システムソフトウェアと
オペレーティング・システム 72－3
(1996. 5. 15)

多重階層化負荷管理方式による超並列計算機向き動的負荷分散

布目 淳　平田 博章　新實 治男　柴山 潔

京都工芸繊維大学 工芸学部 電子情報工学科

京都市左京区松ヶ崎御所海道町

あらまし　本稿では超並列計算機向きの負荷拡散速度の向上を目的とした動的負荷分散方式を提案する．本方式では，負荷情報を“プロセッサ領域”単位で階層的に管理する．各プロセッサ領域には１つの管理プロセッサを論理的に設定する．これは新たに負荷を生成したプロセッサからの相対的な距離によって決定する．この方式は，(i) プロセッサ領域を定義して，これを負荷管理の単位とすることで特定のネットワークトポロジに対する依存を回避できる，(ii) 負荷管理機構がシステム全体に均一に分散するため，特定の管理機構にアクセスが集中するボトルネックが生じる可能性が低い，(iii) 負荷を生成したプロセッサからある程度離れた位置の負荷量を管理できるため，隣接プロセッサのみに負荷を割り付ける方式と比べて負荷拡散速度を向上できる，などの特長を有する．

キーワード　動的負荷分散, 超並列計算機向き OS

Dynamic Load Balancing Scheme based on the Hierarchical Management of Overlapped Processor Region for Massively Parallel Computers

Atsushi NUNOME,Hiroaki HIRATA,Haruo NIIMI,Kiyoshi SHIBAYAMA

Department of Electronics and Information Science, Faculty of Engineering and Design, Kyoto Institute of Technology

Gosyokaido-Cho,Matsugasaki,Sakyo-ku,Kyoto

Abstract　We propose a dynamic load balancing scheme to improve load spreading speed on massively parallel computers. In this scheme, processor elements (PE's) to be process migration targets are logically grouped into “processor regions”, and some processor regions are also included into a larger processor region at the upper level in the processor region hierarchy. This hierarchy is (re)constructed depending on the position of the PE which initiates a process migration. One of PE's in a processor region at each layer manages load status of all PE's in that region. Our scheme can be adopted independent from network topologies of machines, and can improve the load spreading speed with low costs of our load balancing control.

key words　dynamic load balancing, massively parallel computers' OS

1 はじめに

超並列計算機で動的に負荷を分散する際には，負荷管理のオーバヘッドを軽減するために負荷情報を局所的に管理し，その局所的な負荷情報を元にした大域的負荷分散を行う必要がある．ところが局所的な負荷情報のみを用いて負荷分散制御を行うと，負荷がシステム全体に拡散するまでの遅延時間が大きくなり，負荷の分布する範囲がなかなか広がらない．これは，超並列計算機のように大規模なプロセッサ構成を持つ計算機ではほとんど負荷分散されていないことに等しく，動的負荷分散制御の効果がほとんど現れない可能性がある．

本稿では，超並列計算機において負荷を速やかに拡散する動的負荷分散方式を提案する．負荷拡散遅延を減らすためには広範囲の負荷情報を収集することが必要になる．本方式では，比較的小さいオーバヘッドで負荷情報を管理するために，プロセッサ領域を負荷管理の単位として，プロセスの割付け時のみ階層的に負荷情報を収集するようにした．

2 従来の方式

スタック分割動的負荷分散方式 (Stack Splitting Dynamic Load Balancing; STB 方式)[1] は，負荷の分割と供給を行う要素プロセッサ (PE) を特定せず，全ての PE がそれらの処理を行う．アイドル状態となった PE は，負荷の供給を受けるまで，ある戦略をもとにして他の PE に順番に負荷の供給を要求する．

STB 方式では，最終的に自分を除く全ての PE に対して，負荷の要求を行う可能性がある．これは，システム内の全ての PE を負荷割付けの対象としており，超並列計算機上では，遠距離 PE との通信距離や通信回数の点でオーバヘッドが大きく，性能上の問題になりやすい．

動的負荷浸透方式 (Dynamic Local Load Spreading Method;LLS 方式)[2] は，各 PE が隣接 PE に対する自分の相対的な負荷量を計算し，自分が近傍内で比較的負荷が重いと判断した場合にのみ隣接 PE に負荷を割り付ける方式である．割り付ける負荷の量は，隣接 PE の負荷量がほぼ均等になるように決定する．LLS 方式は，全 PE が自分を中心とする隣接 PE に対して同様の動作を行うことで，システム全体の負荷の均等化を図るものである．

