マルチメディア通信と分散処理 71−11
（１９９５．７．１３）

通信量を削減する動的負荷分散

松井健一，松田亘弘，藤井章博，根元義章
東北大学大学院 情報科学研究科
〒980-77 宮城県 仙台市 青葉区 荒巻字青葉
E-mail:matsui@nemoto.ecei.tohoku.ac.jp

あらまし　複数の計算機をネットワークで接続した分散システムにおいて，計算機間の負荷不均衡を改善する負荷分散の概念は，応答時間の向上や，のみならず信頼性の向上につながるなど多くの面で有用である．また，計算機の能力が異なる場合，負荷分散は計算機資源の有効利用を図ることが出来る．

本稿では，分散システムをグループ化し，負荷探索対象計算機を限定して，負荷探索に要する通信量を削減する動的負荷分散方式を提案する．同時に，非均一分散システムでの提案方式の性能を，シミュレーションにより評価する．実験の結果から，提案手法は分散システムの負荷が高い場合でも，平均プロセス応答時間は長くなるものの，メッセージ量を低く抑えることが確認された．

Reducing a Communication Cost on a Dynamic Load Sharing in Distributed Systems

Ken-ichi MATSUI, Takehiro MATSUDA, Akihiro FUJII, and Yoshiaki NEMOTO
Graduate School of Information Sciences, Tohoku University
Aoba-ku, Sendai-shi, Miyagi Pref., 980-77 JAPAN
E-mail:matsui@nemoto.ecei.tohoku.ac.jp

Abstract　Load sharing in a distributed system provides benefits in various aspects, such as improvements to responcetime and system reliability. When several processors in a distributed system have different abilities, performance of applications can be improved by load sharing as well.

In this report, we propose an adaptive load sharing method which reduces communication costs. In this method, nodes are devided into several groups. Then in each group, each node communicate one another, however they do not communicate with other group, that the number of communications are reduced.

Extensive simulations are carried out to estimate the performance of the proposed method in heterogenious distributed computation. The results indicate that the proposed method, although increase an average response time than that of previous method, reduces communication cost in case the load of distributed system is high.

1 はじめに

複数の計算機がネットワークによって結合されたシステムを分散システムと呼ぶ．分散システムでは，一部の計算機のみに負荷が集中し，他の計算機が遊休状態になるといった，計算機間の負荷が不均衡な状態が生じる．ここで，計算機の状態を把握し，最適な負荷の分散・配置を行い，分散システム全体として応答時間の短縮や，資源の利用率の向上と言った性能向上を行う概念を負荷分散という．

負荷分散アルゴリズムは大きく，静的負荷分散と動的負荷分散の2通りに分けられる．静的負荷分散は，分散システムの状態 (プロセッサの負荷，プロセスの情報) をあらかじめ統計処理等によって評価しておき，この情報に基づいて負荷の分散を行う．静的負荷分散は数学的に解析が可能である反面，性能面では負荷の動的変動が考慮されていないため，動的負荷分散より劣ることが多い．これに対して動的負荷分散は，動作中の分散システムの状態を用いて，システムが動作中に負荷を移送する．これにより，分散システムの変動に対応した負荷の分散が行える．これは静的負荷分散に比べても大きな効果を持つ．

しかし動的負荷分散を行うには，システムの変動を常に把握する必要があり，そのためのメッセージ交換量が大量になる．これは性能に影響するおそれがある．実際，基本的な動的負荷分散アルゴリズムである最短アルゴリズム[1](shortest algorithm) では，メッセージ交換量は大量になり得る．

最短アルゴリズムは次のように動作する．負荷の軽い計算機が分散システム内の全計算機の負荷を検索し，負荷の大きい計算機から順にプロセスを受け取る．ここで，分散システム全体の負荷状態を把握しなければならず，メッセージ交換量が多くなるという問題点がある．反面，正しい負荷情報に基づくプロセス移送を行うため，適切な移送先を見つけることが可能である．

最短アルゴリズムは全ての計算機を検索するが，負荷状態を最短した計算機の内，実際に移送対象となる計算機はごく少数であり，無駄なメッセージ交換が行われている．そこで，分散システム内の全計算機とメッセージを交換するのではなく，ある特定のグループとメッセージ交換を行い，メッセージ量を減らす方法が考えられる．特に計算機の能力が非均一である分散システムにおいては，システムを能力の低い計算機と高い計算機に分割することで，能力の低い計算機への負荷集中を防げる．

