

クリティカルの基準変更による動的スケジューリング アルゴリズムの性能分析及びスケジューリング可能性解析

佐々木 理成[†] 兪 明連[†] 横山 孝典[†]

東京都市大学[‡]

1. 研究背景

近年、組込みリアルタイムシステムではマルチプロセッサ技術の利用が一般化してきている。しかし、従来のシングルプロセッサ環境下で最適な EDF (Earliest Deadline First) [1] を基にしたマルチプロセッサ向けのアルゴリズムは多く提案されているが、確立には至っていない。本研究では、EDF にクリティカルタスクルールを付加した EDCL (Earliest Deadline Critical Laxity) [2] の優先度判定方法であるクリティカルの基準を変更し、その性能分析やスケジューリング可能性解析から適切なクリティカルな基準について探る。

2. システムモデル

マルチプロセッサでのリアルタイムスケジューリングにおける一般的なシステムモデルを想定する。システムは M 個のプロセッサを持ち、 N 個の周期的タスクから構成されるタスクセット $\tau = \{\tau_1, \tau_2, \dots, \tau_N\}$ が与えられる。各タスク τ_i は $\tau_i = (C_i, T_i)$ と定義され、 C_i , T_i はそれぞれ τ_i の最悪実行時間と周期を表す。また、 $U_i = C_i/T_i$ をタスク τ_i の利用率とし、 $U(\tau)/M$ をシステム利用率と定義する。各タスク τ_i は周期 T_i の間隔でジョブを生成し、タスク τ_i が生成する k 番目のジョブを τ_{ik} と表す。 τ_{ik} は時刻 r_{ik} でリリースされ、デッドライン d_{ik} は次のジョブのリリース時刻と一致する。ある時刻 t におけるジョブ τ_{ik} の残り実行時間を $C_{ik}(t)$ とするとき、 τ_{ik} の余裕時間 $L_{ik}(t)$ を $L_{ik}(t) = d_{ik} - t - C_{ik}(t)$ と定義する。

3. 従来研究

EDF (Earliest Deadline First)

EDF はデッドラインの早いタスクに高い優先度を与えるアルゴリズムであり、マルチプロセッサ環境下ではスケジューリング成功率が低いという問題点を持つ。スケジューラの起動はタスクの起動時と完了時のみ行う。

EDCL (Earliest Deadline Critical Laxity)

EDCL は EDF においてクリティカルなタスクに最高優先度を与えるアルゴリズムである。クリティカルなタスクとは、EDF でスケジューリングする場合に実行されるタスクの中の最小の残り実行時間よりも、小さい余裕時間を持つタスクのことをいう。スケジューラの起動はタスクの起動時と完了時のみである。EDF に比べスケジューリング成功率は向上したが、依然として低い。

4. 分析モデル

以下に EDCL におけるクリティカルの基準を変更した 6 つのモデルを示す。任意のスケジューラの起動時刻 t_s において、EDF でソートした場合に実行されるタスクの中で最小の残り実行時間を $e_{min}(t_s)$ とし、タスク τ_i の残り実行時間を $C_i(t_s)$ 、余裕時間を $L_i(t_s)$ とする。

- モデル 1: $e_{min}(t_s) > L_i(t_s)$ (元の EDCL)
- モデル 2: $[e_{min}(t_s) \times 1/2] > L_i(t_s)$
- モデル 3: $[e_{min}(t_s) \times 3/2] > L_i(t_s)$
- モデル 4: $[C_i(t_s) \times 1/2] > L_i(t_s)$
- モデル 5: $[C_i(t_s) \times 1/4] > L_i(t_s)$
- モデル 6: $[C_i(t_s) \times 3/4] > L_i(t_s)$

モデル 2/3 はそれぞれ元のモデル 1 に比べ、クリティカルになりにくく/なりやすく設定する。モデル 4 以降は $e_{min}(t_s)$ ではなく $C_i(t_s)$ を用いて周期の長いタスクでも比較的余裕をもって実行されやすいようにしている。

5. 評価

モデル 1~6 のアルゴリズムについてシミュレーションにより評価を行う。システム利用率 70% から 100% の範囲で 5% おきに一様乱数を用いて生成したタスクセットをそれぞれ 1000 個投入し、各アルゴリズムでシミュレーションをする。プロセッサ数は 4, 8, 16 の 3 通りとし、各タスク利用率の範囲は $[0.01, 1.0]$ とする。評価項目はスケジューリング成功率、平均コンテキストスイッチ回数、平均スケジューラ起動回数、平均デッドラインオーバー量の 4 つである。図 1, 2 にプロセッサ数 16 の場合のスケジューリング成功率と平均コンテキストスイッチ回数の結果を示す。

The performance analysis and schedulability test on dynamic scheduling algorithm by critical criteria