LLS 方式では，負荷の拡散が隣接 PE に限定されるため，超並列計算機のようにシステム規模が大きい場合に，負荷の拡散遅延が問題となる．また，LLS 方式は負荷情報の報告が短い間隔で行われると，その都度，隣接 PE の負荷量を均等化しようとして，負荷の連続的な移送が起こりやすい．

LLS 方式を改良した世代別動的負荷浸透方式 (Dynamic Local Load Spreading Method - Generation;LLS-G 方式)[3] では，負荷情報の報告と分散を世代別に行っている．負荷情報の報告を各世代に関して一度しか行わないことで，均等化のタイミングを適当な間隔に保ち，負荷の連続的な移送を防止している．しかし，LLS-G 方式も LLS 方式と同じく，負荷の割付け先が隣接 PE に限られている．このため，負荷の拡散のしやすさを調整できるものの，やはり負荷の拡散遅延が問題となる．

以上に述べたように，並列計算機の負荷分散方式として提案されているこれらの方式をそのまま超並列計算機に適用する際には，負荷管理のコストや負荷拡散遅延などの問題が生じる．特に，負荷管理の局所性と負荷拡散速度とは互いに相反する要件であり，最適なトレードオフを見い出す必要がある．

3 多重階層化負荷管理方式

本方式では，負荷情報の管理単位として “プロセッサ領域” を，また負荷量として “PE のプロセスキュー内の実行可能状態のプロセス数” をそれぞれ用いる．ここで “プロセッサ領域” とは，領域管理 PE を中心として一定の範囲内にある PE を論理的にグループ化したものである．

3.1 階層化プロセッサ領域

本方式では，3.2 節で述べるように，各 PE は隣接する PE の負荷情報を常に保持している．すべての PE は，負荷分散制御の必要に応じて管理 PE となり，これが 1 つのプロセッサ領域の負荷情報の管理を行う．最小のプロセッサ領域は管理

PE とそれに隣接する PE とから構成される．

最小のプロセッサ領域では隣接する PE の負荷情報しか得られないため，幾つかのプロセッサ領域をまとめて，より大きなプロセッサ領域として負荷を管理する．このようにして，階層的にプロセッサ領域を構成する．プロセッサ領域をどのように構成するかで，ある PE を管理 PE としたときのシステム内の PE を次のように分類できる．

非領域 PE プロセッサ領域に属さない PE．負荷情報はプロセッサ領域単位で管理されるため，この PE からは負荷情報を得ることができない．つまり，負荷分散時に管理 PE に対して負荷情報を通知しない．

単一領域 PE 唯一の管理 PE に対して負荷情報を通知する PE.

重複領域 PE 複数の管理 PE に対して負荷情報を通知する PE.

重複領域 PE が多いということは，複数の管理 PE が同一の PE から同一の負荷情報を収集するため，1 つ 1 つの管理 PE の重要性が低いことを意味する．ある管理 PE と，隣接する領域の管理 PE との距離を十分に取ることで，重複領域 PE の数を減らすことができる．このようにして重要性が低い管理 PE を除くことにより，非領域 PE の増加という条件下で管理 PE の数を減らすことができる．

プロセッサ領域の構成例として，2 次元メッシュ結合網への適用方法を述べる．

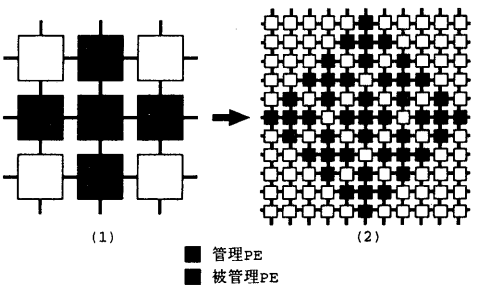

図 1: メッシュ結合の計算機での領域の拡張例

2 次元メッシュ結合網では，図 1(1) のように，1 つの管理 PE と周囲の 4 つの PE から最小プロセッサ領域が形成される．さらに，広範囲の負荷情報を収集するために周辺の複数の最小プロセッサ領域をまとめて上位階層のプロセッサ領域を構成する．周辺の最小プロセッサ領域をどのように選ぶかは，中心となる最小プロセッサ領域の管理 PE がそのプロセッサ領域周辺のどの PE を管理 PE とみなすかによって決まる．

