B-012

オーバーヘッドを抑制したマルチプロセッサに対応したスケジューリングアルゴリズムの提案

Proposal of scheduling algorithm with reduces overhead for multi processors

平井 佑樹† 俞 明連†
Hirai Yuki Yoo Myungryun

1. 研究背景

組込みシステムでは、複雑な処理を高速で行うために、高性能かつ省電力なプラットフォームが求められ、マルチプロセッサ技術が利用されている。そのため、マルチプロセッサ環境での組込みリアルタイムシステム向けのリアルタイムスケジューリングが必要となっている。

現在、シングルプロセッサ環境で最適である EDF(Earliest Deadline First)[1] に基づいた EDCL(Earliest Deadline Critical Laxity)[2]、林のアルゴリズム[3]、小林のアルゴリズム[4]、内田のアルゴリズム[5]などの研究が行われている。しかし、デッドラインミスを抑えることと、オーバーヘッドを抑えることが両立することができていない問題がある。

2. 研究目的

本研究では、特にプロセッサ数が多い場合に、スケジュール成功率を維持したまま、コンテキストスイッチの発生回数とスケジューラの起動回数を抑制するアルゴリズムの提案を目的とする。

3. システムモデル

本論文でスケジューリングの対象とするタスクセットは、周期的タスクで構成される。また、システムモデルはマルチプロセッサ環境での研究における一般的なシステムモデル[6]として仮定する。システムは同一性能のプロセッサを M 個持ち、N 個のタスクで構成されるタスクセット$\tau = \tau_1, \tau_2, \tau_3, \ldots, \tau_n$ が与えられる。各タスクτ_i は、最悪実行時間C_iと周期T_iを持ち、タスクの表記はこれらを用いて$\tau_i = (C_i, T_i)$と表す。また、タスクτ_iの利用率(1 つのタスクが実行を完了するまでに必要な CPU の割合) をタスク利用率といい、$U_i = C_i/T_i$ で表す。そして、タスクセットτ に含まれるタスクの利用率の合計を$U(\tau) = \sum_{i=0}^{n} U_i$ と示し、$U(\tau)/M$をシステム利用率とする。

ジョブのデッドラインまでに、実行していなくても良い残り時間を余裕時間という。余裕時間はジョブの緊急度を示す。

4. 従来研究

4.1 EDF

EDF(Earliest Deadline First)[1]は、動的優先度スケジューリングアルゴリズムで、その時点で次のデッドラインが早いタスクに高い優先度を与える手法である。このアルゴリズムでは、スケジューラの起動はジョブの起動時と終了時に行う。

4.2 内田のアルゴリズム

内田のアルゴリズム[5]は EDF を基に改善したアルゴリズムを基としたものである。具体的には「余裕時間がクリティカルのジョブに最高優先度を与える」「デッドラインの最も早いジョブのうち、最小の残り実行時間を持つジョブに準最高優先度を与える」というルールを追加している。内田のアルゴリズムでは、更にタスク利用率が 0.5 を超えている全てのジョブに対して疑似デッドラインを設定している。疑似デッドラインとは、通常のデッドラインに加え、タスクの優先度を決定するために、追加でタスクを分割する疑似的なデッドラインのことである。疑似デッドラインは最悪実行時間・周期ともに 4 分割している。また、スケジューラの起動はジョブの起動時と終了時に行う。

5. 提案手法

基本的に内田のアルゴリズムと同様に動作を行うが、疑似デッドラインを設定する条件を変更した。具体的には、タスク利用率が 0.75 以上のすべてのジョブに対して実行時間・周期を 4 分割する疑似デッドラインを設定し、それ以外のジョブの中で、タスク利用率が 0.5 以上のすべてのジョブに対して実行時間・周期を 2 分割する疑似デッドラインを設定する。