本論文では，分散システムをグループ化することによって，最短アルゴリズムにおいてもメッセージ交換量を減らす改良アルゴリズムを提案する．提案する改良アルゴリズムは，最短アルゴリズムの負荷検索部分を改良したものである．すなわち分散システム中の計算機をいくつかのグループに分割し，グループ内で負荷情報を交換する．これにより分散システム内の全計算機とメッセージを交換することが防止され，結果としてプロセス移送に要する通信負荷を少なくする．しかし提案アルゴリズムは，負荷情報の交換相手を限定するため、情報が不正確になることも予想される。

提案方式によるメッセージ交換量の削減，およびグループ化により予想される情報量の減少に伴う応答時間の増加傾向を把握するためにシミュレーションを行った．シミュレーションの結果から，提案方式は分散システムの負荷が高い状態でも，応答時間は比較的大きいものの，メッセージ交換量は低く抑えられることが分かった．

以下，2章では，仮定する分散システムのモデルを定義する．3章では，提案する改良アルゴリズムを述べる．4章では，改良アルゴリズムをシミュレーションにより評価した．5章ではまとめを述べる．

2 分散システムモデル

計算機と分散システムを以下のようにモデル化する．

1. **計算機**
   - プロセッサ (Processor)
   - プロセス分散機構 (Process Distributor)
   - プロセス待機キュー (Process Queue)

   生起したプロセス，あるいは移送されたプロセスは，一旦プロセス待機キューに入り，プロセス分散機構によって，自プロセッサで処理されるか，他の計算機へ移送されるか決定される．
2. **計算機の機能とプロセスの生起**
   - 各計算機は単一のプロセッサを所有する．
   - プロセッサの性能は必ずしも同一ではない．
   - プロセス分散処理に必要な時間は，プロセッサの処理時間と比べて十分速い．
   - 情報処理の基本的な単位はプロセスとする．
   - プロセスは計算機においてプロセッサを使用する．

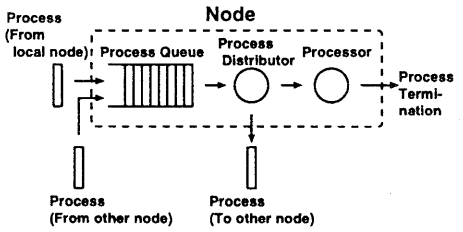

図 1: 計算機のモデル

- プロセスは独立に，各計算機で生起する．

3. 分散システム
   - 全ての計算機間は高速のネットワークによって，相互に隣接するように接続されている．
   - 任意の計算機間でプロセスを移送できる．
   - 計算機間を行き来するメッセージは遅延無く直ちに到着し，プロセス分散機構により処理され，プロセス移送・処理自体には影響しないものとする．

3 動的負荷分散アルゴリズム

動的負荷分散アルゴリズムの一つである最短アルゴリズム[1] (shortest algorithm) を基本とした，負荷探索メッセージ数を削減する改良アルゴリズムを提案する．改良アルゴリズムは，分散システム上の計算機を複数のグループに分割する．そして，負荷探索メッセージは，このグループ内で送信する．

3.1 記号の説明

アルゴリズムの記述に，以下の記号を用いる．

N	:計算機数
h_i	:計算機．ここでiは計算機のIDで最大$0 \leq i \leq N$で変化する．
l_i	:計算機h_i上のプロセス数
LH_i	:計算機h_iでのプロセス数上限しきい値
LR_i	:計算機h_iでのプロセス数下限しきい値
m, m'	:1回の負荷分散時の総移送プロセス数，1計算機への移送プロセス数
S_k	:分散システムを分割した際の1領域
N_k, s_k^i	:分割領域S_k内の計算機数，計算機ID
N_{kL*}	:分割領域S_k中でプロセス数しきい値を満たす計算機の数
T_i	:計算機h_iでのプロセス移送回数しきい値

3.2 改良アルゴリズム

1. (負荷の探索)
   $l_i < LR_i$となった時，$h_i(1 \leq i \leq N)$は$l_j(s_k^1 \leq j \leq s_k^{N_k}, j \neq i)$を検索する．**2**へ．
2. (プロセスの移送)
   $l_j > LH_j$を満たす$h_j \in S_k$の個数をN_{kLR}とした時，1のh_iはN_{kLR}の値に従ってプロセスを次のように移送する．
   $m \leq N_{kLR}$の場合 → **2 − a**へ
   $0 \leq N_{kLR} < m$の場合 → **2 − b**へ
   $0 = N_{kLR}$の場合 → **2 − c**へ

