テクニカルノート

超並列計算機向き負荷量予測型動的負荷分散方式の改良

布目 淳[†] 平田博章[†]
新實治男[††] 柴山 潔[†]

負荷変化速度を用いて負荷量を予測する動的負荷分散方式において，さらに高精度の予測を行うための指標として，負荷変化加速度を導入した．また，より大規模な並列計算機システムに対応できるように，負荷情報の交換方式を改良した．これらの改良によって，すでに提案している負荷量予測型方式の性能を最大で約15%改善できる．

An Improvement of Dynamic Load Balancing Scheme with Load Prediction Mechanism for Massively Parallel Computers

Atsushi Nunome,[†] Hiroaki Hirata,[†] Haruo Niimi[††]
and Kiyoshi Shibayama[†]

We had proposed a dynamic load balancing scheme which employs load prediction from past load variation speed. In this paper, we introduce the load variation acceleration to enhance the accuracy of the prediction, and also optimize the load management scheme. Our new scheme can achieve the performance improvement by 15% than the previous one.

1. はじめに

我々は，すでに文献1)において，(i) 負荷量の時間的変化（負荷変化速度）を用いて各要素プロセッサ（Processing Element; PE）の負荷量を予測すること，(ii) 通常のプロセッサ間通信メッセージに負荷情報を付加して配送することにより，負荷情報の更新状況を部分的に改善すること，の2点を特徴とする動的負荷分散方式を提案している．この方式により，それまでに提案されていた負荷量の絶対量の比較のみで負荷分散を行う方式に比べて，優れた負荷分散効果が得られるとともに，PE間での局所的な負荷情報の交換において，その更新頻度を低く抑えても十分な負荷分散効果が得られることが確認できた．

負荷変化速度を用いて予測を行う目的は，並列プログラムの実行過程において，抽出される並列性が多い過程や少ない過程に応じて的確な負荷分散戦略を実施することにあり，実際，抽出される並列性が急激に増減する場面では大きな効果を発揮する．しかし，より詳細な調査の結果，PE間でほぼ負荷が均衡している状態では，わずかな負荷量の変化に過敏に反応し，その結果，不必要な負荷移送を繰り返して，プログラム実行時間の短縮を阻害するという問題が明らかとなった．

そこで，今回は，将来の負荷量をより正確に予測するための指標として変化加速度を導入することにより，動的負荷分散方式を改良した．また，効率的に負荷情報を収集するよう，負荷情報の交換方式にも改良を施した．

2. 提案方式

ここでは動的に展開される探索木の1つの節点の処理を1つの並列処理単位とし，これを「アクティビティ」と呼ぶことにする．PEに割り付けられたアクティビティの個数を，その「PEの負荷量」と定義する．なお，本方式ではネットワークトポロジとして2次元メッシュを対象とする．

2.1 負荷情報の交換方式

文献1)の方式では，自PEから一定距離内に存在

† 京都工芸繊維大学工芸学部電子情報工学科
Department of Electronics and Information Science, Kyoto Institute of Technology
†† 京都産業大学工学部情報通信工学科
Department of Information and Communication Sciences, Kyoto Sangyo University

する PE の負荷情報を，各々区別して記憶・管理していた．負荷分散戦略を実施する際には，自 PE を基点とする 2 次元メッシュ上の方向で近接 PE を分類し，そのグループごとの負荷量の合計値を比較した．この方式の提案の狙いは，負荷がどこからどの方向へ広がろうとしているのか，全体的な負荷の流れを負荷分散戦略に反映させる点にあった．しかし，コストやオーバヘッドの問題で，それまでの方式と同様，負荷情報を収集する範囲は近傍に限られ，十分に目的を達成しているとはいえなかった．そこで，今回は，大規模な並列計算機環境においてもシステム全体の負荷情報を効率良く収集できるよう，以下のように改良する．

今回提案する方式では，システムの規模に応じた個数の,「負荷情報収集トークン（以下，トークン）」と呼ぶ固定長のデータを巡回させる．たとえば $N \times N$ メッシュのシステムであれば，縦方向と横方向の軸上に各々1 個ずつ，全体で $2N$ 個のトークンを軸方向に沿って往復させる．

