

連続メディア処理における時間制約と通信遅延に適應するタスクスケジューリング

滝沢 泰久,[†] 芝 公仁,^{††} 大久保 英嗣,^{†††}

[†] (株) ATR 環境適応通信研究所
^{††} 立命館大学大学院理工学研究科
^{†††} 立命館大学理工学部

連続メディアを扱うアプリケーションを周期タスクとしてスケジュールする場合、その時間制約とタスク間の依存関係を満たすためにリアルタイムスケジューラとリアルタイム同期プロトコルとを組み合わせる方式が有効とされている。しかし、この組み合わせ方式は、予測が困難な変動する環境には適用できない。また、リアルタイム同期プロトコルは密結合による共有メモリモデルを前提としているため、疎結合モデルによるメッセージ通信にはそのまま利用できない。本論文では、単一プロセッサ上の任意の周期タスク間でメッセージ通信する動的な環境において、Parallel Distributed Processing モデルと熱力学的モデルを用いることにより、その時間制約と通信依存関係に適應するタスクスケジューリングポリシーを提案し、その性能評価について述べる。

キーワード スケジューリングポリシー リアルタイムシステム マルチメディア

An Adaptable Task Scheduling for Timing Constraints and Communication Delays on Continuous Media Processing

Yasuhisa Takizawa,[†] Masahito Shiba,^{††} Eiji Okubo,^{†††}

[†] ATR Adaptive Communication Research Laboratories
^{††} Graduate School of Science and Engineering, Ritsmeikan Univ.
^{†††} Department of Computer Science, Faculty of Science and Engineering, Ritsmeikan Univ.

Combination schemes of real-time scheduler and synchronization protocols are useful for scheduling periodic tasks which manipulate continuous media. However, these schemes can not be applied to dynamic environments. Furthermore, these schemes can not be applied to the message passing, because shared memories can not be used. In this paper, a new scheduling policy based on Parallel Distributed Processing model and thermodynamics model, which is adaptable for tasks with timing constraints and communication constraints is proposed, and its performance evaluation is presented.

Keywords Scheduling Policy Real-Time System MultiMedia

1. はじめに

マイクロプロセッサの急速な進歩により、パーソナルコンピュータ（以降PC）やワークステーション（以降WS）上で動画や音声に代表される連続メディアを処理するアプリケーション（以降連続メディアAP）が数多く出現している。連続メディアAP[4]は、その処理するメディアの特性上、周期タスクとしての時間制約を持つ。マルチメディアシステムでは、周期タスクが複数実行され、かつ互いに通信を行う場合が多い。そのため、PCやWS上で周期タスクをスケジューリングする場合、リアルタイムスケジューラとリアルタイム同

期プロトコルとを組み合わせる方式が多く採用されている。しかし、この組み合わせ方式は、任意のタスクが実行され、タスク間の通信や入出力処理が行われる動的な実行環境では、十分に機能しない。

一方、PCやWS上のオペレーティングシステム（以降OS）は、必要最小限のサービスを提供するマイクロカーネルとマイクロカーネル上で動作するシステムサーバプロセス群により構成されている。この構成では、アプリケーションのシステムサービス要求はシステムサーバプロセスとの通信に置き換えられる。また、アプリケーションタスク間においても同様の方式が多く利用されている。すなわち、タスク間のデータ交換

方式は、メッセージ通信方式が大部分を占める。従って、このようなデータ交換環境では、リアルタイム同期プロトコルで用いる排他制御方式は利用できない。

本稿では、連続メディア処理において以上の問題点を解決するために、

- ・スケジュール環境は、事前に予測が困難な動的な環境である。
- ・スケジュール問題は、タスクの時間制約と通信依存関係から構成される。
- ・いくつかのタスクは入出力処理を行う。
- ・タスク間のデータ交換方式は、疎結合モデルによるメッセージ通信方式である。

の4つを前提として、Parallel Distributed Processingモデル（以降PDPモデル）[5]と熱力学的モデル[3]を用いた適応機能を従来の組み合わせ方式に付加することにより、高いスケジュール可能性を導く適応的スケジューリングポリシ Adaptive Deadline Modification（以降ADM）を提案する。また、そのADMの性能評価についても述べる。