図 1(2) は，中心となる管理 PE から距離 4 にある PE のうちの 8 つを周辺の最小プロセッサ領域を構成する管理 PE とみなしたときのものである．

上位階層のプロセッサ領域を決定する際，管理 PE 間の距離が十分取られていないと重複領域 PE が生じる．逆に，管理 PE 間の距離が離れ過ぎると，周辺のプロセッサ領域との間に非領域 PE が生じる．中心となる管理 PE からは非領域 PE の負荷情報が得られないため，負荷を直接割り付けることはできない．この点については，3.3 節で議論する．

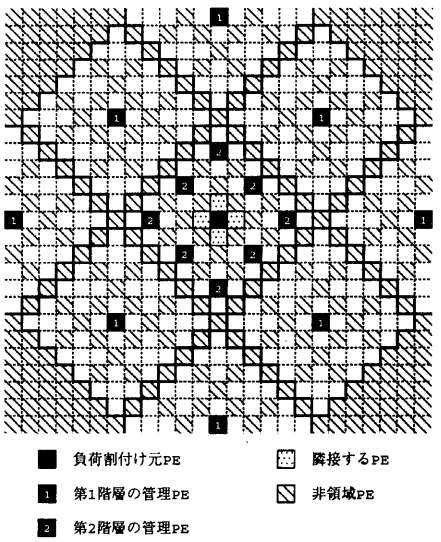

図 2: メッシュ結合の計算機への適用例 (階層数 3)

図 1(2) のように構成された複数の最小プロセッサ領域について，図 2に示すように，各管理 PE を階層的に管理する．この図は，中心に位置する PE がプロセスの割付け元 (これを第 0 層管理 PE とする) となった場合に，どのようにプロセッサ領域が階層化されるかを示したものである．第 0

層管理 PE は，距離が 12 の箇所に位置する 8 個の管理 PE(第 1 層管理 PE) のアドレスを保持している．また，第 0 層管理 PE は第 1 層管理 PE も兼ねるので，距離が 4 の箇所に位置する 8 個の第 2 層管理 PE のアドレスも保持している．第 0 層管理 PE は，周辺の 8 個の第 1 層プロセッサ領域の内部にある第 2 層管理 PE のアドレスは保持しない．

3.2 管理 PE の決定方法

管理 PE が下位階層にあるプロセッサ領域の管理 PE を管理することで，多数の PE を特定の PE が集中管理する必要がなくなり，負荷管理の分散化が実現できる．

本方式では，プロセッサ領域はプロセス生成時にのみ利用される．この領域は論理的に階層化されており，新たにプロセスを生成した PE を根 (第 0 層管理 PE) とした木構造を構成する．プロセッサ領域の階層を構成する方法は以下の通りである．

- 各 PE は隣接 PE と共に最小領域を形成し，これを最下層とする．
- 各 PE は自分を第 n 層の管理 PE とした時の，第 $(n+1)$ 層の管理 PE の位置を保持する．
- 領域の階層は，最上位層である第 0 層から第 $(N-1)$ 層までの N 層構成である．ここで，階層数 N はシステム規模に応じて静的に決定されるパラメータである．

ある PE における各階層の管理 PE の位置は，領域の形状やネットワークトポロジに依存する．ただし，これはあらかじめ静的に決定しておくものであり，実行時に動的に変化することはない．

管理 PE の決定は，次のようにする．すなわち，図 3のように，d_n を第 n 層に位置する管理 PE 間の最短距離，r_n を第 n 層の管理 PE が管理する領域の半径，S を領域間に設ける余地としたとき，以下の式にしたがって値を決める．なお，$S(0 \leq S \leq 2r_{N-1})$ はネットワークトポロジによって決定されるパラメータであり，領域同士が重ならない最小の値にするのが望ましい．

$$d_n = \begin{cases} 0 & (n=0) \\ 2r_n + 1 + S & (0 < n \leq N-1) \end{cases}$$

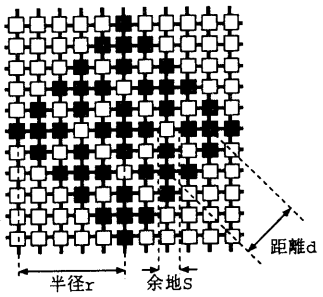

図 3: 管理 PE 決定時のパラメータ

$$r_n = \begin{cases} 1 & (n=N-1) \\ 3r_{n+1} + 1 + S & (0 \leq n < N-1) \end{cases}$$

