

3 L-7

PVMにおける負荷情報に基づいたタスク割り当て方式

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

1 はじめに

近年、科学技術計算を行う際に処理時間の短縮のために並列処理が用いられている。並列処理を行うには並列計算機を用いる方法と分散システムを仮想的に並列マシンとして用いる方法がある。後者を仮想並列マシンと呼び、これを実現する代表的なソフトウェアにPVMがある。仮想並列マシンには、各ノードの負荷を考慮してタスクの割り当てる、すなわち負荷分散をしなければ、良い性能が得られないという問題点がある。しかし、PVMでは負荷を考慮してタスクを割り当てることが出来ない。そのため、負荷分散の有効性と問題点が明らかになっていない。

本研究では仮想並列マシンにおける負荷分散の有効性と問題点を明らかにするために、PVM上に負荷分散アルゴリズムを実装し、負荷分散を行わない場合との並列プロセスの応答時間を比較した。また、その時の各ノードの負荷の時間推移の測定を行った。結果から、負荷分散を行うことにより、並列プロセスの応答時間の短縮が出来るという利点が明らかになった一方で、ノードの負荷が高い時に負荷の集中が起こるという問題点が分かった。

2 PVMにおける負荷分散の効果と問題点

PVM上でタスクの割り当てに負荷分散を行う場合と行わない場合の並列プロセスの応答時間を比較する。この実験では(1)並列プロセスの応答時間、(2)各ノードの負荷の時間推移、(3)負荷分散を行う際に必要な負荷情報を得るためにオーバヘッドを測定した。

2.1 負荷分散アルゴリズム

負荷分散アルゴリズムとして、Biddingアルゴリズムを用いる。Biddingアルゴリズムは定期的に収集する各ノードの負荷情報に基づいてタスクの割り当てを行うアルゴリズムである。また負荷分散を行わないアルゴリズムとしてNLS(No Load Sharing)アルゴリズムを用いる。NLSアルゴリズムは各ノードの負荷状態を考慮せず、タスクを全てのノードに割り当てるアルゴリズムである。Bidding、NLSアルゴリズムのフローチャートをそれぞれ図1、2に示す。

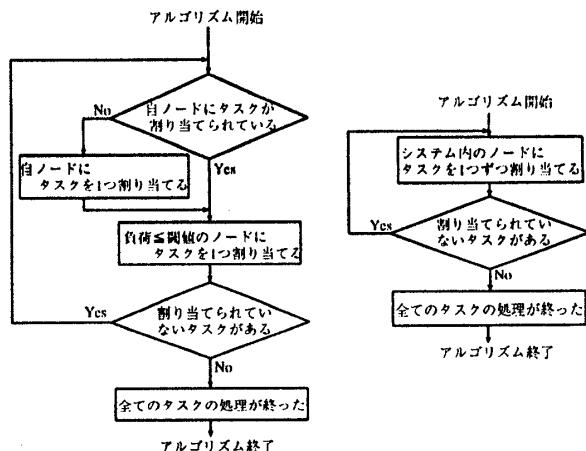


図1: Bidding アルゴリズム 図2: NLS アルゴリズム

2.2 実験環境

処理能力の異なる5台のWSを用いて実験を行った。以下に各WSのハードウェア構成を示す。
評価に使用したWS

ノード名	機種・OS	Memory size
Node A	SS20,Solaris2.3	32Mbytes
Node B	SS5,Solaris2.3	32Mbytes
Node C	SUN IPX,SunOS4.1.4	32Mbytes
Node D	SUN IPX,SunOS4.1.4	32Mbytes
Node E	SS2,SunOS4.1.3	32Mbytes

2.3 実験方法

実験に際して、並列プロセスは一台のWSのみで発生するものとし、PVMを用いて他のWSに分散させる。並列プロセスの発生は実験で使う計算機の中で最も処理能力の高いNode Aで行った。各ノードの負荷としてプロセス数を用いた。負荷情報の収集間隔は15秒である。また、全てのノードで他のユーザプロセスが無い状態で行った。

実験に用いた並列プロセスはMandelbrot集合の計算である。Mandelbrot集合の計算はループ間に依存関係が無く、並列に処理する時にタスク間の通信が必要ないという特徴を持つため、仮想並列マシンの負荷分散方式の評価に有効な例題である。計算を行うMandelbrot集合の領域は $-0.5 \leq x \leq 1.5$, $-1 \leq y \leq 1$ とした。実験ではこの領域を5つに分割し、負荷分散アルゴリズム

ムに従って各ノードに割り当てる。

2.4 実験結果と考察

2.4.1 負荷分散による並列プロセスの応答時間の短縮

NLS, Bidding アルゴリズムを用いた時の並列プロセスの平均応答時間を測定した結果を図 3 に示す。
 $\rho = (\text{並列プロセスの平均処理時間}) / (\text{並列プロセスの平均到着時間})$ である。