2-a. $l_j > LH_j$を満たす$h_j \in S_k$のうち，l_jが最大の値をもつh_jから順に，h_iへプロセスをm'個移送する．$m = \sum m'$である．**3**へ．

2-b. $l_j > LH_j$を満たす$h_j \in S_k$のうち，l_jが最大の値をもつh_jから順に，h_iへプロセスをm'個移送する．ここで$m = \sum m'$である．$m - N_{kLR} \times m'$個のプロセスは移送しない．**3**へ．

2-c. h_iはプロセスを移送しない．**3**へ．

3. (移送回数の制限) 但し，移送対象となるプロセスの既移送回数が，移送回数しきい値T_iを越えている場合は移送しない．**1**へ戻る．

また，グループ間の計算機能力に明らかな差がある場合は，負荷探索メッセージを一方のグループから他方のグループへ一方的に送信することもある．

4 シミュレーション

提案した動的負荷分散方式を評価するため，シミュレーションを行った．以下の結果はUNIX上にC言語を用いて実現したシミュレータにより行う．

4.1 シミュレーション条件

計算機数

分散システムに含まれる計算機数Nは20[台]とした．

プロセスの平均処理率

分散システムは非均一とし，計算機速度は異なるとした．具体的には1プロセスの各プロセッサにおける処理は，平均処理率μ_iがそれぞれ1，

2[1/s] のポアソン分布に従うと設定し，各々を基準計算機，高速計算機とみなす．基準計算機の数，高速計算機の数をともに10[台] とした．

領域分割

分散システム Sは領域S_1とS_2に2分割した．S_1は基準計算機が$N_1 = 10$[台]，S_2は高速計算機が$N_2 = 10$[台] からなる．S_1とS_2はそれぞれのグループ内で負荷を監視する他に，S_1はS_2を一方的に負荷監視・移送対象とする．

しきい値と移送プロセス数

計算機が負荷分散行う判断基準とする上限しきい値LH_iおよび下限しきい値LR_iは，S内の計算機全てを通して等しくそれぞれ3, 1[個] とする．また1回の負荷分散時の総移送プロセス数mを3[個]，その際の1計算機からの移送プロセス数m'を1[個] とする．

プロセスの平均発生率 プロセスの平均発生率$\lambda_i[1/s]$は各ノードにおいて指数分布に従い，また全て等しく，$5 \times 10^{-2} \sim 95 \times 10^{-2}[1/s]$まで変化させた．

シミュレーション回数

シミュレーション上で進む時間はプロセスの平均到着間隔の約10^4倍に相当する．回数の不足による誤差は少ないものと考えられる．

プロセス移送時のオーバーヘッド

プロセスの移送にはある時間が必要である．この移送に伴う遅延時間は，プロセスの応答時間に含まれるため，移送を繰り返すと応答時間が長くなる．移送されなければ生じなかった応答時間の増加を，プロセス移送時のオーバーヘッドとする．本シミュレーションでは，プロセスは逐次処理によって処理されるため，経過時間を元にしたオーバーヘッドは設定できない．そこで単純に各ノードの平均処理時間の2倍を，1回の移送に際するオーバーヘッドとした．

以上のシミュレーション条件を表1にまとめた．

4.2 性能指標

ここでは評価に際する指標値について述べる．

プロセスの平均応答時間 プロセスが生成してから終了するまでの時間をプロセス応答時間w_iと定義する．全生成プロセス数をnとすると平均応答時間は$W = \frac{\sum_{i=1}^{n} w_i}{N}$と定義される．$W$が小さい程，負荷分散の効果が高いと言える．

表 1: シミュレーション条件

パラメータ	設定値
計算機数	$N = 20$[台]
計算機速度	基準 $\mu_1 = 1[1/s]$
	高速 $\mu_2 = 2[1/s]$
計算機構成	基準 $N_1 = 10$
	高速 $N_2 = 10$
上限しきい値 LH	3[個]
下限しきい値 LR	1[個]
移送回数しきい値 T	10[回]
総移送プロセス個数 m	3[個]
移送プロセス個数 m'	1[個]
プロセス平均発生率	$0.05 \leq \lambda \leq 0.95[1/s]$
シミュレーション時間	$\lambda \times 10^4$
オーバーヘッド	$\mu_1 \times 2$

プロセスの応答時間の変動係数 wの変動を調べるために，変動係数cを導入する．$c = V/W^2$で表される．ここでVはプロセスの応答時間の分散で$V = \sum_{i=1}^{n}(w_i - W)^2$と表される．プロセスの応答時間の変動は小さい方が望ましい．小さいと常に一定のプロセス処理時間が期待できるからである

