(処理能力は均一で1[unit/sec.])との比較評価を行う。システムBは総処理能力はシステムAと等しいが、各ノードは均一処理能力であり、システムCは全てのノードの処理能力がシステムA中の最も高い処理能力を持つノードと等しい場合である。ここでは、

表2 処理能力

システム	N1, N2	N3~N10	総処理能力
A	4	1	16
B	1.6		16
C	4		40

ノード毎のタスクの到着率が等しい均一負荷の条件でシミュレーションを行った。平均応答時間を図6. 1に示す。図中、STDは基準モデルを示す。

図6. 1より、システムAの平均応答時間は、低負荷ではもっとも処理能力の高いシステムCに近く、中～高負荷では処理能力の低い基準モデルとシステムBとの中間的な値をとり、高負荷ではシステムBに近づくことがわかる。

図6. 2は、各システムの利用率を示している。この図より以下のことが言える。まず不均一システムAは、低～中負荷にかけて高速なノードの利用率が上がるが、低速なノードの利用率は殆ど0に近いことから、高速なノードが低速なノードに優先して利用されると言える。一方、高速なノードの利用率が1に飽和する中負荷からは、低速なノードの利用率が上がり始めており、高負荷では低速なノードも利用されるようになることがわかる。

図6. 3は各システムの自ノード実行率である。この図より以下のことがわかる。まず均一システムに対してであるが、自ノード実行率は、4. 1で述べたように、均一システムでは、自ノード実行率が高ければ、自ノードがアイドルであるか、どのノードもアイドルでないかのいずれかの場合である。均一システムに対するシミュレーション結果も、自ノードがアイドルである確率の高い低負荷と、どのノードもアイドルでない確率の高い高負荷(0.9以上)で高くなっている。また、処理能力が高いシステムほど、自ノード実行率の特性は全体に高くなっている。これは、タスクの平均サービス時間(タスクサイズ/演算速度)に対するタスク転送時間の割合が、演算速度の増加により相対的に増加することが原因である。すなわち、本方式では所要時間推定の際にタスク転送時間も考慮するため、ノードが高速である程、平均サービス時間に対するタスク転送時間の割合が増加し、自ノード実行率は高くなる。

次に、不均一システムに対しては、高速なノードは、均一システムと同様な自実行率の変化であるが、低速なノードは、低負荷でも低い自実行率であることがわかる。これは、高速なノードに優先的にタスクが割り当てられるという利用率の結果を裏付けている。

## 5. まとめ

本稿では、提案を行っている自律負荷分散方式を複数のセグメントからなる大規模なマルチコンピュータ環境に適応した場合、問題となると考えられる、(1) 既知ノード数の限定、(2) ネットワークのセグメント化、(3) 処理能力の非均一性が性能に及ぼす影響について評価を行い、大規模なシステムにも適応可能であるという本方式の有効性の検証を行った。

自律負荷分散方式とは、マルチコンピュータ環境に

おいて、独立した複数のタスクを別々のノードで実行するという負荷分散により処理の高速化を図る際、タスク実行ノードの決定の自動化を目的としている。本方式では、ユーザからタスクが投入されたノードが、他のノードと1対1通信を用いてタスクの依頼とこれに対する諾否からなる交渉を行い、依頼を引き受けたノードがまた別のノードと交渉を行うといったことを繰り返すことで、タスク実行ノードを決定し、タスクを授受することにより負荷分散を行う方式である。

自律負荷分散方式に対し、システム内のノード全てのノード情報を持ち、単一のネットワークに接続された、処理速度が均一なノード10台(N1,N2,...,N10)のモデルと、これに前述の既知ノード数の限定、ネットワークのセグメント化、処理能力の非均一性等の条件を加えたモデルとの比較評価を行った結果、以下のことが示された。

(1) 非均一負荷で、交渉時間 $T_n=0.01[\text{sec}]$ で、N1とN5,...,N10は互いの情報を持たないという制限による平均応答時間の影響は、低負荷では少なく、高負荷でも数%程度である。

(2) セグメント外の通信路特性は、セグメント内の2倍要するとした場合、均一負荷( $\lambda_i = \lambda$  ( $i=1,2,\dots,10$ ))では、セグメント化による平均応答時間の影響は少ない。非均一負荷では、特にN1が属するセグメントの他ノードの応答時間が増加(10%程度)し、セグメント外のノードの応答時間は減少する。しかしながら、この影響は高負荷であるほど少ない。

(3) 処理速度が等しい基準ノード8台と、その4倍の速度である高速ノード2台という非均一システムでは、均一負荷の条件で、タスクは高速ノードに優先的に割り当てられ、低負荷では高速ノード10台の特性に近くなり、高負荷では、基準ノード10台の特性に近づく。

以上の評価結果から、上述の条件の下では、既知ノード情報の制限による影響は少なく、セグメント化に対しては、特に高負荷なシステムの場合、セグメント化しても多くのノードで負荷分散を行うことにより、高速化に有効であり、また、高速な計算機が少数台あるようなシステムでも、低負荷であれば高速なノードの応答が得られるといった有効性が明らかとなった。今後、本方式の実装が課題として残されている。

## 参考文献

- [1] Reid G. Smith: "The Contract Net Protocol: High-level Communication and Control in a Distributed Problem Solver", IEEE Transaction on Computers, Vol.C-29, No.12 (1980)
- [2] 小川智之, 小林真也, 木村春彦, 武部 幹: "自律的な負荷分散方式の評価", 進学技報 Vol.93, No.436 (1994)