図 1 に提案手法の動作例を示す。

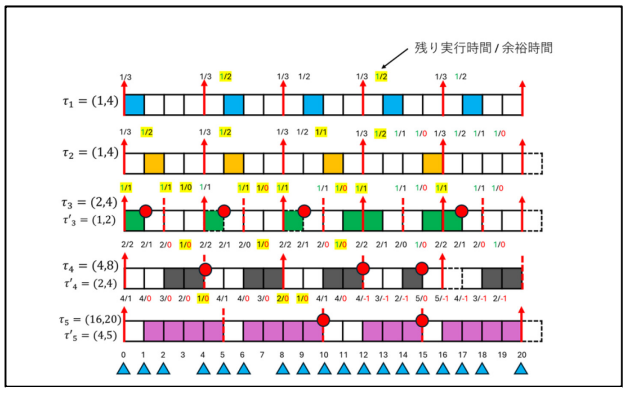

図 1 提案手法の動作例

6. 評価・考察

6.1 シミュレーションについて

シミュレーションはM, U_{total} の 2 つのパラメータを使用する。Mはプロセッサ数、U_{total} は各タスクセットのタスク利用率の合計である。また、U_{total}/M をシステム利用率と定義する。

システム利用率が 1%から 100%まで 1%ごととなるように、U_{total}の値を変えながら、シミュレーションを行う。

また、利用率ごとに 100 個のタスクセットを生成した。プロセッサ数 M は 2,4,8,16,32,64,128 の 7 通りとする。タスクセット生成時、タスクセットのタスク利用率の合計U_τが$U_\tau < U_{total}$である限り、タスクを生成して追加していく。生成される各タスクの実行時間はタスク利用率U_iは$0.0 < U_i <= 1.0$の範囲となるように、ランダムに設定する。この時、タスク利用率の合計が予め設定していたU_{total}を超えないように考慮する。また、タスクの周期は[100, 200, 400, 800, 1600] の中からランダムに決定する。評価するものは、デッドラインオーバー量、コンテキストスイッチの回数、スケジューラの起動回数の 3 項目とする。なお、本原稿では、プロセッサ数が 16 の時のデータのみ掲載する。

6.2 デッドラインオーバー量

デッドラインオーバー量の比較結果を図 2 に示す。これらの結果から、内田の手法と比較すると提案手法は少しデッドラインオーバー量が増えてしまっているが、EDF と比較するとデッドラインオーバー量を抑えることができている。これは、タスクの分割数を削減したことによって、一部ジョブにプロセッサが割り当てられづらくなり、時間内に完了しなかったジョブが増えてしまったからである。

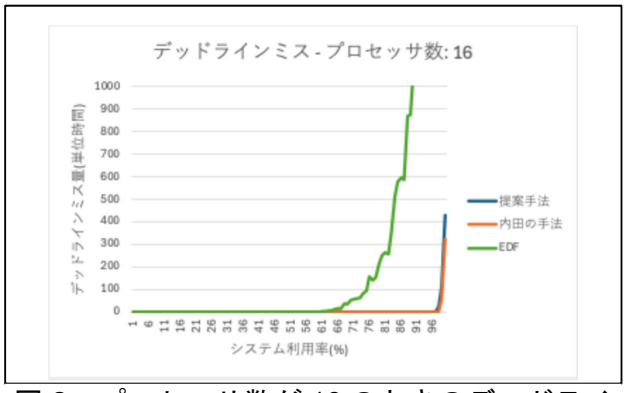

図 2　プロセッサ数が 16 のときのデッドラインオーバー量

6.3 コンテキストスイッチ発生回数

コンテキストスイッチ発生回数の比較結果を図 3 に示す。これらの結果から、内田の手法と比較すると、コンテキストスイッチの発生回数を抑えることができている。これは、タスク利用率が 0.5 以上のタスクに対して、内田の手法では 4 分割していたところを、2 分割に減らしたことで、一定条件を満たしたタスクの分割数が減り、タスクの実行が途中で中断される回数が減少したためである。

