計算場における分散プロセスの準最適配置

上原 稔† 所 真理雄††

広域分散環境では，遠距離通信における通信遅延を無視して効率よい分散計算は実現できない．そこで，その通信遅延を考慮した計算モデルとして計算場モデルが提案されている．計算場では，プロセス間にグルーピングのための引力と負荷分散のための斥力を作用させ，そのバランスによりプロセスを最適な位置に配置し，効率よい分散計算を実現する．本論文では，計算場において協調プロセスを準最適位置に配置するためのアルゴリズムを提案し，その評価を行う．ここでは協調プロセスの最適配置問題を物理的メタファーに基づく負荷分散とグルーピングの統一というアプローチで解決する．その結果，本アルゴリズムが理論的上限である静的配置と比べて遜色ない性能を引き出せることを示す．

A Suboptimal Allocation for Distributed Processes in Computational Field

Minoru Uehara† and Mario Tokoro††

In wide area distributed environments, we need to take communication delays into account in order to achieve efficient distributed computing. Computational Field Model (CFM) is proposed as the computational model for such computing environments. In this model, a suboptimal allocation is realized by allocating each process at the position where two kind of forces balance: attractive force for grouping/clustering and repulsive force for load sharing. In this paper, we propose a dynamic algorithm for allocating distributed processes to their suboptimal positions in CFM and evaluate it. We solve this problem by employing the cost function which is integrated with load and communication delay. We show that our algorithm gives almost the same performance as that is given by static optimal allocation.

1. はじめに

近年，ワークステーションの低価格化とローカルエリアネットワーク（LAN）の普及を背景に，それらを相互接続した広域ネットワーク（WAN）が発達してきた．これと同時に，WAN上で透過な広域分散環境を実現する研究も行われてきた[16]．広域分散環境は，以下の特徴を有する：

1. 開放性

広域分散環境では，システムの構成要素やその結合が時々刻々と変化する．このように，構成要素やそれらの結合が動的に変化することを開放性といい，そのような分散環境を開放型分散環境という．

2. 地理性

広域分散環境では，通信遅延時間を無視した透過性が通信コストの増大を招き，計算全体の応答時間を著しく遅らせる可能性がある．このように，その計算コストが構成要素の地理的配置に依存することを地理性といい，そのような分散環境を地理的分散環境という．

開放性に関する研究はさまざまな分野で既に数多く行われている[14),17)]．そこで，本論文では，主として地理性に関する問題を扱う．

地理的分散環境では，計算全体の応答時間が計算資源の地理的配置に依存するため，計算資源の最適配置が必要になる．ここでは，計算の要求がシステムに到着してからその計算が終了するまでの経過時間を応答時間と呼び，応答時間が最小となる資源配置を最適配置と呼ぶことにする．地理的分散環境における応答時間は通信遅延が無視できる非地理的分散環境と比べて増加する．そのため，地理的分散環境では通信遅延を含めた応答時間の評価が必要となる．

† 東洋大学工学部情報工学科
Department of Information and Computer Sciences, Faculty of Engineering, Toyo University
†† 慶應義塾大学理工学部
Department of Computer Science, Keio University

いま，計算資源としてプロセスに注目し，目的の計算が互いに通信し合う複数の協調プロセスで構成されていると仮定する．また，プロセスはどの位置で計算しても同じ結果を返すと仮定する．すなわち，プロセスの再配置により応答時間だけが変化する．そこで，プロセスを一箇所に集中して配置すると，通信遅延は減少するが，その周辺の処理装置の負荷が高まり，応答時間を増加させる可能性がある．逆に，プロセスを分散して配置すると，処理装置の負荷は分散されるが，通信するプロセス間の平均距離が長くなり，通信遅延が増大してしまう．このように，負荷と通信遅延は相反する性質を持ち，一方のみを減少させても応答時間は改善できない．したがって，負荷と通信遅延の両方を考慮して，最適配置を求める必要がある．そのような最適配置を動的に計算するアルゴリズムの提案が本論文の主目的である．

ここで，厳密な最適配置とは限らないが，応答時間が最小に近い資源配置を準最適配置と呼び，準最適配置を実現するアルゴリズムを準最適配置アルゴリズムと呼ぶことにすると，本論文で求めるアルゴリズムは準最適配置アルゴリズムである．我々の目標が広域分散環境にあることは既に述べたとおりである．広域分散環境の一側面である開放型分散環境では，厳密な最適状態を実現しようとすると，全体の状態を把握するために多大なコストを要するため，本来の計算以上に最適配置計算のコストが費やされる可能性がある．このため，最適化の度合を準最適状態に緩和し，その範囲で可能な限り最適状態に近い準最適配置を探す．今後，特にことわらない限り準最適配置の意味で最適配置という語を用いる．

我々の最適配置アルゴリズムのアイデアは以下のとおりである．最適配置は通信遅延と負荷の相対的な関係で決定されると予想される．そこで，通信遅延に比例した引力と処理装置の負荷に比例した斥力を仮定し，それらのバランスにより最適配置を実現する．すなわち，個々のプロセスは通信情報に基づいて他のプロセスとの間の引力（通信遅延に比例した力）を計算し，同時に全方位へ作用する斥力（その近辺の処理装置の負荷に比例した力）を求め，そのベクトル和で移動する方向と量を決定する．その過程を繰り返すことで全体として最適配置へ収束していく．

このようなプロセスの集団的振舞は生態的計算(ecological computing)[7] もしくは創発的計算 (emergent computing)[3] などと呼ばれ，分散計算の重要なテーマとなっている．これらの分野では，比較的単純な構成要素が全体的な集団として示す複雑な動作を研究する．このためしばしば自然系との比較がなされた[8),10)]．本論文においても，当初，プロセスの集団的振舞を自然現象に対応させ，力学的意味を与えようと試みた．しかしながら，プロセスの振舞に対応する適切な自然現象は見い出せず，この試みは成し得なかった．そこで，コスト関数の意味を，計算場という仮想世界の独自の法則の中に見い出していく方針を採った．