上式の条件だけでは，第 n 層管理 PE から d_n の距離に位置する PE が全て管理 PE となり，非領域 PE が減る代わりに重複領域 PE が多数生じる．重複領域 PE が多くなると無駄な負荷収集トラフィックが発生するため，これを回避するために，管理 PE を適度に間引く必要がある．具体的には，同階層で隣接する領域の管理 PE 同士が，全て d_n の距離を持つように管理 PE を限定する．

このように各階層の管理 PE からの距離によって下位階層の管理 PE を決定することにより，メッシュ，トーラス，リングなどの静的網に属する各ネットワークトポロジに対しても本方式の適用が可能になる．ただし，静的網においても PE 間にバイパス経路が存在する場合には適用が難しい．これは PE 間の距離によっては，1 つの PE が異なった複数の階層の管理 PE として設定される可能性があるためである．したがって，このような場合には，管理 PE の決定において上式をそのまま使用するのではなく，バイパス経路の存在形態に応じて再調整を行う必要がある．

3.3 プロセスの移送

負荷が一定量以下の状態になった PE は，隣接する PE の内，最も負荷の重い PE からプロセスを引き取る．引き取るプロセスの量は，割付け元の負荷量の 1/2 を限度にする．これは，プロセスの引き取り過ぎによる連続的な負荷拡散動作の発生を抑制するためである．

この方式だけでは拡散速度が遅く，分散範囲が局所的になるので，拡散速度を速めるために次に述べる方式を併用する．

各PEが直接的に知っている負荷量は，隣接PEだけのものである．この情報だけを用いる限り，負荷の割付け先として選択できるPEは隣接PEだけとなり，短時間においては分散が局所的なものに制限されてしまう．拡散範囲を拡大するためには，より広範囲の負荷情報を収集する必要がある．このため，3.2節までに述べた負荷管理機構を用いて，小さいオーバヘッドでより広範囲の負荷管理を目指す．

プロセスの割付け元となるPEは，次の手順で割付け先のPEを探索する．

1. 自分を第0層の管理PEとして階層領域を形成する．
2. 探索メッセージを第1層の管理PEに送出する．
3. 管理PEは，自分が管理する下位階層の管理PEに探索メッセージを転送する．
4. 最下層まで到達するか，保持されていた負荷情報(3.4節)があれば，最小負荷のPEの位置と，その負荷量を要求元の管理PEに返送する．
5. 管理PEは下位階層から受け取った負荷量の内で，最も小さいものを上位階層の管理PEに返送する．
6. 第0層管理PE(割付け元PE)では，第1層管理PEから返された負荷量の内で，最も負荷の軽いPEにプロセスを割り付ける．

この方式では，プロセスを一度に移送できる範囲が，第1階層で管理するすべてのプロセッサ領域の範囲に制限される．

相互結合網の通信状況によっては，下位階層からの返信の遅延が大きいことも考えられる．上位階層の管理PEは返信待ちの制限時間を設定し，それまでに返信メッセージが到着しない領域については負荷量を最大とみなす(プロセス割付けの対象外にする)などの処置が必要である．

問い合わせメッセージには，問い合わせを行ったPEのアドレスを付加して下位階層の管理PEに転送する．管理PEは常に固定の上位領域に属しているのではなく，別のPEが第0階層管理PEになった時は，その都度，プロセッサ領域やその階層が再構成されるため，その状況に応じて管理PEでなくなったり，また，別のレベルの管理PEとして負荷分散制御に関与することになる．したがって返送先のPEのアドレスは必要な情報である．返信メッセージには，返信を行った階層のレベル，そのレベルでの最小負荷のPEのアドレス，およびその負荷量が含まれる．管理PEは複数の階層レベルの管理PEになるため，階層レベルは必要な情報である．

3.4 負荷情報の保持

前節で述べた方式では，プロセスの割付け時に負荷情報を収集するため，プロセスが頻繁に生成されると，負荷探索メッセージが大量に発生する．このような事態は，負荷分散オーバヘッドの増大につながるため，負荷探索トラフィックの削減が必要となる．

あるプロセッサ領域の管理PEは，プロセッサ領域の階層が再構成された場合に再び管理PEになり得る．このため，一度収集した負荷情報を一定時間，階層毎に保持することで，下位階層の領域の負荷探索トラフィックを削減できる．

負荷情報を保持する時間が長いほど，負荷探索トラフィックの削減には効果がある．しかし，得られる負荷情報の信頼性は低下する．このため，あるタイミングで保持した負荷情報を破棄する必要がある．