システム内の各 PE は，隣接 PE からトークンを受信すると，そのトークンに記録されている負荷情報を自 PE 内にコピーし，自 PE の負荷量を加えた値をトークンにして，トークンを受け取った側とは反対側の PE へ送信する．ただし，メッシュの辺上に位置する PE は，トークン内の総負荷量をクリアして自 PE の負荷量のみを送り返すため，トークンにはメッシュ端からの総負荷量が記録されることになる．

このように，各々の PE では縦横の軸上を往復する 2 つのトークンが通過するため，両軸上にある PE 群の負荷量を知ることができる．さらにこのことを利用して，トークンがある PE を通過する際には，トークンの進行方向に向かって右側と左側に存在する PE 群の負荷量についても，それぞれを加算する．これら 3 種類の値を記録したトークンが直交軸上を往復することにより，自 PE を原点とする直交軸上の 4 つ（東西南北）の PE 群の総負荷量と，軸で分割される 4 つの領域の総負荷量を得ることができる．これらの情報をもとに，それらの領域間で負荷量を平均化することにより，システム全体の負荷量の均衡を図る．

なお，メッシュサイズが非常に大きく，トークンの往復時間が長くなりすぎると，負荷情報を取得する遅延時間が長くなり，負荷割付けを正確に行うことができなくなる．このような環境では，1 つの軸上で複数個のトークンを往復させる．

2.2 負荷割付け先の決定方法

各 PE では，トークンから得た負荷情報を定期的に保存し，それらの時間的な変化から将来の負荷量を予測する．負荷量の予測は領域単位で行い，現在の負荷変化速度を s，履歴中の変化速度の時間変化から求めた負荷変化加速度を a，トークンから得た領域の総負荷量を l とすると，現在より時間 t 後の予測総負荷量 L は，単純な積分により，

$$L = \frac{a}{2}t^2 + st + l \qquad (1)$$

となる．ただし，各領域中の PE 数には差があるため，各 PE では領域内の 1 PE あたりの予測負荷量を考え，これが領域間で平均化されるように，以下のように負荷の割付け先と割付け量を決定する．

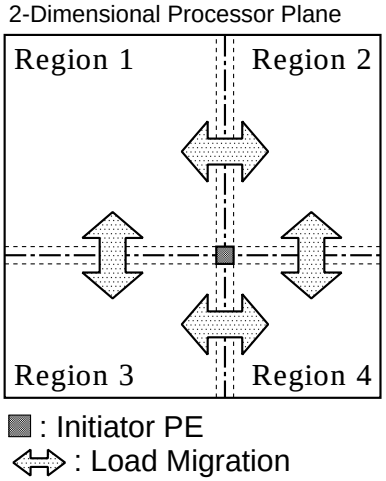

図 1 4 領域間での負荷移送
Fig. 1 Load migration among four regions.

まず，トークンの経路にあたる領域は，2 つの大きな領域の境界線であるため，この領域の負荷量は隣接する領域へ均等に配分し，**図 1** の Region 1〜4 のように 4 つの領域の間で負荷の平均化を図る．最初に，前回提案の方式[1]と同様に，システム全体の総負荷量を全 PE 数で割ることで，1 PE あたりの目標負荷量を求める．この目標負荷量と領域に含まれる PE 数をもとに，4 つの領域のそれぞれに存在すべき目標負荷量を求め，実際に各領域に存在する負荷量が，これに対してどれだけ過不足しているかを調べる．これにより，領域間でどれだけの量の負荷を移送すれば有効かが求まる．各 PE は，領域間での負荷の移送量を領域の境界面に位置する PE で等しく分担するものとして，自 PE がどれだけの負荷量を隣接 PE との間で移動するかを決定する．

3. 性 能 評 価

3.1 評 価 条 件

2 次元メッシュトポロジの超並列計算機を，汎用シ

表 1 システムパラメータ
Table 1 Parameters of simulator.