前述のアルゴリズムにおいて重要な問題点は，力のバランス点が最適配置となることの保証が必要なことである．この証明は通常以下の過程で行われる．(1)応答時間の定式化を行う．(2)応答時間の理論的最小値を求める．(3)力の式を立てる．(4)力がバランスする配置を求め，そのときの応答時間を求める．この結果が(2)と一致すれば命題は証明される．しかし，この証明過程において，力の式からそのバランスする配置を求める過程は多体問題であり，解析的手法の適用は困難である．また，この方法では力の立式と証明を適切な式を発見するまで繰り返さなくてはならない．これは大変な手間である．したがって，適切な力は最適配置から逆算されることが望ましい．この逆算過程を容易に行うために，我々は最適配置アルゴリズムを一般的な勾配法に限定し，評価関数に議論を集中することにした．ここではもっとも身近な勾配法の例として古典的力学系を比喩に説明する．このような力学系では，物体にポテンシャルエネルギーの勾配に比例した大きさと向きで力が作用し，さらに摩擦などが作用すると，エネルギーが最小となる方向へ推移する．これを，最適配置計算に対応付けると，応答時間は最小とすべき目標であり，エネルギーに対応する．したがって，応答時間の勾配に比例した力が作用するプロセスは応答時間を極小とする(準)最適配置へ移動する．勾配法が局所的な極小値へ収束することは周知の事実であるため，極小化の保証は勾配法自身により与えられる．残る問題は応答時間の勾配を求めるために応答時間を定式化することである．応答時間の定式化はモデルを定め，そのモデルに基づいて応答時間を計算する式を組み立てる．そこで，我々は，地理的分散環境における分散計算のコストを見積もるために計算場モデル[12)]に基づく計量モデルを考えた．しかし，定式化された応答時間を解析的に解くアプローチは組合せの複雑さのために断念せざるをえなかった．そこ

で，応答時間を数値的に求めることにした．

本論文の構成は以下のようになっている．第2章では，関連研究について述べる．第3章では，計算場に基づく協調プロセス群のモデルと計量法を提案する．第4章では，静的最適配置を解析的に求める．第5章では，動的準最適配置アルゴリズムの評価関数を決定する．第6章では，本アルゴリズムの有効性を検証する評価を行う．最後に結論を述べる．

2. 関連研究

本研究の関連分野は負荷分散やグルーピングなどの資源配置問題を中心とする．これらの分野の中で我々が仮定する計算機環境にそのまま適用できるものはあまり多くない．第一に，開放型分散環境に対応するためには動的アルゴリズムでなければならない．例えば，負荷分散では，文献1)，2)，9)，11) などがある．しかし，これらのアルゴリズムはタスクが単一のプロセスであることを仮定したもので，協調プロセスのような一つのタスクが複数の互いに通信し合うプロセスから構成される系を考慮していない．地理的分散環境における協調プロセスに基づく分散計算では，通信遅延が生じるため関連するプロセスを地理的な近傍に配置して通信遅延を減少させることが重要になる．この目的には参照関係に基づくグルーピング[15]だけでは不十分である．

また，協調プロセスをサポートするアルゴリズム[4),10)]もあるが，いずれも通信遅延を考慮したものではない．文献4) は分割統治問題をトップダウンでOR並列に解く場合に有効であるが，協調プロセスの通信トポロジーがより複雑な系には適用できない．文献10) は通信木を自己組織化するためより広範囲の問題に適用できる．文献4) が木構造の問題を木構造のアーキテクチャーで解くのに対して，文献10) は一般的な問題を木構造のアーキテクチャーで解く．広域分散環境は開放型分散環境でもあるためより広いアーキテクチャーに適応できる必要がある．

そこで我々は計算場[12)]の持つ高い抽象化能力を利用し，アーキテクチャーに依存しない動的最適配置アルゴリズムを提案した[13),18)]．しかし，これらは性能の改善を示したのみで，理想的な最適配置との隔たりなどの理論的限界には言及していない．

3. 計算場の計量モデル

ここでは，計算場における協調プロセス群のモデルを定義し，その一般的性質について述べる．本論文で対象とするタスクは内部に並列性を有し，複数の相互に交信しあう分散プロセスの集合として表現される．このようなプロセスを分散協調プロセスと呼ぶ．個々のプロセスは，以下のような特徴を持つ．

- 並行に動作する．
- 他のプロセスとメッセージを介して同期通信する．送信者は受信者が返信する承認メッセージを受けとるまで待つ．
- 複数のアクティビティを持たない．
- 計算場の座標を持ち，自由に移動 (migrate) できる．
- 近辺の負荷情報を読みとる．

このような協調プロセス群を特徴付ける諸元について述べる．本モデルでは，タスクとプロセスの単位があり，それぞれ以下のように表される．

タスク タスク T はプロセスの集合 P で表される．また，その統計的特徴は，$\langle C, \varepsilon, v_m \rangle$ で定義される．ここで，C は通信関隔の分布関数，ε は通信頻度行列，v_m はメッセージの通信速度である．なお，ε は，ε_{ij} がプロセス p_i から p_j への平均通信頻度を表すような行列である．また，タスク中のプロセスの総数を $N=|P|$ で表す．

プロセス プロセス $p_i \in P$ は，組 $\langle x_i, m_i, v_i, E_i \rangle$ で表される．ここで，x_i は計算場におけるプロセス p_i の位置座標，m_i はプロセスの移動に関する慣性，v_i はプロセスの単位慣性あたりの移動速度，E_i はイベントの履歴である．また，E_i はイベント列により $E_i=e_1e_2\cdots e_n$ と表せる．イベント e_j は生成 c，送信 s_k，受信 r_k，終了 t のいずれかである．また，n をそのプロセスのイベント数と呼ぶ．

計算場 計算場 F は，組 $\langle G, X, L \rangle$ で表される．ここで，G はネットワークで，処理装置やその結合を表す．X は場の定義域，L は負荷関数である．

本論文の評価では，通信間隔 C は時間的に変化せず，指数分布とする．また，通信頻度行列は一様 ($\varepsilon_{ij}=const$) とする☆．プロセスの応答時間はその終了イベントの時刻であり，タスクの応答時間はそれを構成するプロセスの応答時間の最大値である．応答時間は

☆ これはワーストケースに近いアプリケーションの特徴を表現したものであり，この前提によりマクロなアルゴリズムでは個々のプロセスの通信の偏りを無視してアルゴリズムを大幅に簡略化することができるが，これは本意ではない．

個々のプロセスのイベント数と配置に依存する．また，システムの利用率は応答時間とプロセスの分布範囲に関連し，安定な系ではイベント数が無限に近づくにつれて一定値に収束する．本論文では，不安定な系は扱わない．

本論文では，すべてのプロセスの慣生 m_i を1とし ($m_i=1$, *for all i*)，プロセスの移動速度 v_i とメッセージの移動速度すなわち通信速度 v_m を等しいものとする ($v_m=v_i$, *for all i*)．これはメッセージとプロセスのサイズが近いという粒度的仮定ではなく，どのようなサイズの通信も単位距離あたりの遅延時間は同じであることを意味する．これはネットワークの通信能力を最大限理想化したためである．同じ理由から通信における衝突は起こらないものとする．この仮定は強すぎるが，実際のシステムの多様性は広く，それを特定化した評価はかえって可能性を摘むおそれがある．むしろ上限を求める意味で本研究ではこのような設定を行った．