負荷探索メッセージ量 ノードが負荷を探索する際に，交換するメッセージ量を観測する．メッセージ量が大量になると，ネットワークに対する負担が大きくなる．

4.3 シミュレーション結果

本節では提案アルゴリズムのシミュレーション結果に対して，4.2節の観点から，**1.** プロセス平均応答時間，**2.** プロセス応答時間の変動係数，**3.** 負荷探索メッセージ交換量，および **4.** 平均応答時間とメッセージ量の関係について考察する．

1. プロセス平均応答時間

図2は，システムへ加える負荷を変化させた場合のプロセス平均応答時間を示したものである．これを見ると，提案した改良アルゴリズムは負荷が小さい(< 0.5)時点では，他のアルゴリズムと変わらない応答時間を

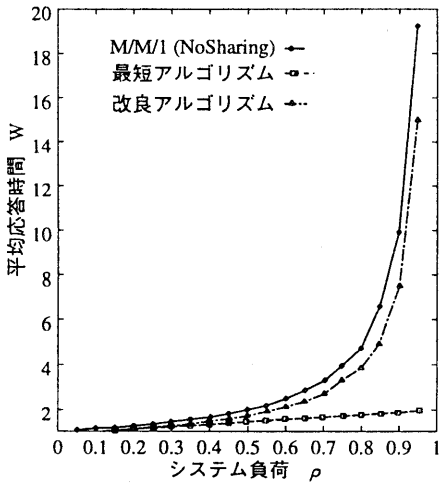

図 2: プロセス平均応答時間

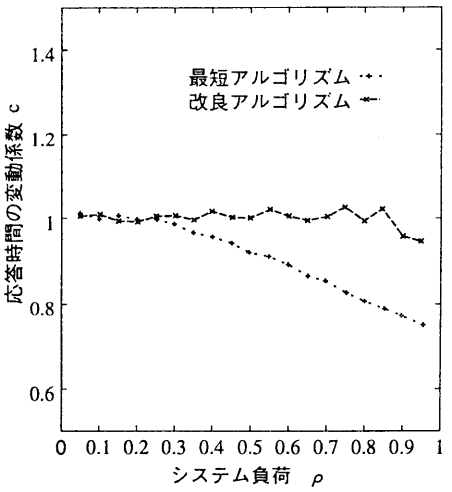

図 3: プロセス応答時間の変動係数

示す．しかし負荷が大きくなると，応答時間は負荷分散を行わない場合に近くなる．これは負荷分散が不完全に行われていることを示している．領域を分割したことによって，プロセス分散機構が，負荷移送に適するノードを発見できなくなったことが原因である．応答時間の面で，提案アルゴリズムは負荷が低い場合は最短アルゴリズムと同等なものの，高くなると比較的大きくなる．

2. プロセス応答時間の変動係数

図3は，システム負荷を変化させた場合のプロセス応答時間の変動係数である．これを見ると，改良アルゴリズムの応答時間変動係数はほぼ1を保っており，応答時間のばらつきが一定である．しかし最短アルゴリズムの変動係数は1を下回っており，改良アルゴリズムの応答時間と比べてよりばらつきが少ないといえる．この原因も同様に，改良アルゴリズムが最短アルゴリズムと比べて，負荷情報を正確に把握できなかったためである．

3. 負荷探索メッセージ交換量

図4はシステムへ加える負荷を変化させた場合のメッセージ交換量を示したものである．これを見ると，負荷が増加して行くに従って，最短アルゴリズムはメッセージ交換量を増加させて行く．一方改良アルゴリズムのメッセージ量は，全般に最短アルゴリズムと比べて少なく抑えられていることが分かる．また，改良アルゴリズムは負荷が0.5付近で，最短アルゴリズムは負荷が0.8付近でメッセージ回数が減少傾向を示す．これは，今回のシミュレーションで受け手主導方式を採用したためである．すなわち図4の各々の頂点付近に対応する負荷の値で，分散システム中に受け手となり得るプロセス数の少ないノードの数が減少しはじめたことを示す．つまり改良アルゴリズムによるグループ化で，最短アルゴリズムと比べて，分散システムの負荷飽和が比較的早く始まることを示している．

