2H-06　マルチプロセッサ環境における マイグレート可能タスクの導入

大畠 祐一*‡　石川 知雄†‡　　高田 広章§¶

1　はじめに

近年，大規模なリアルタイム制御システムの高速化に対する要望が高まってきている．高速化の手法の一つとして，多数の入出力装置が接続されるような制御システムにおいては，システムを複数のプロセッサエレメントで構成し，プロセッサエレメントの処理を各入出力装置に限定させ，機能を分散させる機能分散型マルチプロセッサシステムの採用を挙げられる．図1に機能分散型マルチプロセッサの典型的な例を示す．

マルチプロセッサの環境においては，要求性能の変化によって，入出力装置の増減や，プロセッサ数の変更が生じることがある．その際にシステムの設計変更を最小限にとどめる必要がある．つまり，システムにスケーラビリティが要求される．プロセッサ数などが増加すると，資源の競合などが生じる可能性が増大し，デッドラインが守れなく恐れがある．その結果，システムのスケーラビリティに対する要求を満たすことができなくなってしまう．そこで，タスクを各プロセッサエレメントに閉じた処理とエレメント間での通信などの処理の二つのクラスに分類する事によってこの問題を解決しようという試みがなされている．[1]

本研究では，タスクをクラスに分類することに注目して，機能分散型マルチプロセッサにおいて，新たにプロセッサエレメント間にまたがった処理を実行できるタスククラスを導入し，システム全体のスループットを向上させることを目的としている．

*yuichi oohata
†tomo ishikawa
‡Information and Communication Labo. Musashi Institute of Technology
§Hiroaki Takada
¶Dept. of Information and Computer Sciences, Toyohashi Univ. of Technology

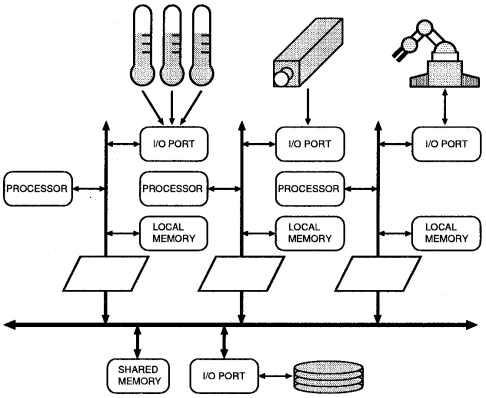

図 1: 機能分散型マルチプロセッサシステム

2　グローバルクラス

前述の二つのタスククラスはプロセッサエレメントに処理するクラスをプライベートクラスと呼び，プロセッサエレメント間の通信などを行うタスククラスをローカルクラスと呼ぶ．前者は高い優先度を持つが，資源に対する利用可能性の制限は多く，後者は，優先度は前者よりも低いが，資源に対する利用可能性の制限が少ない．

本研究で新たに実装するタスククラスをグローバルクラスと呼ぶ．また，以下では，グローバルクラスに属するタスクをグローバルタスクと呼ぶ．グローバルクラスは優先度の点では前述の二つのタスククラスよりも低く設定されており，また，資源に対する利用可能性についてもローカルクラスよりは制限が多い．しかし，グローバルクラスには以下のような特徴がある．

- 処理に使用するプロセッサエレメントを選ばない．
- 途中でプリエンプトされたとしても，他のプロセッサエレメントがアイドリングして

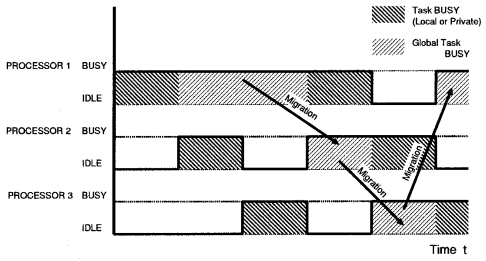

図 2: タスクマイグレーション

いれば，そこで処理を再開できる．

これらのことにより，システム中のアイドリングしているプロセッサを有効活用することによって，システム全体のスループットを向上させる．タスクが他のプロセッサに処理を移すことをタスクのマイグレーションと呼ぶ．タスクマイグレーションの概念を図 2に示す．グローバルタスクのコードおよびデータは，共有バス上に接続された共有メモリに存在しているため，これらの操作が可能になる．

3 タスクマイグレーションによる性能向上

タスクのマイグレーションを実装することによって，システムの性能はタスクマイグレーションによるシステムのスループットの向上と，タスクマイグレーションを行うことによるオーバヘッドとの二点によって決定する．ここでは，タスクマイグレーションの実装の予備実験をもとに性能向上についての考察を行う．

予備実験ではタスクマイグレーションにおけるバス使用のオーバヘッドとシステム全体のスループットの変化を調査する目的で行った．周期的に一定時間，優先度の高いタスクがプロセッサを占有し，その合間にグローバルタスクが処理

	タスク周期 (msec)	タスク占有時間 (msec)
モデル A	15	5
モデル B	30	10
モデル C	15	10

表 1: 実験モデル

	タスクマイグレーション	
	有り	無し
モデル 1	1272194	1300233
モデル 2	999386	1297064
モデル 3	1931813	1299894

表 2: 実験結果

を実行した場合にグローバルタスクが行った処理量をを調べることを目的としている．また，実験に際しては表 1 に示す三つのモデルを考えた．

表 2 にタスクマイグレーション実装の予備実験の結果を示す．この数値は相対的なものであり単位はない．予備実験の結果から，タスクのマイグレーションを行う際のオーバヘッドとなると考えられたのが，共有メモリをアクセスする際のオーバヘッドである．共有メモリはローカルメモリと比較してアクセスの際により多くの時間がかかる．また，バスの使用頻度が上がることにより，システムの性能が大きく落ちることが判明した．

4 まとめ

機能分散型マルチプロセッサシステムにおいて，アイドリングしているプロセッサを利用する手法についての提案を行った．また，予備実験の結果によりグローバルクラス実装に際しての方針を示すことができた．また，現在，予備実験の結果から，バスの使用頻度を下げる方向で実装を進めている．

参考文献

[1] 高田 広章，坂村 健，”非対称マルチプロセッサシステムのためのスケーラブルなリアルタイムカーネルの構想”，信学技報 (1995 年実時間に関するワークショップ RTP’95)，vol.94, no.573, pp.1-8, 電子情報通信学会，Mar.1995.

[2] 鍋田 努，大畠 祐一，石川 知雄，高田 広章，”機能分散マルチプロセッサシステムにおけるマイグレート可能なグローバルタスクの導入”，信学技報 (1998 年 実時間処理に関するワークショップ RTP’98) vol.97, no.569, pp. 49-56,Feb.1998