単方向 1 : 1 高速同期機構を用いた組込み制御並列化

中村　陸[†,††]　枝廣正人[†]

組込み制御領域において，マルチ・メニーコア化とこれを活用するための制御ソフトウェアの並列化が進められている．しかし期待される処理性能のボトルネックとして，タスクスケジューリングやプロセッサ間通信のオーバヘッドが存在する．1 コア 1 タスクに静的割付けし，単方向 1:1 高速同期機構による通信を用いることで，モータ制御モデルを用いた性能評価において，実行サイクルを 25 %程度低減することができた．

Parallelize Embedded Control Software with 1 : 1 Synchronization Mechanism

Riku Nakamura[†,††] and Masato Edahiro[†]

In embedded control field, multicore/manycore processors rapidly grow, and therefore, parallelized software needs to be developed. However, overhead of task scheduling and interprocessor communication is large compared with task granularity. In this paper, tasks are statically allocated on one core one task basis, and 1 : 1 synchronization mechanism is applied. With experiments using a motor control model, execution cycles are reduced by 25%.

1. はじめに

近年，組込み制御領域において電力や発熱の問題からシングルプロセッサの性能向上が困難になってきており，マルチ・メニーコアプロセッサが主流となりつつある．しかし，マルチ・メニーコア化しただけではシステムの性能向上は期待できず，ソフトウェアを並列化し，マルチ・メニーコアに対応させる必要がある．

ソフトウェアを並列化すると複数のタスクが生成されるため，タスクスケジューリングが必要となる．また，異なるプロセッサで実行されるタスク間でのデータ共有や同期のためプロセッサ間通信を行う必要がある．これらは並列化によるオーバヘッドとなる．

組込み制御は，タスクの粒度が小さい．また，図 1 のタスクグラフのように各タスクが少数タスクとのみ通信するという特徴がある．

そこで，同一コア内でのタスクスケジューリングを不要とするために，1 コア 1 タスクに静的割付けし，1:1 高速同期機構によるプロセッサ間通信を用いることで並列化のオーバヘッドを削減する手法を提案する．

† 名古屋大学
Nagoya University
†† 萩原電気株式会社
Hagiwara Electric Co., Ltd.

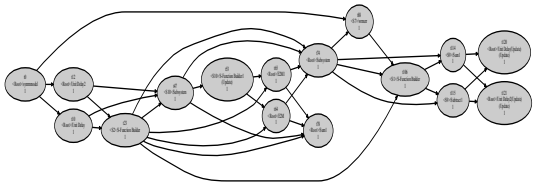

図 1　組込み制御のタスクグラフの例

2. プロセッサ間通信高速化

1 つのコアに対し実行するタスクを 1 つだけ静的に割付け，タスクの生成・終了はシステムの起動・終了時のみ行う．そして，1 つのタスク間通信に対し，1 つの共有メモリ領域と 1 つのフラグを割り当てる．共有メモリは隣接プロセッサ間共有レジスタやローカルメモリに配置することを想定する．タスク間通信は以下のように行う．

