

密結合マルチプロセッサ環境において PREM タスク実行モデルで動作するマルチタスクソフトウェアの性能検証手法

中村 貴史† 中田 明夫†

広島市立大学 大学院情報科学研究科 システム工学専攻†

1. まえがき

性能とリソースの制約が共に厳しく、高い信頼性を要求される組込みソフトウェア開発においては、設計段階で最悪時に性能要求を満たす否かの検証(性能検証)を行うことが有用である[1]. 従来、一定の制御構造に従って起動するタスクが混在したマルチタスク組込みソフトウェアの動作仕様(以下、複数タスク動作仕様)に対してスループット性能検証(スループット(単位時間当たりの入力数)要求を満たすか否かの検証)を行う手法が提案されている[2]. 一方で、近年のリアルタイムシステムは低消費電力化と高性能化の両立を図るため、密結合マルチプロセッサ環境で実装される場合が増えている. そのような環境においては、多数のプロセッサコアがメモリなど少数の共有リソースにアクセスすることとなり、リソース競合に起因する遅延は性能低下やリアルタイム制約違反の要因となる. この問題に対処するため、CPU と周辺機器間のメインメモリアクセス競合を解決するようなタスク実行モデル PREM が提案されている[3]. 文献[2]における性能検証手法を PREM に対応させることにより、性能検証の際に考慮すべき設計の選択肢を増やし、設計対象システムの性能をさらに向上できると考えられる. 本研究では文献[2]の手法を拡張し、密結合マルチプロセッサ環境における PREM タスク実行モデルを使用するシステムを検証可能な性能検証手法を提案する.

2. 複数タスク動作仕様

複数タスク動作仕様[2]は、タスクグラフとリソース割り当て図の組で構成される. タスクグラフとは、順序関係を持つタスク同士を矢印でつなぎ、入力 (Input), 出力 (Output), 並列分岐(par)・合流(join), 選択分岐(choice)・合流(endchoice)などの制御構造を

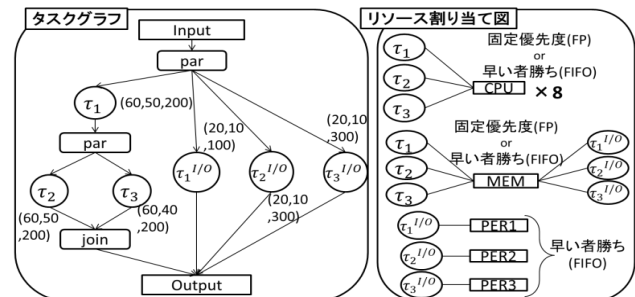


図 1. 複数タスク動作仕様の例

有向非閉路グラフとして表現したものである. 各タスクには最悪/最良実行時間及び相対デッドラインの情報を付加する. リソース割り当て図とは、各タスクが使用するリソースの指定、及びリソース競合を解決するスケジューリング方式を指定した図である. 図 1 において CPU, MEM, PER i ($i=1,2,3$)はそれぞれ CPU, メモリ, 周辺機器 i ($i=1,2,3$)のリソースを表す.

3. 性能検証手法

性能検証問題は、与えられた性能要求と性能モデルから、性能要求を満たすか否か(Yes/No)を得る問題である. 文献[2]では、与えられた複数タスク動作仕様を、それと等価な振舞いを行いかつスループット要求を満たさなければ特定の動作を行う優先権付きストップウォッチペトリネット(PrSwPN)[4,5]に変換し、スループット要求を満たすか否かの検証を行う手法を提案している. 確定的モデルである PrSwPN を用いるため、システムの最悪時性能を考慮した検証が可能となる. PrSwPN の検証には、既存の検証ツール TINA[6]を使用する.

4. PREM(Predictable Execution Model)

PREM とは、タスクを実行するときその実行過程を実行時間が予測できない compatible interval と実行時間が予測可能な predictable interval に分割し、さらに predictable interval を、メインメモリからのデータがキャッシュ内に配置されるフェーズ(メモリフェーズ), キャッシュされたデータのみがアクセスされるフェーズ(実行フェーズ)の 2 つのフェーズに分割するタスク実行モデルである[3]. PREM の基本的な考え方は、メモリフェーズで CPU はメモリアクセス

A Performance Verification Method for Multitasking Software in Predictable Execution Model under Tightly-Coupled Multi-Processor Environment

Takafumi Nakamura† Akio Nakata†

Department of Systems Engineering, Graduate School of Information Sciences, Hiroshima City University†

をまとめて行い、必要な全てのメモリ内容をキャッシュに格納することで、その後の実行フェーズで一定時間 CPU はメモリアクセスを行うことなく処理の実行ができ、その間周辺機器がメモリへのアクセスを行うことができる、という点にある。このように、CPU と周辺機器によるメモリアクセスを相互スケジュールすることにより、メモリアクセス競合を解決し、競合に起因するタイミング遅延を回避することができる。一定の制約の下で PREM に適合したオブジェクトコードを出力するコンパイラ実装が存在する[3].