次に計算場を特徴付ける諸元について述べる．本論文では，整数座標系の一次元離散場を採用し，隣接プロセス間の距離は1とする．負荷関数 $L(x)$ は座標 x に位置するプロセスの数とする．

地理的分散計算には通信遅延があり，これを表現するには従来の並行計算モデルでは不十分である．我々は以前，分散計算モデルとして接触通信モデル[19]を提案したが，ここではその主旨に沿って別の簡易モデルを用いる．並行計算モデル ν-calculus[5),6)] を借用し，それに以下の変更を加える．第一にメッセージを媒体とする同期呼び出しを採用し，第二に移動能力を持たせる．なお，ここでいう同期呼び出しとは返答メッセージを待って進む計算方式である．

本論文では，通信とはメッセージパッシング，すなわち送信側から受信側に何らかのメッセージが伝達されることと定義する．したがって，送信側の送信行為とそれと対になる受信側の受信行為を別個の通信と考えず，合わせて一つの通信行為とみなす．ここで通信の開始時間から終了時間までの時間をその通信における通信時間と呼ぶことにすると，通信時間は，異なるノードへメッセージを転送するのに要する時間（通信遅延）と同期による待ち時間（同期遅延）の最大値で決定される．通信遅延が無視できる通信を，局所通信と呼び，そうでない通信を遠隔通信と呼ぶ．このような定義においては，地理的分散システムでは，局地的な通信でなければ局所通信になり得ない．また，バッファなどを介して少なくとも一方の通信者の同期遅延を解消した通信を非同期通信と呼ぶ．

また，呼び出しとは，呼び出し側が要求メッセージを送信してからそれに対する返答メッセージを受信するまでの行為を意味する．呼び出しは往復二回の通信で実現される．このとき，呼び出し側が返答メッセージを受信するまでブロックされる通信形態を同期呼び出しと呼ぶ．本論文において同期呼び出しを採用した理由は，非同期系では応答時間による時間的コストのほかにメモリ消費などの空間コストを考慮して計算コストを定義する必要があるため，公平な比較が難しいためである．また，後述のアルゴリズムにおいてメッセージの交信と同時に通信頻度を交換するために返答メッセージを必要とする．

さらに，プロセスが新たに獲得した移動能力により地理的分散モデルでは非地理的分散モデルより多様な通信形態が考えられる．**図1**は，本モデルにおける通信を表す時空間ダイアグラムである．同図では，通信の前後で位置を保存する固定型同期呼び出し(a)と移動を伴う移動型同期呼び出し(b)を示している．また，プロセスの状態は線種で表す．実線は活性状態を，点線は待ち状態を表す．なお，斜めの実線はプロセスの移動を表し，斜めの矢印はメッセージを表す．

固定型同期呼び出し(a)は以下のように行われる．送信側はメッセージを送信すると待ち状態になる．メッセージが受信されると受信側から送信側へ承認 (acknowledgment) メッセージを送る．送信側が承認メッセージを受信して終了する．

また，プロセスは通信の前後に移動することで移動型同期呼び出し(b)を実現できる．その場合，送信側はメッセージを送信した後，直ちに移動する．移動だけは並行に行うことができる．メッセージには移動先が記されており，受信側は新たな移動先に承認メッセージを送信する．受信側もメッセージを受信した後

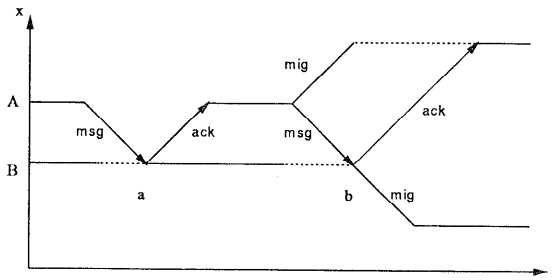


図1 分散計算におけるプロセス間通信の時空間ダイアグラム
Fig. 1 Time-space diagram of process communication.

に移動できる．動的最適配置では，承認メッセージには反作用を伝達する効果がある．

4. 静的最適配置

本論文の目的は，動的に最適配置を実現するアルゴリズムを提案することにあるが，そのためには最適配置状態を確認する必要がある．そこで，本章では静的に最適配置を求め，その特徴をまとめる．

冒頭で述べたように最適配置を求めるには応答時間の定式化が必要である．そこで，ここではタスクの応答時間を解析的に求める方法について述べる．ただし，解析的手法は単純なタスクにしか適用できない．ここでは例としてプロセス数 $N=2$ の場合について説明する．

はじめに通信速度が無限かつ2プロセスの場合について考える．通信速度が無限の場合，送信側 (Sender : S) と受信者 (Receiver : R) は同じ条件で待たされ，対等の関係にある．2プロセスでは，通信と通信の期間で同じ条件が繰り返し適用できるため計算が容易である．2つのプロセス p_1, p_2 の処理時間をそれぞれ t_1, t_2 とすると，その期間の長さは $max(t_1, t_2)$ になる．t_1, t_2 はどちらも平均，分散共に1の指数分布に従うとすると，$max(t_1, t_2)$ の平均値，すなわち1周期の平均応答時間 (CMRT) は以下の式で表せる．