メッシュサイズ	16 × 16 (256 PE)
メッセージ送信時間	20 [単位時間]
トークン送信時間	1 [単位時間]
負荷量予測間隔	1000 [単位時間]
ルーティング方式	固定ルーティング

表 2 評価したプロセスのパラメータ
Table 2 Parameters of estimated process.

基本生成数	生成確率	実行時間	移送時間
5	0.9	500	200

ステムレベルシミュレータ SES/workbench[2]によってモデル化した．シミュレーションに用いたパラメータ値を**表 1** に示す．ここで「メッセージ送信時間」と「トークン送信時間」は，それぞれアクティビティ間の通信メッセージとトークンが，競合なしの場合にリンクを占有する時間である．また，将来のある時刻における予測負荷量を利用する場合，その時刻と現在時刻との時間間隔を「負荷量予測間隔」として示している．これは，式 (1) の t に相当する．この間隔は，全体的な負荷変化の傾向がつかめる程度に長く，また，情報として古くなりすぎないよう短くなければならない．今回は，アクティビティの平均実行時間よりも適度に長い間隔となるよう，1000 [単位時間] に設定した．

このシステムに，アクティビティが**表 2** に示した性質を持つプロセスを投入し，その実行時間を測定する．各々のアクティビティは，表 2 の生成確率に従って子アクティビティを生成する．ここで，探索問題の処理の中盤では探索対象となる枝が増えることを反映させるために，各世代によって生成する子の数を変化させた．具体的には，2 世代目までは基本生成数の子アクティビティを生成し，その後，5 世代目までは 6 世代後に生成数が 2 倍になるよう，親世代での生成数に $\sqrt[6]{2}$ を乗じた個数の子アクティビティを生成する．また，6 世代目以降は，3 世代後に生成数が 1/4 倍になるよう，親世代での生成数に $\sqrt[3]{1/4}$ を乗じた個数の子アクティビティを生成する．

各アクティビティは実行時に親アクティビティおよびすべての兄弟アクティビティを相手に通信を行うものとした．その頻度は，提案方式において負荷量の予測を行わないものとした場合にシステム内の平均リンク利用率☆が 10%，50%，80%程度になるようアクティビティの平均通信回数を調節し，それぞれについて評価を行った．

提案方式では，隣接 PE へ送信するトークンを作成する時間として 1 [単位時間] がかかるものとした．また，負荷量予測に用いる負荷変化速度および加速度の値の更新に 5 [単位時間]，負荷の割付け先を決定する際に変化速度，変化加速度を計算するためのコストとしてそれぞれ 2，4 [単位時間] を要するものとした．

一方，前回提案した負荷量予測型方式[1]（以下，従来方式）では，自 PE から距離が 3 以内にある PE の負荷情報を利用するものとした．負荷変化速度を考慮するか否かにかかわらず，PE 間に生じる通信メッセージに負荷情報を付加し，負荷情報の更新間隔を部分的に短縮している．隣接 PE と負荷情報を交換する間隔は，アクティビティの平均実行時間のそれぞれ 2，4，8 倍となる 1000，2000，4000 [単位時間] に設定した．また，PE 1 つの負荷情報を送信するのに 1 [単位時間] がかかるものとし，その他は，これを基準として従来方式[1]と同じ比率でコストがかかるものとした．

3.2 評価結果

リンク利用率を変化させた場合の両方式におけるシミュレーション結果を，**図 2**，**図 3**，**図 4** にそれぞれ示す．これらの図では，従来方式で速度を考慮しない場合の実行時間を 1 として，これに対する性能向上比を示した．なお，従来方式は LIP と表示している．

速度を考慮しない提案方式に着目すると，図 2，図 3，図 4 のいずれにおいても，従来方式より性能が改善されている．従来方式では，1 つの PE が負荷情報を収集する範囲を広げるに従って管理コストが増大する．今回は，自 PE から距離が 3 以内にある PE の負荷情報を利用したが，これは最も多くの負荷情報を利用できるメッシュ中央部においても，24 個の PE の負荷情報にすぎない．これに対して，提案方式では，全 PE から自 PE を除いた 255 個の PE に関する負荷情報を利用している．負荷量の予測を行わない場合でも，従来方式より性能が改善されていることから，今回の負荷情報交換方式の改良の有効性が確認できる．