以下、2章で新たなポリシADMを提案し、3章でADMの性能評価を述べ、その有効性を議論する。

2. 提案スケジューリングポリシADM

2.1 周期タスクモデル

連続メディア処理における周期タスクモデルを次のように定義する。

(1) 横取り可能

横取り可能なタスクは、実行中に、より高い優先度を持つタスクが到着した場合、実行権をより高い優先度を持つタスクに引き渡すことが可能なタスクである。

(2) 時間制約

連続メディア処理の周期タスクは、処理完了時刻には許容幅があるソフトリアルタイムタスクとして考える。従って、従来の周期タスクモデルにデッドラインの許容幅を追加し、各時間属性を図1のように定める。

(3) メッセージ通信によるデータ交換

マイクロカーネル構成において、タスク間のデータ

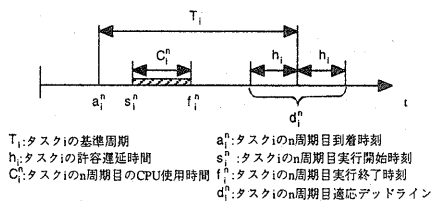


図1 周期タスクの時間制約

交換はメッセージ通信により実施されると考える。

(4) ブロッキング時間

周期タスクが到着し完了するまでの期間で、自分より優先度の低いタスクの実行時間とアイドル時間の和をブロッキング時間とする。

2.2 ADMの構成

ADMは、従来のリアルタイムスケジューラとリアルタイム同期プロトコルの組み合わせ方式（以降RTスケジュール機能）に変動環境に対応する適応機能を付加した構成となる（図2）。

2.3 RTスケジュール機能

リアルタイムスケジューラは、最適なスケジューラである Earliest Deadline First（以降EDF）[2]を用いる。

また、リアルタイム同期プロトコルは事前情報が不要な Priority Inheritance Protocol（以降PIP）[1]を用いる。しかし、前述のようにリアルタイム同期プロトコルは、メッセージ通信によるデータ交換環境にそのまま適用できないので、メッセージ通信機能にPIP機能を追加した Real-Time IPC（以降RT-IPC）[6]を用いる。

2.4 適応機能

メッセージ通信によるデータ交換環境では、資源競合の状況は各タスクのブロッキング時間として現れる。適応機能は、計測されたブロッキング時間とタスク間の通信関係から、ブロッキング時間を抑制するタスク優先度の修正を時間制約の範囲内で行う。これにより、高いスケジュール可能性を導く。

(1) タスク間通信におけるブロッキング時間

送信タスクsと受信タスクrの2つのタスクが通信する場合のブロッキング時間について考える。

送信側タスクsにブロッキング時間が発生する場合、そのブロッキング時間は、式(1)となる。

ただし、 B_s^n は送信側タスクsのn周期目のブロック時間、 $C_{s,send}^n$ は送信側タスクsのn周期目での送信開始時刻

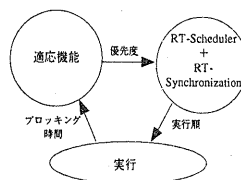


図2 提案ポリシADMの構成

$$B_s^n = \begin{cases} (s_r^k + C_{r,rcv}^k) - (s_s^n + C_{s,snd}^n) & d_s^n < d_r^k \\ s_r^k - (s_s^n + C_{s,snd}^n) & d_s^n \geq d_r^k \end{cases} \quad (1)$$

までの実行時間、 $C_{r,rcv}^k$ は受信側タスク r の k 周期目での受信開始時刻までの実行時間、 d_r^k は受信側タスク r の k 周期目のデッドライン時刻、 d_s^n は送信側タスク s の n 周期目のデッドライン時刻である。

一方、受信側タスク r にブロッキング時間が発生する場合、そのブロッキング時間は、次式となる。

$$B_r^k = \begin{cases} s_s^n - (s_r^k + C_{r,rcv}^k) & d_r^k \geq d_s^n \\ (s_s^n + C_{s,snd}^n) - (s_r^k + C_{r,rcv}^k) & d_r^k < d_s^n \end{cases} \quad (2)$$