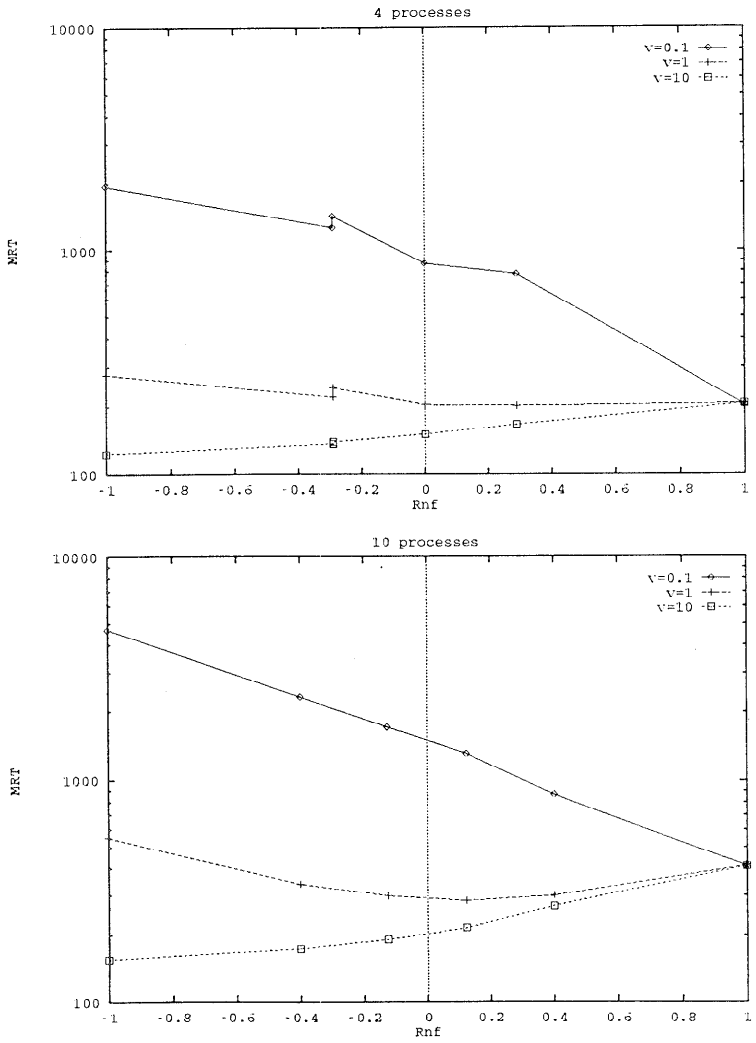


図 2 様相と応答時間の関係
Fig. 2 MRT vs. configuration r_f.

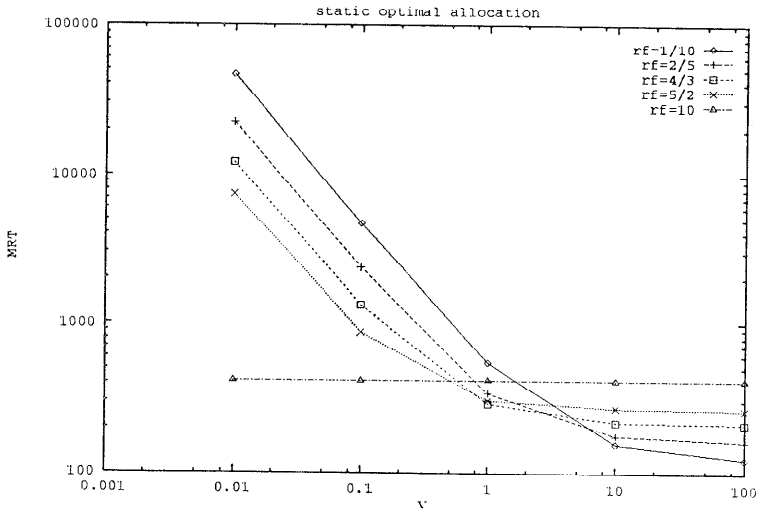


図 3 通信速度と静的最適配置の関係
Fig. 3 MRT vs. v in static optimal allocation.

$$CMRT=\int_0^\infty T\left\{\int_0^T e^{-t_1}dt_1e^{-T}+\int_0^T e^{-t_2}dt_2e^{-T}\right\}dT$$
$$=\frac{3}{2}$$

また一方の処理時間は $min(t_1, t_2)$ となり，その平均は $2-3/2=1/2$ である．したがって，プロセッサの利用率は $2/(3/2+3/2)=2/3\simeq 66\%$ となる．

次に2プロセスで速度が有限の場合について考える．通信速度が有限の場合，送信側は移動のために往復分 2τ だけ余計に通信遅延が生じる．このため CMRT が 2τ 以下になることはない．また受信側と送信側で分布が非対称になる．受信側の密度関数は $exp(-t)$ のままであるが，送信側の密度関数は $t<2\tau$ では0になり，$t\geq 2\tau$ では $exp(-(t-2\tau))$ になる．

送信側が max となるときの平均値 $\overline{T}_s$ は，

$$\overline{T}_s=\int_{2\tau}^\infty\left\{T\int_0^T e^{-t}dte^{-(T-2\tau)}\right\}dT$$

受信側が max となるときの平均値 $\overline{T}_r$ は，

$$\overline{T}_r=\int_{2\tau}^\infty\left\{T\int_{2\tau}^T e^{-(t-2\tau)}dte^{-T}\right\}dT$$

となる．したがって全体の CMRT は，

$$CMRT=\overline{T}_s+\overline{T}_r$$
$$=1+2\tau+\frac{1}{2}e^{-2\tau}$$

となる．またプロセッサの利用率は，$1/(1+2\tau+exp(-2\tau)/2)$ となる．

プロセス数が増えると一回の通信における組み合わせの数が膨大になり，このような計算を行うことが困難になる．そこで，シミュレーションによって求めた結果を**図2**に示す．

ここで，図2で用いられている様相と偏平率 (r_f) について説明する．計算場上のプロセスの空間分布を様相と呼び，各座標に位置するプロセス数を列挙して表現する．例えば，1次元計算場で整数座標に4プロセスを配置する場合，(1, 1, 1, 1)，(1, 2, 1)，(1, 1, 2)，(2, 2)，(1, 3)，(4)の6通りの様相が考えられる☆．最適配置は通信速度に依存し，通信速度が大きければ横に広がり，逆に通信速度が小さければ縦に重なることが予想される．そこで，通信速度に対する変化を明らかにするために，偏平率 (r_f) を導入し，プロセスの配置を順序付ける．

様相 $(a_1, a_2, \cdots, a_m)$ の偏平率 r_f は，$r_f=\max(a_i/m)$ で求められる．したがって，N プロセスの偏平率は主として $r_f\in[1/N, N]$ となる．ただし間に0を含む様相を許せば $r_f\in(0, N]$ となる．なお，異なるプロセス数での評価を同一スケールで比較するため範囲を $[-1, 1]$ (または $(-\infty, 1]$) に正規化した正規化偏平率 ($r_{nf}=log_N r_f$) を用いる場合もある．

図2が示すように，最適配置は通信速度 v に依存し，様相の偏平率に沿って遷移する．$v=10$ の場合，広く分散して配置するとき ($r_{nf}=-1$)，応答時間が最小となる．同様に，$v=0.1$ の場合，1か所に集中して配置するとき ($r_{nf}=1$)，$v=0$ の場合，$r_{nf}=0$ 近辺

☆ これ以外の組み合わせも考えられるが，重要ではない．

でそれぞれ最適配置になる．

静的最適配置は，N個のプロセスの取り得るすべての様相を評価し，通信速度の各値において応答時間が最小となる配置を選択することで求められる．ここでは，10プロセスの場合における静的最適配置を**図3**に示す．

