

1 はじめに

並列処理の一分野であるワークステーションクラスタ型並列処理は、複数のワークステーション (WS) をネットワークにより接続した分散システム上で行う並列処理である。この分野では負荷分散方法の選択次第でシステム全体の効率が良いも悪くなる。負荷分散方法には、静的 (プログラム実行前に負荷分散方法を決定) なものと、動的 (プログラム実行時に負荷分散方法を決定) なものがある。

並列処理で問題解決をする際に、各プロセッサが同一のプログラムのコピーを実行する実行形態を SPMD (Single Program Multiple Data) モデルと呼ぶ。WS クラスタ環境における SPMD モデル上で負荷分散に伴う通信オーバーヘッドを最小限に抑えるためのトポロジー (図2参照) と、それに基づく静的負荷分散手法 [?] が提案された。

本論文では、WS クラスタ環境における SPMD モデル上での動的負荷分散の一手法を提案し静的負荷分散手法との比較実験を行うことにより、提案手法の有効性を確認する。以後、WS をプロセッサと表記する。

2 提案手法

2.1 システム構成

図1に提案手法のシステム構成を示す。図中の細い矢印はタスクの流れを、太い矢印はメッセージの流れを示す。システム中には二つの Queue があり、Active Queue には他のプロセッサから受信したタスクを格納し、Schedule Queue には自分のタスクを格納する。また三つのスレッドが動いており、個々の役割はまず Check Message スレッドは受信したメッセージをチェックし、Estimate Load スレッドは一定時間毎のタスク処理数 (Power) を他のプロセッサに送信し、Processing Task スレッドは二つの Queue のタスクを Active Queue 優先で処理する。

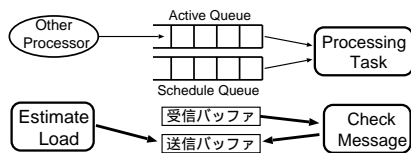


図 1: システム構成

2.2 負荷のパラメータ化による動的負荷分散

以後プロセッサ i を P_i 、 P_i の Power を $Power_i$ と表記する。 P_i が $Power_j$ を P_j から受信した時、もし ($Power_i > Power_j \geq 0$) ならば P_i は P_j に ($Power_i - Power_j$) 個のタスクを自分に送信するように要求する。 $Power$ は、各プロセッサの負荷のパラメータであり、プログラム実行時にプロセッサ間の $Power$ の差分の個数のタスクを移動することが決定されるという点で動的な負荷分散方法である。一方、静的負荷分散手法では移動するタスクの個数は1個とプログラム実行前に予め決められている。

3 実験

3.1 実験方法

元は処理能力が等しいプロセッサ7台の内3台に約65%の負荷をかけ、各プロセッサは粒度の等しい100個のタスクを持っていて、タスクが無くなり次第終了する。その状態でシステムを実行し、負荷のかかったプロセッサのタスク処理数が少なく、終了時間が短い程、そのプロセッサに対して効率が良い負荷分散がなされたものとする。

3.2 実験に用いたトポロジー

図2に実験に用いたトポロジーを示す。図中の丸の中に数字が書いてあるものはプロセッサを示し、矢印は負荷情報の流れを、数字は各プロセッサの負荷を、網がかかったプロセッサは負荷がかかっていることを示す。

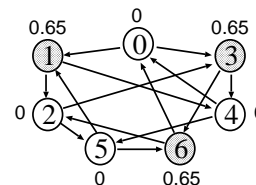


図 2: プロセッサ7台の時のトポロジー

3.3 実験結果

表1に各プロセッサの終了時間を、表2に各プロセッサのタスク処理数を示す。負荷のかかっている P_1, P_3, P_6 どのプロセッサも提案手法の方が終了時間が短く、タスク処理数が少ないので負荷分散の効率が良いといえる。

負荷分散方法	終了時間 (s)						
	P_0	P_1	P_2	P_3	P_4	P_5	P_6
静的	52.97	114.20	54.83	147.93	45.68	46.13	137.70
提案手法	68.29	44.71	52.79	141.76	55.52	55.60	103.24

表 1: 各プロセッサの終了時間

負荷分散方法	タスク処理数 (個)						
	P_0	P_1	P_2	P_3	P_4	P_5	P_6
静的	107	77	121	99	101	102	93
提案手法	150	30	116	95	119	121	69

表 2: 各プロセッサのタスク処理数

4 まとめ

静的負荷分散手法との比較実験により、提案手法の有効性が確認できた。しかしながら、本負荷分散手法は局所的な2プロセッサ間の負荷を平均化するものであるため、システム全体の負荷平均化については今後の課題である。

参考文献

- [1] Matthew P. Tiemeyer, Johnny S.K. Wong: "A task migration algorithm for heterogeneous distributed computing systems", The Journal of Systems and Software, VOL.41, NO.3, JUNE 1998