ただし、 B_r^k は受信側タスク r の k 周期目のブロッキング時間である。

(2) ブロッキング時間を減少させるデッドライン

ブロッキング時間は、式(1)(2)より、通信するタスクの実行開始時刻に依存する。そこで、ブロッキング時間を減少させるように実行開始時刻を制御するため、デッドラインを修正する。修正されたデッドラインを適応デッドラインと呼び、次のように算出する。

- ・最初の周期では、ブロッキング時間は不明であるため、互いに通信するタスクの実行開始時刻を時間制約範囲内で、可能な限り近付ける適応デッドラインを算出する。
- ・2周期目以降は、前周期の実行の結果生じたブロッキング時間に応じて適応デッドラインを算出する。すなわち、ブロッキング時間がある場合は、開始時刻が早いと判断し、実行開始時刻を遅らす適応デッドラインを算出する。ブロッキング時間がない場合は、開始時刻が遅いと判断し、実行開始時刻を早くする適応デッドラインを算出する。

(3) PDPモデルの応用

実際の実行環境では上記の通信関係が複数混在する。すなわち、扱う問題は、同時多重制約問題である。

ADMは、多重制約問題を処理するために、認知科学のPDPモデルを応用する。PDPモデルは、相互に関連する複数の制約に対して充足度の高い状態を算出する処理モデルである。このPDPモデルを、タスク間の通信関係と計測されたブロッキング時間から次のような制約に対して動作させる。

- ・通信するタスク同士は協調して、重要度を上げ下げする。
- ・通信しないタスク同士は抑制して、一方が重要度が高い場合は低く、一方が重要度が低い場合は高くする。
- ・ブロッキング時間がある場合は、重要度を下げる。

・ブロッキング時間がない場合は、重要度を上げる。これにより、上記制約を充足する各タスクの重要度を探し出し、この重要度と時間制約から適応デッドラインを算出する。

(4) 熱力学モデルの付加

動的なタスク実行環境において、ブロッキング時間は常に変化する。PDPモデルは、この変化に応じて、1つの状態に静止することなく、いくつかの制約充足度の高い状態間を移動する必要がある。このため、PDPモデルに熱力学モデルを付加し、温度による物質の熱揺動をPDPモデルの処理で擬似する。すなわち、充足度の高い状態から離れた場合は温度を高くし、PDPモデルの処理動作を大きく振動させ新しい状態を見つける可能性を高める。一方、充足度の高い状態の近傍にある場合は温度を低くし、PDPモデルの処理動作を現在の状態の近傍で小さく振動させ、その状態を維持する。

この温度による熱揺動制御により、PDPモデルを変動する環境に連続的に動作させる。

3. 性能評価

ADMを分散OS Solelc[7]に実装し、性能評価を行った。プロセッサはPentium-S 200MHzを使用し、またスケジューリング分解能は1 msecとした。

3.1 評価方法と評価タスクセット

評価はメッセージの到着順に処理するFIFO、メッセージの優先度順に処理するPRIQおよびメッセージの優先度順に処理し、受信タスクの優先度をメッセージキュー内の最も高い優先度にするPIPの3つのRT-IPCに適応機能を付加した場合と付加しない場合の合計6つのケースを比較する。

(1) 評価タスクセット1

評価タスクセット1は、クライアント/サーバモデルに基づく要求時処理型である。クライアントの各周期で実行時間はランダムに変動させる。また、サーバタスクの実行時間および擬似ドライバでの遅延時間もクライアントの実行時間に連動し変動させる。

各タスクの通信関係と時間制約を図3に示す。図中の{}はタスクの時間制約を示し、左から基準周期、許容遅延時間、起動時刻、平均実行時間、最小実行時間、最大実行時間を示す。単位はすべてmsである。