5. 提案手法

[2]の手法は、各タスクの実行開始時に必要な全リソースを獲得し、終了時に全リソースを解放するという動作意味に基づいてシステムの振舞いをモデル化し、性能検証を行う。提案手法は、[2]の手法に基づいて次の方法で PREM の動作をモデル化する。リソースとして複数の CPU、単一のメモリ及び周辺機器を、タスクとして CPU を用いて実行されるタスク群と周辺機器によるメモリアクセスタスクを仮定する。CPU を用いて実行される各タスクの実行過程を PREM に基づき 3 種類のサブタスク(*compatible interval* 及び、*predictable interval* のメモリフェーズ・実行フェーズ)に分解する。*compatible interval* とメモリフェーズでは、開始時にメモリと CPU リソース 1 個を獲得し、終了時に解放する。実行フェーズでは、開始時に CPU リソース 1 個を獲得し、終了時に解放するものとする。また、*compatible interval* の実行時間は任意(最悪実行時間と最良実行時間の間で任意に変動)とし、*predictable interval* は、複数タスク動作仕様から PrSwPN への変換ツールの仕様上、メモリフェーズ・実行フェーズとも固定時間で実行されるものとしてモデル化する。最後に周辺機器のメモリアクセスタスクを、メモリリソースおよび周辺機器リソースを獲得して任意時間で実行するものとしてモデル化する。

6. 実験

PREM の有用性検証がスループット性能検証によって可能か否かを特定のタスクモデルに対して確認する事を実験目的とする。図 1 の複数タスク動作仕様の例題を、従来の方針で記述したものと、PREM モデルに従い記述したものの両方でスループット性能検証を行い比較した。PREM モデルで記述したも

表 1. 提案手法による検証結果

スループット要求 (入力/単位時間)	スループット要求を満たすか否か	
	PREM	従来のタスクモデル
1/300	Yes	Yes
1/250	Yes	Yes
1/200	Yes	No

のにおいては、元のタスクの最悪実行時間の 1/3 ずつをメモリフェーズ、実行フェーズ、*compatible interval* に分割した。CPU とメモリのスケジューリング方式は固定優先度(FP)と早い者勝ち(FIFO)の両方の場合について検証した。実験結果を表 1 に示す。表 1 の結果より、提案手法は従来手法よりも厳しいスループット要求を満たすということが確認できた。なお、スケジューリング方式の違いによって検証時間などの差はあるもののスループット性能については同様の結果が得られた。

7. あとがき

本研究では、密結合マルチプロセッサ環境における PREM タスク実行モデルを使用するシステムを検証可能な性能検証手法を提案した。実験的評価により、性能検証において PREM の有用性を検証することが可能であることが分かった。今回はメモリフェーズと実行フェーズの実行時間を固定としたが、本来の PREM では *predictable interval* 内のメモリフェーズと実行フェーズの実行時間の内訳は変動する。従って今後の課題としては、*predictable interval* の固定された実行時間の中における各フェーズの実行時間の変動を表現することなどが挙げられる。

参考文献

- [1]Balsamo, S., Di Marco, A., Inverardi, P. and Simeoni, M.: Model-based Performance Prediction in Software Development: A Survey, *IEEE Trans. Softw. Eng.*, Vol.30, No.5, p.295-310, (2004).
- [2] 百々太市, 中田明夫, “ブリエンティブスケジューリングによりリソースを共有する複数タスク動作仕様の性能検証”, 組込みシステムシンポジウム(ESS2010)論文集, pp. 107-112, 情報処理学会, 2010.
- [3]R. Pellizzoni, E. Betti, S. Bak, G. Yao, J. Criswell, M. Caccamo, and R. Kegley, “A predictable execution model for cots-based embedded systems”, in *Proceedings of the 17th IEEE Real-Time and Embedded Technology and Applications Symposium*, 2011.
- [4]Berthomieu, B., Peres, F., and Vernadat, F., “Model Checking Bounded Prioritized Time Petri Nets”, *Proc. of 5th Int. Symp. on Automated Technology for Verification and Analysis (ATVA 2007)*, LNCS, vol.4762, pp.523-532, 2007.
- [5] B. Berthomieu, D. Lime, O. H. Roux, F. Vernadat, “Reachability Problems and Abstract State Spaces for Time Petri Nets with Stopwatches”. *Journal of Discrete Event Dynamic Systems*, vol.17, pp.133-158, 2007.
- [6]Berthomieu, B. and Vernadat, F.: “Time PetriNets Analysis with TINA”, *Proc. of 3rd Int.Conf. on the Quantitative Evaluation of Systems(QEST 2006)*, IEEE Computer Society Press, (2006).