5. 動的最適配置

本章では最適配置アルゴリズムのための評価関数について検討する．ここでは，第1章で述べたように，基本アルゴリズムを定義した後，評価関数について議論する．基本アルゴリズムは以下のとおりである．

基本アルゴリズム 各プロセスiは，通信頻度ε_{ij}，通信相手の位置座標m_jを持つ．メッセージには送信プロセスの位置座標が含まれる．同期呼び出しにより，送信および受信の度にこれらの情報を更新する．受信プロセスは受信時に自分の所有する情報を用いて統一コスト$C_i(x)$とその最小となるx_{min}を計算する．さらにある閾値Tを参照し，$C_i(x_i)-C_i(x_{min})>T$ならば$x_i'=x_{min}$とし，そうでなければ$x_i'=x_i$とする．受信プロセスは同期の後自分の新たな位置を含む承認メッセージを送信し，x_i'に移動する．なお，負荷は個々の位置で場より得られる．

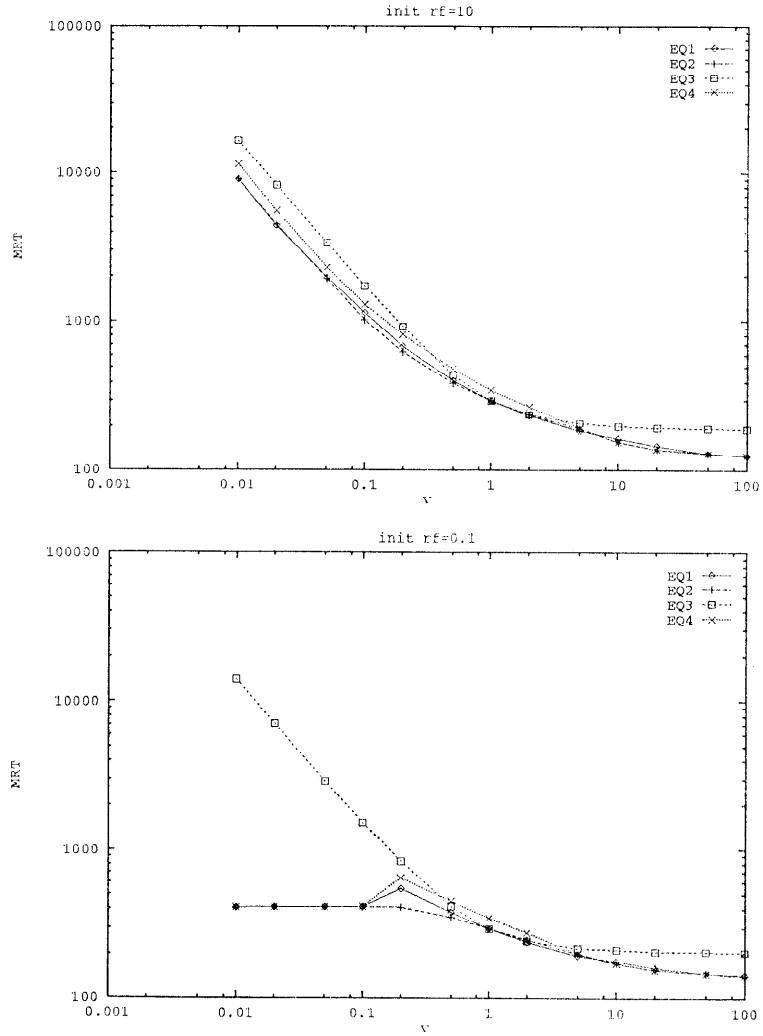


図4 コスト式の比較
Fig. 4 The comparison with cost expressions.

このアルゴリズムは，原理上最適配置へと収束するが，地理的分散環境では通信遅延のために $C_i(x)$ の最小値がその時点での大域的な最小値である保証はない．そのため後述の評価では前節で求めた静的最適配置と比較することで最適性を検証しながら評価を行う．

ここで，平均応答時間 (MRT) を増加させる遅延を原因別に分類すると以下のようになる．

負荷遅延 複数のプロセスが処理装置で逐次的に処理される結果生じる応答時間の遅れのこと．正確には処理装置のプロセススケジューリング方法に依存するが，一般的には負荷に比例して大きくなるため，プロセスを分散させることで改善できる．ある処理装置でそのプロセスを単独で処理した結果 T 時間かかるプロセスを他の $n-1$ 個のプロセスを同時に処理すると，nT 時間かかると考えられる．処理装置の TSS 量子時間を 0 に近付けるとその処理装置のプロセスは並列に動作し，かつ応答時間が n 倍になると考えられる．

通信遅延 メッセージが移動することにより生じる遅延時間のことで，ある地点を出発し別の地点へ到達するまでに要する時間である．仮に，メッセージが x_1 から x_2 へ速度 v で移動したとすると，予想される通信遅延は $|x_2-x_1|/v$ である．第3章に述べた通信容量の仮定から，これはプロセスの移動にかかる時間についても同様である．

次に，負荷遅延と通信遅延の式を変形させて準最適配置を実現する式を決定する．はじめに，負荷遅延と通信遅延の比較的単純な結合式を評価する．以下の評価関数を用いてシミュレーションした結果を**図 4** に示す．

EQ 1 $c_1\sum d^2/v+c_2L$

EQ 2 $c_1\sum d^2/v^2+c_2L$

EQ 3 $c_1\sum d^2+c_2L$

EQ 4 $c_1\sum d/v+c_2L$

EQ 3 を除き同じ傾向を示した．EQ 1 と EQ 2 はほとんど一致して良好であった．しかし，一部で EQ 2 が優れていた．そのため EQ 2 が 4 者の中でもっとも最適配置に近い配置を与えることがわかった．また，図 3 の静的最適配置と比較しても遜色ない．このことから本論文では，統一コストを以下のように定めた．

統一コスト プロセス i に対する統一コスト $C_i(x)$ は，

$$C_i(x)=\frac{c_1}{v^2}\sum_j \varepsilon_{ij}d(x,x_j)^2+c_2L_i(x)$$

である．ここで，c_1, c_2 は定数，v は通信速度，ε_{ij} は通信頻度行列，d は距離関数，x_i はプロセス i の位置，$L_i(x)$ はプロセス i の認識する x 近傍の負荷である．また，このときプロセス i にとっての最適位置は，$C_i(x_i')=minC_i(x)$ となる位置 x_i' である．

上式における定数 c_1, c_2 を定めるため，これらの定数と MRT の関係を**図 5** に示す．図 5 は，$c_1=1$ として c_2 と v を変化させ，その時々の MRT をプロットしたものである．これにより $c_1=c_2=1$ で十分であることがわかる．したがって，以後の評価は $c_1=c_2=1$ で行う．