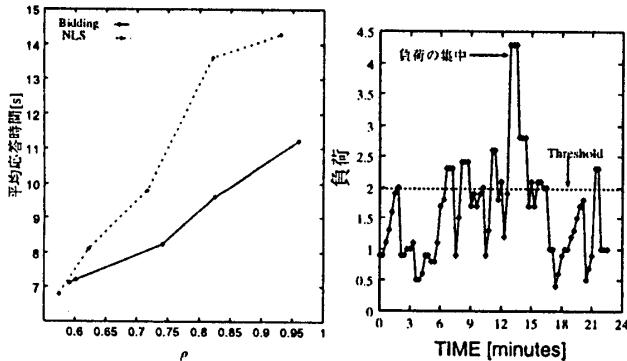


図 3: NLS, Bidding アルゴリズムを用いた場合の平均応答時間
図 4: Node D の負荷の時間推移

図 3 より、 ρ が 0.6 以上、すなわち、システムの負荷が中程度以上では Bidding アルゴリズムの方が平均応答時間が短く、性能が良いことが分かる。図 4 は Bidding アルゴリズムにおいて $\rho = 0.96$ の時、すなわち、システムの負荷が高い時の Node D の負荷の時間推移である。この図より、Bidding アルゴリズムにおいてシステムの負荷が高い状況では、負荷の集中が起きることが分かる。

負荷の集中は、負荷情報の収集間隔が長いために信頼性のある負荷情報が得られない、しきい値が一定であるために各ノードの負荷が高い時はタスクを割り当てるごとの出来るノード数が少なくなる、ということが原因であると考えられる。この負荷の集中は、仮想並列マシンにおけるタスクの応答時間の増大、システム動作の不安定を引き起こす。負荷の集中は、負荷分散によってシステム内のノードの負荷を均一にするという観点から改善すべき問題点である。

2.4.2 負荷情報を得るためにオーバヘッド

負荷分散を行うには負荷情報を取得することが必要だが、同時に、負荷情報の取得にはある程度のオーバヘッドがある。そこで次の実験では、各ノードの負荷情報を得るために要するオーバヘッドを測定した。NLS, Bidding アルゴリズムに従って計算を行わないタスクを割り当てる時の各ノードの負荷を図 5 に示す。

この結果より、負荷情報を得るために各ノードに大きな負荷を与えることが分かる。負荷情報の収集によるオーバヘッドは出来るだけ小さくすることが必要であるが、そのためには負荷情報の収集間隔を大きくしなければならない。しかし、負荷情報の収集間隔を大きくすると信頼性の高い負荷情報が得られなくなる。

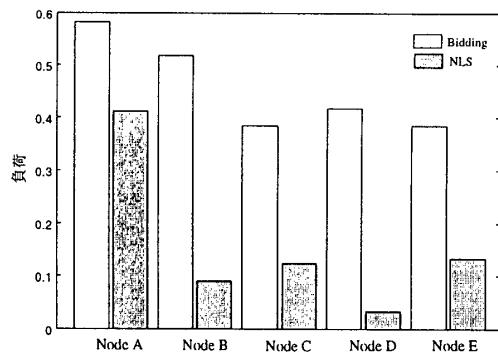


図 5: 負荷情報の取得がノードに与える負荷
そのため、最適な負荷情報収集間隔を定める手法が必要である。

2.5 負荷分散における問題点と対策

今回の実験結果から、PVM において負荷分散を行うとき、ノード全体の負荷が高い場合、負荷の集中が起こる、といった問題点があることが分かった。この問題点を解決する方法として、しきい値を動的に変化させること、負荷情報収集間隔を小さくすることが考えられる。しかし、負荷情報の収集間隔を小さくすると信頼性の高い負荷情報が得られる代わりに負荷情報取得のためのオーバヘッドが大きくなるというトレードオフがある。よって、このトレードオフにおいて最適な負荷情報収集間隔を定める手法が必要である。

現在、今回の実験で明らかになった問題点を解決するために、システムの負荷状況にあわせて動的にしきい値を変化させ負荷の集中を抑制する負荷分散アルゴリズムの開発を完了し、その評価中である。について研究を行っている。また、負荷情報収集間隔とオーバヘッドのトレードオフの関係について、最適な負荷情報収集間隔を定める手法について研究を行っている。

3まとめ

本稿では PVM 上でタスク割り当てに負荷分散を行うことによって並列プロセスの応答時間と各ノードの負荷の時間推移を測定した。その結果、負荷分散を行うことにより並列プロセスの応答時間が短縮されることが分かった。また、負荷分散を行うことによる問題点が明らかになった。

参考文献

- [1] S. Zho "A Trace-Driven Simulation Study of Dynamic Load Balancing" IEEE Trans.Softw.Eng. Vol.14, No.9, Sep. 1988
- [2] J. Pruyne et al "Providing Resource Management Service to Parallel Applications" Proc. of the Second Workshop on Environment and Tools for Parallel Scientific Computing, May 1994
- [3] J. Pruyne et al "Parallel Processing on Dynamic Resources with CARMI" Workshop on Job Scheduling Strategies for Parallel Processing IPPS'95, April 25 1995