4. 平均応答時間とメッセージ量の関係

図5は，分散システムに加えられる負荷が増加していく時の，プロセスの平均応答時間と総メッセージ交換量間の関係変化を調べたものである．メッセージ交換量の観点からこの図を見ると，改良アルゴリズムにおけるメッセージ交換量は最短アルゴリズムに比べて低く抑えられてることがわかる．一方平均応答時間の観点から見ると，最短アルゴリズムの方が短い応答時間を示している．このことから提案アルゴリズムはメッセージ交換量を減少させるが，しかしプロセス応答時間とメッセージ交換量の減少は両立しないことがわかる．

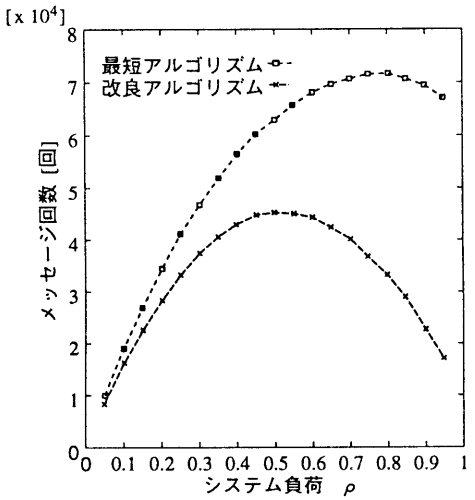

図 4: 負荷探索メッセージ交換量

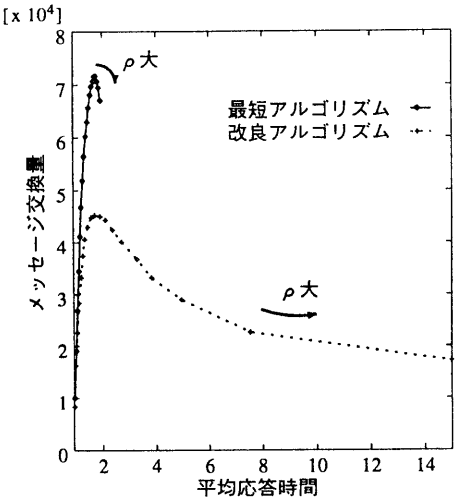

図 5: 平均応答時間 vs メッセージ交換量

5 まとめ

本稿では，動的負荷分散アルゴリズムの一つである最短アルゴリズムを改良し，メッセージ交換量を削減する動的負荷分散アルゴリズムを提案した．具体的には，分散システム中の計算機をグループに分割し，グループ内でメッセージ交換を行うよう制限した．提案アルゴリズムは最短アルゴリズムよりもメッセージ交換量を減少することが，シミュレーションにより確認できた．しかしシミュレーションの結果から，応答時間は最短アルゴリズムに比べて増加することもまた確認された．原因としてはシステム領域を分割した事により，システムの負荷情報が不正確になったためと考えられる．

本手法の適用が有効であるのは，メッセージ交換量が多くなる規模の大きい分散システムなどであると考える．しかし適用までには，グループ化にともなう情報量の低下という問題点を解決しなければならない．ゆえに今後の課題として，負荷情報の正確さを保つ手法の開発があげられる．例としては，分散システムのグループ化を動的に行う手法の開発，あるいは負荷情報のデータベース化とこれの有効な利用手法の開発等が考えられる．

参考文献

[1] Derek L. Eager, Edward D. Lazowska, and John Zahorjan: Adaptive Load Sharing in Homogeneous Distributed Systems, IEEE Trans. Software Eng., 12, 5, pp.662-675 (May 1986)

[2] 前川 守: オペレーティングシステム, 岩波書店, pp.353-377 (1988)

[3] 朴 圭成, 芦原 評, 清水 謙多郎, 前川 守: 分散オペレーティングシステムにおけるプロセス移送の方式, 情処論, Vol.31, No.7, pp.1080-1090 (July 1990)

[4] 芦原 評: 動的負荷分散におけるメッセージ交換の削減, 情報処理学会研究会報告 92-オペレーティング・システム-55-1, 8 (June 1992)

[5] Sol M.: Task Allocation for Maximizing Reliability of Distributed Computer Systems, IEEE Trans. on Computers, Vol. 41, No.9, pp.1156-1168 (September 1992)

[6] 李 相吉吉, 計 宇生, 松方 純, 浅野 正一朗: 負荷ベクトルを用いた負荷分散方式の検討, 信学論, D-1, Vol.J76-D-1, No.3, pp.118-129 (March 1993)

[7] Songnian Zhou,Xiaohu Zheng,Jingwen Wang AND Pierre Delisle: Utopia: a Load Sharing Facility for Large, Heterogeneous Distributed Computer Systems, Software-Practice and Experience, Vol.23(12), pp.1305-1336 (Dec 1993)