[†] Risei Sasaki, Myungryun Yoo, Takanori Yokoyama

[‡] Tokyo City University

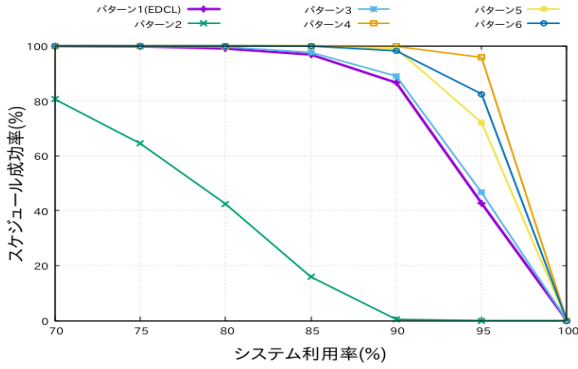


図 1. スケジュール成功率

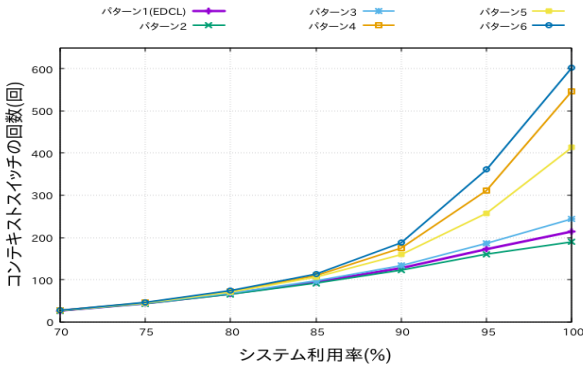


図 2. 平均コンテキストスイッチ回数

図 1 より, $e_{min}(t_s)$ を用いるモデル 1~3 より $C_i(t_s)$ を用いるモデル 4~6 の方がスケジュール成功率は高くなった. これは, 周期の長いタスクが高負荷な状態でもクリティカルになりやすいためだと考えられる. 一方で, 図 2 のコンテキストスイッチ回数については, 比較的クリティカルになりやすいモデル 4~6 で優先度の変更が頻発した. よって, スケジュール成功率の面では, モデル 4 がクリティカルの基準に最も適しているが, コンテキストスイッチの回数の面で問題がある.

6. スケジュール可能性解析

本研究では, 反応時間解析 (Response-time Analysis : RTA) [3]を用いて可能性解析を行う. 解析に用いる 2 つの用語を以下のように定義し, 詳細な証明は[3]に基づく.

- ・干渉長 $I_k^i[a, b]$: 区間 $[a, b]$ でタスク τ_k が τ_i によりブロックされ実行できない区間の合計長
- ・仕事量 $W_i[a, b]$: 区間 $[a, b]$ でタスク τ_i が実行しなければならない実行量

6 つのモデルにおいて, 仕事量が最大となる状況を想定し, それを図 3 に示す.

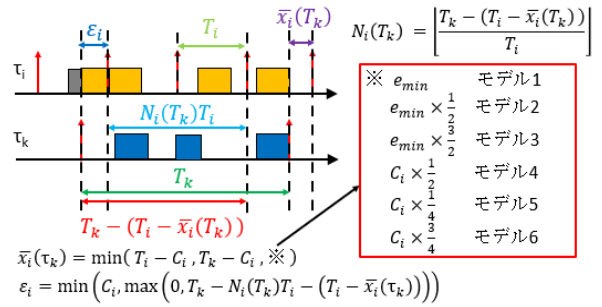


図 3. 仕事量が最大となる状況

図 3 より, タスク τ_i の仕事量の最大値が式 (6. 1) で求まり, それを用いるとタスク τ_i の干渉長の上限値が式 (6. 2) で表せる.

$$W_i^{ub}(T_k) = \min(T_k, (N_i(T_k) + 1)C_i) + \min(C_i, \max(0, T_k - N_i(T_k)T_i - (T_i - \bar{x}_i(\tau_k)))) \quad (6. 1)$$

$$\hat{I}_k^i(R_k^{ub}) = \min(\omega_i^{ub}(R_k^{ub}), W_i^{ub}(T_k), R_k^{ub} - C_k + 1) \quad (6. 2)$$

よって, タスク τ_k の反応時間の上限値 R_k^{ub} は $R_k^{ub} = C_k$ から始まる式 (6. 3) の不動点反復法で求められる.

$$R_k^{ub} \leftarrow C_k + \left\lceil \frac{1}{M} \sum_{i \neq k} \hat{I}_k^i(R_k^{ub}) \right\rceil \quad (6. 3)$$

すなわち, 全てのタスク $\tau_k \in \tau$ が式 (6. 4) を満たせば, スケジュール可能である.

$$L_k^{lb} = T_k - R_k^{ub} \geq 0 \quad (6. 4)$$

7. 結論

本研究では, EDCL の優先度判定方法であるクリティカルの基準を変更し適切なクリティカルの基準について議論した. 今後は分析したモデルのプログラムにスケジュール可能性解析を組みこみ, より厳密な評価を行う予定である.

謝辞

本研究は JSPS 科研費 20K11755 の助成を受けたものです.

参考文献

- [1] C.L.Liu and J.W.Layland: "Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment", Journal of the ACM, Vol.20, No.1, pp.46-67, 1973.
- [2] S.Kato and N.Yamazaki: "Global EDF-based scheduling with laxity-driven priority promotion", Journal of systems Architecture, Vol.57, No.5, pp.498-517, 2011.
- [3] M Bertogna and M Cirinei: "Response-Time Analysis for globally scheduled Symmetric Multiprocessor Platforms", 28th IEEE International Real-Time Systems Symposium, pp.149-160, 2007.