書込みタスク: 指定されたメモリアドレスにデータを書込み，フラグを 1 とする．

読込みタスク: フラグをチェックし，1 ならばデータを読込み，フラグを 0 とする．0 ならばウェイト（ループ）する．

2.1 実装手法

本稿では，フィードバック制御を行うモデルを対象と

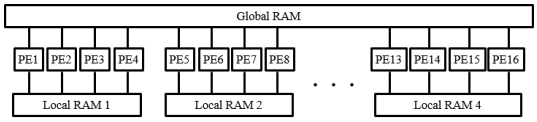

図 2 階層メモリ構成

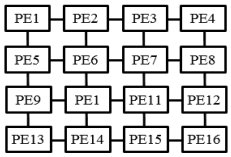

図 3 メッシュ構成

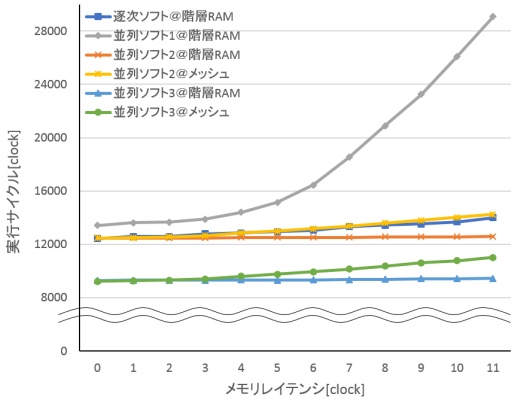

図 4 実験結果

し，MATLAB/Simulink モデルを仮定する．MATLAB/Simulink で記述した制御モデルを Real-Time Workshop Embedded Coder により，逐次 C コードに変換する．この C 逐次コードをモデルベース並列化ツール[1] を用いて並列化する．これにより，MATLAB/Simulink の 1 つのブロックを 1 つのタスクとする並列化が行われる．

3. 評 価 実 験

提案方式によるオーバヘッドの削減効果を確認するため，ルネサスエレクトロニクスのマルチコアシミュレータを用いて実験を行った．マルチコアシミュレータは 4 個のハードウェアスレッドを実行可能な PE を 4 個搭載しており，合計 16 個のハードウェアスレッドを実行できる．ここでは，ハードウェアスレッドをコアとして扱い，1 コア 1 タスクを割り当てた．

3.1 評価ソフト

MATLAB/Simulink のモータ制御モデルから，以下のソフトを作成し比較した．

逐次ソフト: MATLAB/Simulink から生成したソフト

並列ソフト 1: モデルベース並列化ツールで並列化したソフト

並列ソフト 2: 並列ソフト 1 に対し，提案手法を適用したソフト

並列ソフト 3: 並列化ソフト 2 に対し，同期タイミングを調整したソフト

3.2 メモリ構成

階層メモリ構成（図 2）とメッシュ構成（図 3）の 2 つの構成で評価した．階層メモリ構成は，4 コア毎に 1 個の Local RAM と全コアからアクセス可能な Global RAM を配置し，Local RAM へのアクセスレイテンシを 0 サイクルとし，Global RAM へのアクセスレイテンシは 0 サイクルから 11 サイクルまで変化させた．メッシュ構成では，4 コア*4 コアのメッシュ上に並んだ各コアに Global RAM を配置し，Global RAM へのアクセスレイテンシは自コアからのアクセスレイテンシは 0 とし，他コアからのアクセスレイテンシは ((hop 数+1) * (0〜11)) サイクルとした．

3.3 実 験 結 果

実験結果を図 4 に示す．モデルベース並列化ツールを用いて並列化すると，逐次ソフトよりも実行時間が増加したが，提案手法を適用することで，逐次ソフトと同程度の実行サイクルとなった．更に，同期タイミングを調整することで，逐次ソフトよりも 25%実行サイクルを低減した．

3.4 考 察

実験では，階層メモリ構成よりもメッシュ構成の実行サイクルが長くなった．今回の実験では，タスクのコア割付けでタスク間の通信を考慮していないため，通信レイテンシが増加したと考えられる．そのため，タスク間通信を解析し，最適なタスク配置をすることでより高速化が可能と考える．

4. お わ り に

1 コア 1 タスクに静的割付し，単方向 1 : 1 高速同期機構を用いたプロセッサ通信を行うことで並列化のオーバヘッドを削減する手法を提案した．モータ制御モデルを用いた評価を行い，提案手法を用いて並列化を行うことで逐次実行よりも 25%実行サイクルを低減できることを確認した．

参 考 文 献

1) T. Kumura, Y. Nakamura, N. Ishiura, Y. Takeuchi, M.Imai: Model Based Parallelization from the Simulink Models and Their Sequential C Code. SASIMI 2012 Proceedings.