本アルゴリズムで閾値 T はプロセスやメッセージの移動のブレーキの役割を果たしている．また，本アルゴリズムの閾値 T に対する評価を**図 6** に示す．評価は $c_1=c_2=1$, $N=10$, $n=200$ で，初期配置 $r_{nf}=-1$ と $r_{nf}=1$ について行った．いずれの場合でも T はあまり大き過ぎなければ特に差はないといってよい．以後の評価は $T=1$ で行う．

6. 評　　価

本章では前章で求めた最適配置アルゴリズムの有効性を検証する．10 プロセス，200 イベントにおける通信速度と MRT の関係を**図 7** に示す．比較のために

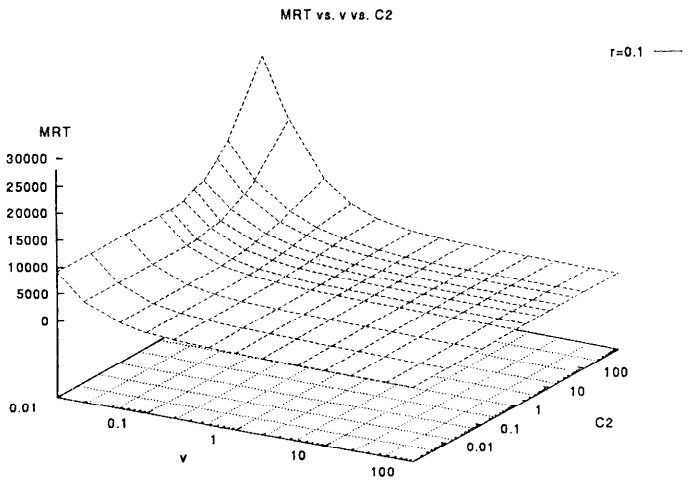


図 5 係数と通信速度に対する応答時間の関係
Fig. 5 MRT vs. v and c_2.

$v \geq 10$, $v=1$, $v \leq 0.1$ における静的最適配置を重ねて図示する．また，動的最適配置は初期値として $r_{nf}=\pm 1$ の場合を示す．この結果，得られた動的最適配置は，v の各領域の静的最適配置に若干劣るものの，ほとんどすべての領域でそれに近い応答時間を実現しており，通信速度に対して優れた適応性を持つことがわかる．

また，初期配置 $r_{nf}=\pm 1$ のグラフを比較すると，v で違いが顕著にあらわれた．これは一種の初期遷移問題と考えられる．イベント数 200 ではシステムの動的振舞いが安定するには十分であっても，初期遷移の影響を無視するには不十分である．初期遷移の影響と動的安定性を示すため**図 8** に様相の時間的変化を示す．図 8 は，$v=1.0$ と $v=10$ のそれぞれにおいて初期配置を $r_{nf}=1$ と $r_{nf}=-1$ とするときの r_{nf} の時間的推移を示したものである．プロセス数が少ないため ($N=10$)，r_{nf} は離散的な値をとるが，収束の過程は示すことができる．また，最終状態における平均的様相を**図 9** に示す．これは，横軸に場の座標をとり，縦軸にプロセス数をとったプロセスの平均分布を表す図である．v が大きくなるにつれて平たくなっていくのがわかる．

再び，図 7 に戻り，初期配置 $r_{nf}=-1$ の場合，$0.1<v<1$ で応答時間の極大箇所が存在する．これは

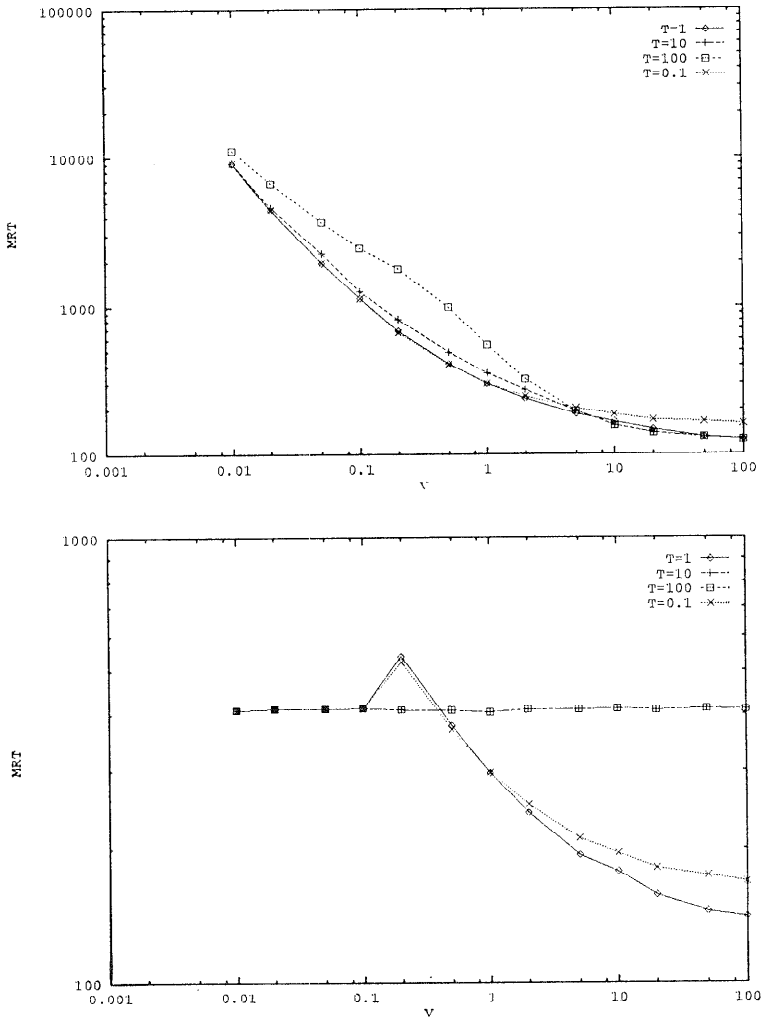


図 6 閾値 T の評価
Fig. 6 Evaluation of threshold T.