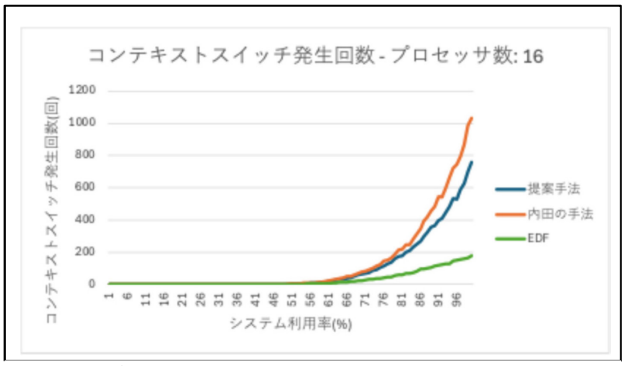

図 3　プロセッサ数が 16 のときのコンテキストスイッチ発生回数

6.4 スケジューラ起動回数

スケジューラ起動回数の比較結果を図 4 に示す。これらの結果から、内田の手法と比較すると、スケジューラの起動回数を抑えることができている。これは、コンテキストスイッチの発生回数の場合と同様に、タスク利用率が 0.5 以上のタスクに対して、内田の手法では疑似デッドラインで 4 分割していたところを、2 分割に減らしたことで、一定条件を満たしたタスクの分割数が減り、タスクの途中で起動していたスケジューラが起動しなくなった箇所があるためである。

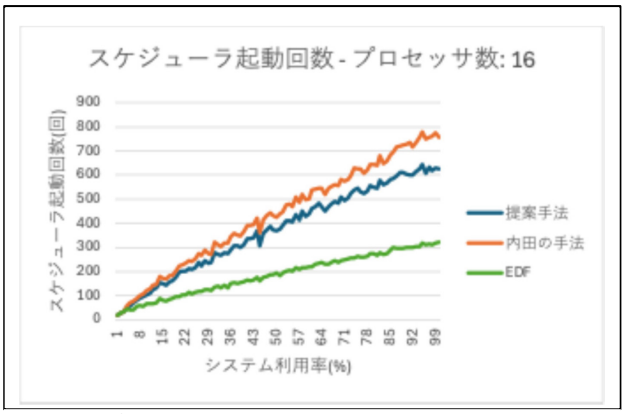

図 4　プロセッサ数が 16 のときのスケジューラ起動回数

7. 結論

本研究では、複数のタスクに対して疑似デッドラインを設ける従来研究を元に、疑似デッドラインの設定方法を調整することで、スケジュール成功率を大きく下げずに、コンテキストスイッチの発生回数とスケジューラの起動回数を削減するリアルタイムスケジューリングアルゴリズムを提案した。特に、プロセッサ数が多く、システム利用率も高いときに、内田の手法と比較するとコンテキストスイッチの発生回数とスケジューラの起動回数を削減し、スケジュール成功率を維持することができていると言える。

参考文献

[1] C.L.Liu ・ J.W.Layland. ’ Scheduling Algorithms for Multiprogramming in a Hard-RealTime Environment’, Journal of the ACM, Volume 20, Issue 1, 1973.
[2] S.Kato and N.Yamasaki ’Global EDF-based scheduling with laxity-driven priority promotion’, Journal of systems Architecture, Vol.57, No5, pp.498-571(2011).
[3] 林竜太, 兪明連, 横山孝典, ‘EDF を基にしたスケジューリングアルゴリズムの提案’, 電子情報通信学会技術研究報,Vol.115,No.374,pp.27-32(2015).
[4] 小林大起,’疑似デッドラインによるスケジュール成功率が高いアルゴリズムの提案’, 東京都市大学卒業論文,2019.
[5] 内田紘太, ‘疑似デッドラインを用いてデッドラインミスを削減するスケジューリングアルゴリズム’, 東京都市大学卒業論文,2024.
[6] 武田瑛, 船岡健司, 加藤真平, 山崎信行,”マルチプロセッサにおけるグローバル RM に基づくリアルタイムスケジューリングアルゴリズム”, 信学技報 ,vol.107,no.559,DC2007-116,pp.191-196,2008.