提案方式で負荷量の予測に変化加速度を用いる効果は，リンク利用率が高い状況下でより顕著に現れており，変化速度のみを用いる場合に比べて最大で約 3%性能が向上している．図 2 のように，リンク利用率が低く，ネットワークに余裕がある場合は，変化速度による予測と変化加速度による予測の結果にあまり大きな差はない．しかし，アクティビティ間の通信が頻繁

☆ 通信リンクの両側の PE 上に実行可能なアクティビティが存在する時間のうち，アクティビティ間の通信メッセージによってリンクが使用された時間を T_{use}，リンクが使用されていない時間を T_{free} として，リンク利用率を $T_{\mathrm{use}}/(T_{\mathrm{use}}+T_{\mathrm{free}})$ と定義する．

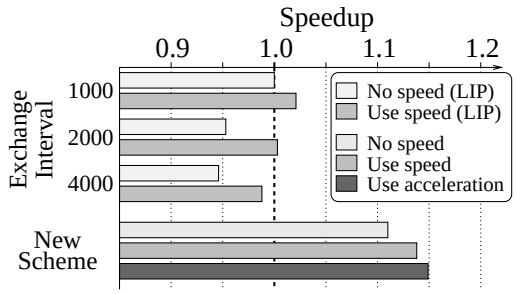

図 2 リンク利用率 10%での性能
Fig. 2 Performance under the condition of 10% link utilization.

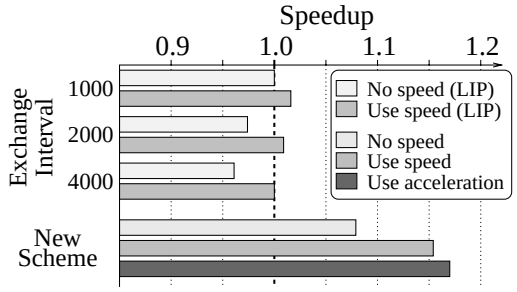

図 3 リンク利用率 50%での性能
Fig. 3 Performance under the condition of 50% link utilization.

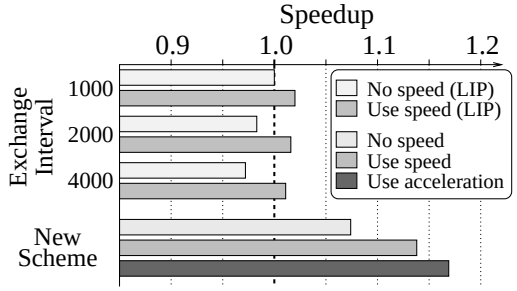

図 4 リンク利用率 80%での性能
Fig. 4 Performance under the condition of 80% link utilization.

に行われる状況では，リンクを長時間占有する負荷移送によって通信メッセージの待ち時間が長くなり，その結果，総実行時間も長くなってしまう．提案方式では，変化加速度を用いてより精度の高い予測を行うことで，負荷移送の回数を最小限に抑え，性能の低下を回避できている．

4. おわりに

本論文では，前回提案した負荷量予測型動的負荷分散方式[1)]に対して，

- 負荷量予測に負荷変化加速度を用いることで予測精度を改善する，
- 効率良く全システム中の負荷情報を収集する方式を提案する，

の 2 点で改良を行った．シミュレーションによる評価の結果，今回の改良により，約 15%の性能改善が得られることが分かった．

今後は，並列度が動的に変化する代表的なアプリケーションを用いて，さらに詳細な評価を行う予定である．

謝辞 本研究の一部は，文部省科学研究費補助金（10480062，11480069，11878052，12780218）の補助による．

参考文献

1) 布目 淳，平田博章，新實治男，柴山 潔：超並列計算機のための負荷変化速度を考慮した動的負荷分散方式，電子情報通信学会論文誌（D-I），Vol.J83-D-I, No.9, pp.936–945 (2000).
2) Scientific and Engineering Software, Inc.: *SES/workbench Release 3.2 Creating Models* (1998).

(平成 12 年 12 月 22 日受付)
(平成 13 年 2 月 1 日採録)