通信コストが大きいにも拘らず最初に負荷分散で押し出され，その分のコストが無視できないまま残ったものと考えられる．また，二つの初期配置のうち $r_{nf}=1$ となる集中配置を選んだ方が比較的短命なタスクに対しても有効であるといえる．なぜなら，$v \ll 1$ のとき $r_{nf}=-1$ が払うコストより $v \gg 1$ のとき $r_{nf}=1$ が

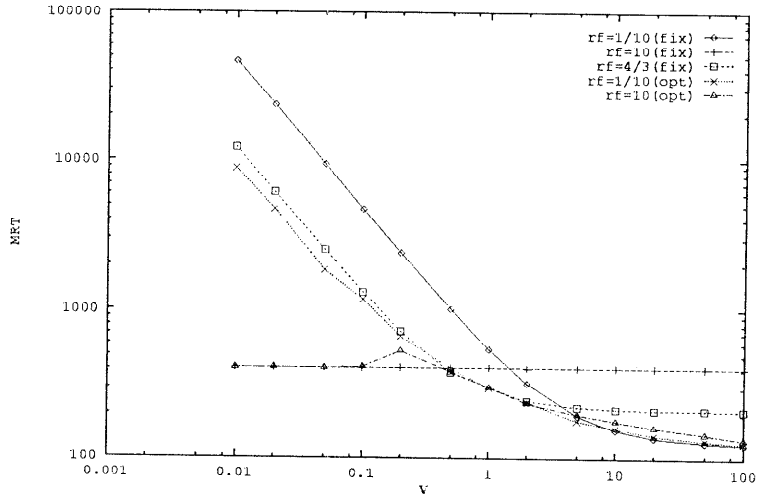


図 7 最適配置における通信速度に対する平均応答時間
Fig. 7 MRT vs. v in dynamic optimal allocation.

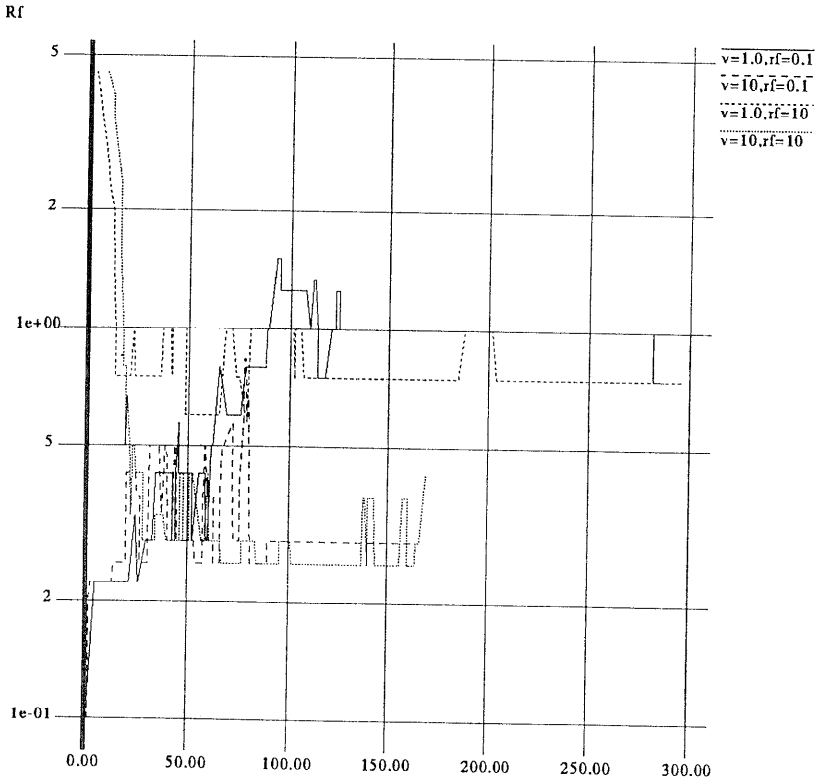


図 8 様相の時間的変化
Fig. 8 The transition of configurations.

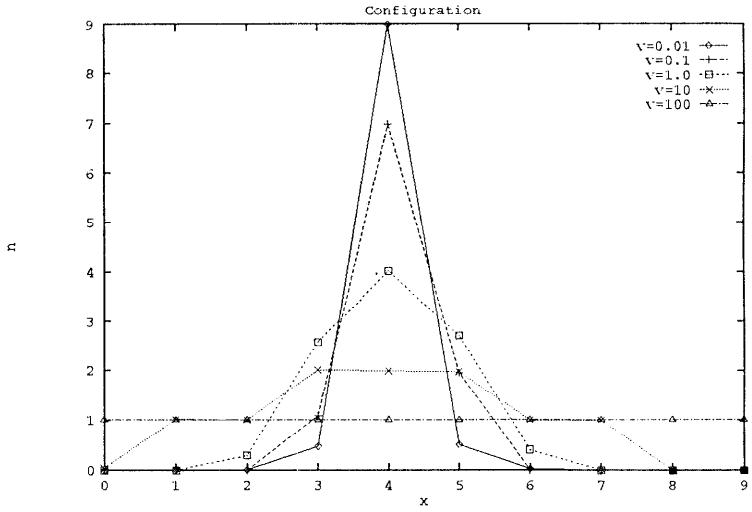


図 9 動的最適配置の最終的な様相
Fig. 9 Final configurations in dynamic optimal allocation.

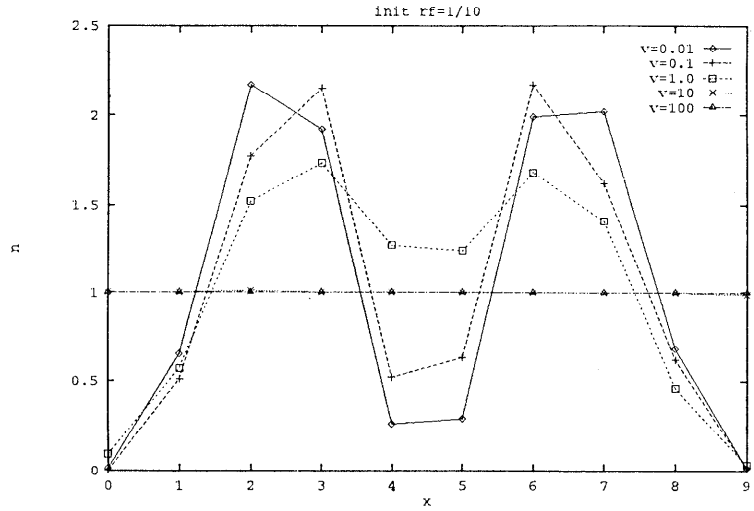


図 10 動的グルーピング
Fig. 10 Dynamic grouping.