(2) 評価タスクセット2

評価タスクセット2は、パイプラインモデルによる前処理型/予約処理型である。各タスクの通信関係と

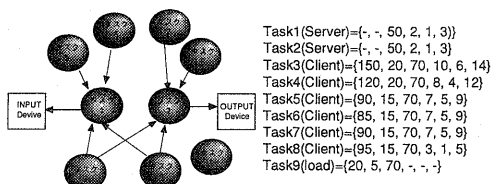


図3 評価タスクセット1の通信関係と時間制約

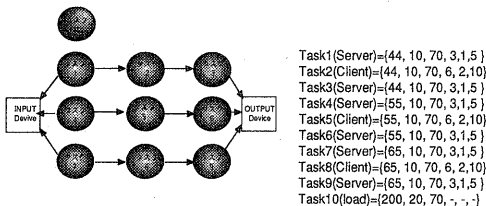


図4 評価タスクセット2の通信関係と時間制約

時間制約を図4に示す。

3.2 スケジューリング成功率

各評価タスクセットに関して、入出力遅延状況に応じて周期タスクのスケジューリング成功率の変動を前述の6つのケースについて比較評価した。

(1) 評価タスクセット1の結果

評価タスクセット1の結果を図5に示す。RT-IPCの種類に関わらず、適応機能を付加したケースがスケジューリング成功率が良い結果となっている。

(2) 評価タスクセット2の結果

評価タスク2の結果を図6に示す。スケジューリング成功率はRT-IPCの種類に全く依存せず、適応機能を用いたケースで常に高い。

以上の(1)と(2)結果から、適応機能は、従来の方式では考慮されていない入出力遅延やタスク間通信による競合に対して有効に機能し、高いスケジュール可能性を導くことが分かる。

4. おわりに

本稿では、従来方式において問題であった1) スケジュール環境が事前に予測が困難な動的な環境である、2) スケジュール問題がタスクの時間制約と通信依存関係により構成される、3) いくつかのタスクは入出力処理を行う、4) タスク間のデータ交換方式はメッセージ通信方式である、といった4つの問題を解決するADMを提案し、さらに、ADMは高いスケジューリング可能性を導くポリシであることを示した。

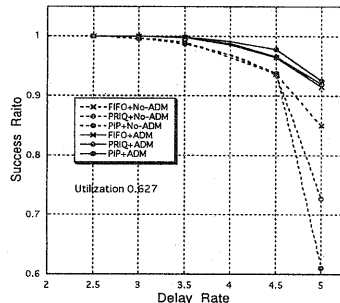


図5 評価タスクセット1の遅延状況に応じたスケジューリング成功率

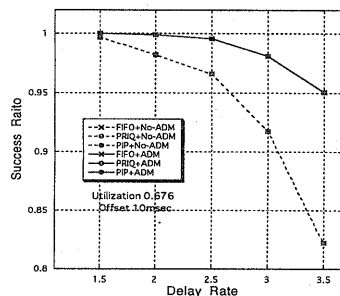


図6 評価タスクセット2の遅延状況に応じたスケジューリング成功率

参考文献

- [1] L. Sha, R. Rajikumar, and J. P. Lehoczky: Priority Inheritance Protocols: An Approach to Real-Time Synchronization, IEEE Trans. on Computer, Vol.39, No.9, pp 1175-1185 (1990).
- [2] C. L. Liu and J. W. Layland: Scheduling algorithms for multiprogramming in a hard real time environment, J. ACM, Vol.20, No.1, pp.46-61 (1973).
- [3] R. Yavatkar and K. Lakshman: Optimization by Simulated Annealing, Science, Vol.220, pp. 671-680 (1983).
- [4] B. Furth: Multimedia Systems: An Overview, IEEE Multimedia, Vol.1, No.1, pp. 47-59 (1994).
- [5] D. E. Rumelhart, J. L. McClelland and the PDP Research Group: PARALLEL DISTRIBUTED PROCESSING, The MIT Press (1986).
- [6] H. Tokuda, T. Nakajima and H. Tokuda: RT-IPC An IPC Extension for Real-Time Mach, In the Proceedings of 2nd Microkernel and Other kernel Architecture, USENIX (1993)
- [7] 芝公仁, 大久保英嗣: 分散オペレーティングシステム Solelc の構成, 情報処理学会研究会報告 2000-OS-84, pp.237-244 (2000).