図4に負荷量問い合わせのタイミング例を示す．ここではN層の構成を採った場合を考える．第n層管理PEと第$(n+1)$層管理PEの間の平均通信時間をT_n，第n層管理PEがメッセージを伝搬するまでの平均遅延時間をD_n，最下層管理PEが隣接PEから負荷情報を収集し上位階層に返送するまでの平均遅延時間をPとする．このとき，第n層管理PEの問い合わせから第$(n+1)$層管理PEの応答までの遅延時間W_nは，おおよそ次

のように定義できる．

$$W_n = \begin{cases} 2\sum_{i=n}^{N-2} T_i + 2\sum_{i=n+1}^{N-2} D_i + P & (n < N-2) \\ 2T_{N-2} + P & (n = N-2) \end{cases}$$

負荷割付け元 PE が最終的に負荷情報を収集し，割付け先 PE を決定する時刻，つまり負荷割付け元 PE の問い合わせから W_0 だけ経過した時刻を**保持限界**とする．この時刻以降は，負荷割付け元 PE が負荷を実際に割り付けるため，負荷状態が変化してしまう．また、各階層の管理 PE は上位階層に負荷量を返送する時点で，上位階層の平均通信時間とメッセージの平均伝搬遅延時間から保持限界を求めることができる．

既に負荷情報を保持している管理 PE が上位階層から負荷情報の問い合わせを受けた場合には，次のようにして保持された負荷情報を返すかどうかを決定する．まず，問い合わせを行った時に最下層まで問い合わせが伝わる時間 $W_n/2$ を考慮し，最下層の負荷情報が得られる時刻を予測する．この時刻が前回の問い合わせから求められた保持限界を越えていると，得られる負荷量は負荷割付け元 PE が実際に負荷を割り付けた後なので，負荷状態が変化している．したがって，この場合には保持された負荷量をクリアし，下位階層へ再度負荷情報の問い合わせを行う．保持限界を越えていない時は，少なくとも前回の負荷割付けによって負荷状態が変化している可能性は少ないので，保持している負荷情報を上位階層に返送する．

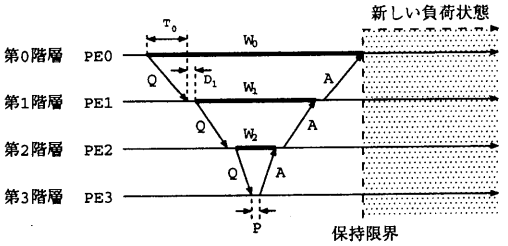

図 4: 保持情報破棄のタイミング

4 おわりに

本稿では，これまでに提案されている主な動的負荷分散方式を超並列計算機に適用する際の問題点を明らかにし，その問題点を解消する方式の提案を行った．本方式は空間的に多重化された階層領域によって負荷を管理し，プロセス割付け先を広範囲の PE から選択できるようにした．これにより，隣接 PE だけに負荷を割り付ける方式に比べ，負荷拡散速度の改善が期待できる．しかし，プロセス割付け時に負荷情報を収集するため，割付けが頻繁に発生すると，負荷情報収集のためのトラフィックが大量に発生し，オーバヘッドが増加する．本研究では，負荷情報を収集する際に一定時間保持することで，トラフィックの増加を抑えるようにした．また，管理 PE の決定方法について，ネットワークトポロジに比較的依存しない方式を定式化した．

今後は本方式の性能評価を行い，より多くのネットワークトポロジに対する適用方法や，システム規模に応じた最大分散範囲の決定方法を確立したい．

謝辞

本研究の一部は，文部省科学研究費補助金・試験研究 (B)(1) No.07558156 の補助による．

参考文献

[1] 古市昌一, 中島克人, 中島浩, 市吉伸行: スタック分割動的負荷分散方式とマルチ PSI 上での評価, 電子情報通信学会技術研究報告 CPSY91-8, pp.33-40,1991.

[2] 佐藤令子, 佐藤裕幸: 疎結合型マルチプロセッサ上の拡散型負荷分散の一方式, 電子情報通信学会技術研究報告 CPSY92-9, pp.1-7,1992.

[3] 佐藤令子, 佐藤裕幸, 中島克人: 疎結合型マルチプロセッサ上の拡散型動的負荷分散方式=LLS-G 方式=, 並列処理シンポジウム JSPP'93, pp.363-369,1993.