払うコストの方がはるかに小さくて済むからである．また，後述のマルチタスクの場合でも有利となることが予想される．

次にグルーピングの効果を調べる．評価タスクは一つのプロセスを共有する二つのグループで行った．その通信頻度行列 ε は，

$$\varepsilon_{ij}=\begin{cases}1 & (i,j=1,\cdots,5 \ \textit{or} \ i,j=5,\cdots,10)\\ 0 & \textit{otherwise}\end{cases}$$

図 10 に 2 グループからなるタスクに本アルゴリズムを適用した結果を示す．図 10 は横軸に場の座標をとり縦軸にプロセス数をとったプロセスの分布図である．二つの山が観測され，一つのタスクが本アルゴリズムにより密接に関連する二つのグループに分けられたことが確認された．

7. 結　　論

本論文では，計算場における協調プロセスの動的最適配置アルゴリズムを提案し，その評価を行った．計算場は分散計算を効率よく行うために提案された分散計算のモデルである．計算場では，プロセス間にグ

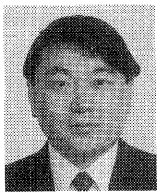
ルーピング効果を及ぼす引力と負荷分散効果を及ぼす斥力のバランスにより最適配置を実現する．本論文では同様の振舞いを一種の勾配法とその評価関数に帰属させ，シミュレーションによって解析した．

この結果以下のことが明らかとなった．第一に，本アルゴリズムには通信速度に関して広範囲な適応性が認められた．第二に，十分長命なタスクでは理想的な静的最適配置に近づいていくことが確認された．第三に，短命なタスクで生じる初期遷移問題は，初期配置を一か所に集中させる戦略を採用することで改善できた．最後に，複数のタスクにおいてはグルーピングの効果が確認された．以上のことから，本論文の動的最適配置手法は，論文中で規定されたような種々の制約はあるものの，地理的分散環境における本方式の有効性を示唆するには十分な結果が得られた．今後は実際の地理的分散環境においてタスクや実行環境の測定を行い，より現実的なモデルに基づく評価や実際のシステムにおける実装を行う必要がある．

参考文献

1) Chowdhury, S.: The Greedy Load Sharing Algorithm, *Journal of Parallel and Distributed Computing*, Vol. 9, pp. 93-99 (1990).
2) Eager, D. L., Lazowska, E. D. and Zahorjan, J.: Adaptive Load Sharing in Homogeneous Distributed Systems, *IEEE Trans. Softw. Eng.*, Vol. 12, No. 5, pp. 662-675 (1986).
3) Forrest, S. (ed.): *Emergent Computation*, North-Holland (1990).
4) Furuichi, M., Taki, K. and Ichiyoshi, N.: A Multi-Level Load Balancing Scheme for OR-Parallel Exhaustive Search Programs on the Multi-PSI, *Proceedings of the Second ACM SIGPLAN Symposium on Principle & Practice of Parallel Programming*, ACM SIGPLAN (Mar. 1990). SIGPLAN NOTICES, Vol. 25, No. 3.
5) Honda, K. and Tokoro, M.: An Object Calculus for Asynchronous Communication, *Proceedings of European Conferences of Object-Oriented Programming 1991*, LNCS No. 612, Springer-Verlag (July 1991).
6) Honda, K. and Tokoro, M.: On Asynchronous Communication Semantics, *ECOOP '91 Workshop on Object-Based Concurrent Computing*, LNCS No. 512, Springer-Verlag (July 1991).
7) Huberman, B. A. (ed.): *The Ecology of Computation*, North-Holland (1988).
8) Kurose, J. F. and Simha, R.: A Microeconomics Approach to Optimal Resource Allocation in Distributed Computer Systems, *IEEE Trans. Comput.*, Vol. 38, No. 5, pp. 705-717 (1989).
9) Lin, F. C. H. and Keller, R. M.: The Gradient Model Load Balancing Method, *IEEE Trans. Softw. Eng.*, Vol. SE-13, No. 1, pp. 32-38 (1987).
10) Lumer, E. and Huberman, B. A.: Dynamics of Resource Allocation in Distributed Systems, Technical Report, PARC, SSL-90-05 (1990).
11) Mirchandaney, R., Towsley, D. and Stankovic, J. A.: Adaptive Load Sharing in Heterogeneous Distributed Systems, *Journal of Parallel and Distributed Computing*, Vol. 9, pp. 331-346 (1990).
12) Tokoro, M.: Computational Field Model: Toward a New Computing Model/Methodology for Open Distributed Environment, *Proceedings of 2nd International Workshop on Future Trends in Distributed Computing Systems*, No. 1 (Sep. 1990).
13) Uehara, M. and Tokoro M.: An Adaptive Load Balancing Method in the Computational Field Model, *Proceedings of the Workshop on Object-Based Concurrent Programming*, April (1991). OOPS MESSENGER, Vol. 2, No. 2.
14) Weiser, M.: Some Computer Science Issues in Ubiquitous Computing, *Comm. ACM*, Vol. 36, No. 7, pp. 74-84 (1993).
15) Williams, I., Wolczko, M. and Hopkins, T.: Dynamic Grouping in an Object-Oriented Virtual Memory Hierarchy, *Proceedings of European Conferences of Object-Oriented Programming 1987*, pp. 87-96 (Nov. 1987).
16) 清水譲太郎，前川　守，所真理雄：分散オペレーティングシステム—UNIX の次にくるもの—，共立出版 (1991).
17) 寺岡文男：VIP：ホスト移動透過性を提供するプロトコル，*Comput. Softw.*, Vol. 10, No. 4, pp. 22-38 (1993).
18) 半田　豊，上原　稔，所真理雄：分散計算におけるオブジェクトの最適配置，*7th Conf. Proc. Japan Soc. Softw. Sc. Tech*, pp. 185-188 (Dec. 1990).
19) 上原　稔，所真理雄：接触通信モデル：開放型分散システムのための計算モデル，*Proceedings of Workshop on Object-Oriented Computing 1990*, No. 1 (Mar. 1990).

(平成5年10月 7 日受付)
(平成6年11月17日採録)



上原　稔（正会員）

1964年生．1989年慶應義塾大学大学院修士課程（電気工学）修了．1993年慶応義塾大学大学院博士課程（計算機科学専攻）単位取得退学．同年東洋大学工学部情報工学科講師．分散計算，プログラミング言語，ユーザインタフェースなどに興味を持つ．ACM 会員．

所　真理雄（正会員）

1947年生．1970年慶應義塾大学工学部電気工学科卒業．1972年同大学大学院修士課程（電気工学専攻）修了．1975年博士課程（電気工学専攻）修了．工学博士．同大学電気工学科助手，専任講師，助教授を経て現在教授．1988年4月よりソニーコンピュータサイエンス研究所副所長を兼務．計算モデル，プログラミング言語，分散・開放型システム，人工知能などに興味を持っている．著書に「計算システム入門」（岩波ソフトウェア科学1）などがある．電気通信学会，ACM, IEEE 各会員．