

マルチプロセッサシステムのヒープメモリ消費量を削減する
リアルタイムスケジューリング手法 LMCLF スケジューリングの換算レート α の提案
組み込みデザイン研究室 新井 諒介 (指導教員 中田 明夫)

1 はじめに

スマートフォン,家電製品,医療機器などの組み込みシステムは大量生産されることが多いため,製造コストの削減は重要な課題であり,メモリ消費量を削減することが,組み込みシステムの開発目標の一つとして挙げられている[1].そこで我々は,マルチタスクシステムのヒープメモリ消費量を動的解析によって削減するスケジューリングアルゴリズム Least Memory Consumption First (LMCF) スケジューリングを提案した[2]. LMCF スケジューリングでは,各タスクの次のステップのヒープメモリの割り当て(消費メモリ増分)は予測可能であると仮定する.消費メモリ増分については,静的コード解析やプロファイリングによって予測する.プロセッサ数を p とすると,次のステップの消費メモリ増分が小さいタスクから p 個選択し,選択したタスクに優先度を付与する. LMCF スケジューリングは,必ずしもメモリ最適なスケジュールを実現するわけではない.一般に,同時起動中のタスク数が少ないほど,メモリ消費量が少なくなると考えられる.このことから,同時起動中のタスク数を可能な限り減らすため,残余実行時間が少ないタスクを優先する方針によって,メモリ消費量のさらなる削減が期待できる.一方で残余実行時間が等しいタスク群に対しては,余裕時間を考慮した方針で優先関係を与えることで,デッドライン制約をより維持しやすくなると考えられる.先行研究では,消費メモリ増分だけでなく,実時間制約を共に考慮した Least Memory, remaining Computation-time, and Laxity First (LMCLF) スケジューリングが提案されている. 比較する値 θ は, $\alpha \times$ 消費メモリ増分 + 残余実行時間 \times 余裕時間で求められるが,先行研究では換算レート α の値がすべて 1 で定義されている.もし,余裕時間がとても大きく,メモリ増分が限りなく小さい場合,時間のみで優先度が決定されてしまう.そこで本研究では,いろい

ろなタスクセットで換算レート α の最良な値を導き出し,その法則性をつかみ,最良な値を提案する.

2 提案手法

LMCLF: 消費メモリ増分だけでなく,実時間制約を共に考慮したスケジューリング手法

$\theta: \alpha \times$ 消費メモリ増分 + 残余実行時間 \times 余裕時間

α : 時間に対するメモリ増分への重みづけの値

分岐: k 時間進んだ時の θ の値が同じになる α の値

ここに本研究の LMCLF のアルゴリズムを示す.

1. 初期状態の分岐の値 x を求める.
2. $\alpha > x$, $\alpha < x$ の時で場合分けして考える.
3. $\alpha > x$ の時の次状態の分岐の値を求める.
4. 必要であれば次々状態の分岐の値も求める.
5. 同じように $\alpha < x$ のも求めていく.
6. 最悪メモリ消費量が最も小さくなる α の範囲が最良の α の値となる.

※このアルゴリズムはまだ検討中なため確定ではない.

従来手法では, α の値がすべて 1 として計算されていたため,もし,時間が大幅に大きくなった場合時間のみで優先度が決定されてしまうことになる.本研究ではどのようなタスクセットでもそのようになることを避けるために,いろいろなタスクセットで α の法則性を見出し,どのようなタスクセットでも対応できるようにしている.

3 実験

実験目的: 最良な換算レート α の模索のため

実験方法: 現段階では 2 タスク 1 プロセッサの環境下でタスクセットの値を変えながらいろいろなパターンで求める.方法としては提案手法の

LMCLF のアルゴリズムに沿って行う.

実験結果：現段階では、タスクセットによって次状態、
次々状態まで求めると最良な α の値の範囲が
求まることが分かった.まだ実験段階なため
検討中である.

考察：検討中である

4 まとめ

本研究では先行研究で提案された LMCLF スケジューリ
ングの換算レート α の最良な値について提案した.
これ以降はまだ検討中である.

and Distributed Systems, Vol. 28, No. 1,

参考文献 [1] R. Zurawski, “Embedded Systems Handbook, Second Edition: Embedded Systems Design and Verification”, CRC Press, 2009. [2] Y. Machigashira and A. Nakata, “An improved LLF scheduling for reducing maximum memory consumption by considering laxity time”, In Proc. of 12th Int. Symp. on Theoretical Aspects of Software Engineering, pp.144–149, IEEE Computer Society Press, 2018. [3] M. L. Dertouzos and A. K. Mok, “Multiprocessor On-Line Scheduling of Hard-Real-Time Tasks”, IEEE Tran. on Software Engineering, Vol.15, No.12, 1989. [4] C. L. Liu and J. W. Layland, “Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment”, Journal of ACM, Vol. 20, No. 1, pp.46–61, 1973. [5] J. Lee, A. Easwaran, and I. Shin, “Laxity Dynamics and LLF Schedulability Analysis on Multiprocessor Platforms”, Real-Time Systems, Vol. 48, Issue 6, pp716–749, 2012. [6] T. P. Baker, “Comparison of Empirical Success Rates of Global vs. Partitioned Fix-Priority and EDF Scheduling for Hard Real Time”, Technical Report TR-050601, Department of Computer Science, Florida State University, pp.1–14, 2005. [7] J. Lee, “Time-Reversibility for Real-Time Scheduling on Multiprocessor Systems”, IEEE Transactions on Parallel