

周期タスクと非周期タスクが混在する 分散制御システムのためのリアルタイム OS

佐々木智礼[†] 横山孝典[†] 兪明連[†]

東京都市大学[‡]

1. はじめに

分散制御システムはハードリアルタイムシステムであり、制御処理における遅延やジッタは制御性能の低下の原因となるため、デッドラインを守るとともにジッタの少ない分散処理環境が求められる。そこで我々は、通信時間の変動するネットワークを用いたシステムでも高いリアルタイム性を実現することを目的とした分散処理環境を提案した[1]。本分散処理環境は、入出力処理を行うタスクを物理時間に基づいて、それ以外のタスクを論理時間に基づいて管理することで、通信時間の変動により生じる遅延やジッタを許容可能とした。さらに、固定優先度のみでなく、EDF(Earliest Deadline First)スケジューリング[2]も可能なリアルタイム OSを導入することで、タスクの応答時間を短縮可能とした。しかし、これまでは周期タスクのみを対象としていたため、対応できるアプリケーションに制限があった。

本論文では、様々なアプリケーションに対応するために、周期タスクのみではなく非周期タスクをも扱うことが可能な分散処理環境向けのリアルタイム OSを提案する。

2. 分散処理環境

2.1 論理時間を導入した時間駆動分散処理環境

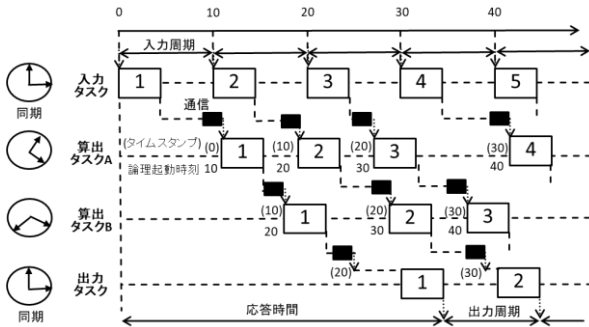


図 1: 論理時間を導入した時間駆動分散処理の動作例

図 1 に本分散処理環境を用いたシステムの動作例を示す。この例では 4 つのノード上にそれぞれ入力タスク、算出タスク A、算出タスク B、出力タスクが分散配置されている。タスクの周期は全て 10 であり、入出力タスクは、実時間(物理時間)に同期した時間駆動(Time-Triggered)アーキテクチャに基づ

いた動作をすることで、ジッタのない処理が行われる。一方、算出タスクは仮想的な時間である論理時間に基づいて管理することで、通信時間の変動を許容している。図 1 を見るとわかるように、物理時間上は、通信時間の変動により算出タスクにジッタが発生しているが、論理時間上はジッタが発生していないものとして扱うことで、時間駆動アーキテクチャと同等の動作を実現している。本分散処理環境では論理時間の起動時刻を論理起動時刻と呼び、各タスクのジョブがメッセージを送信するときに、そのジョブの論理起動時刻をタイムスタンプとして付加することで、ノード間で論理時間に基づいたタスク管理を可能としている。

2.2 スケジューリング

本分散処理環境のリアルタイム OS は、入出力タスクは固定優先度で、算出タスクは EDF でスケジューリングしている。しかしそのまま非周期タスクを扱おうとすると、非周期タスクの優先度を周期タスクの優先度より低く設定する必要があり、応答時間が長くなるという問題が発生する。

この問題を図 2 を用いて説明する。この例では同一のノード上に周期タスクと非周期タスクが 1 つずつ配置されており、周期タスクの周期は 10、実行時間は 4 である。周期タスクの 2 番目のジョブの実行中に非周期タスクが起動されるが、周期タスクが優先されるためプリエンプションは発生しない。そして周期タスクの実行終了後、非周期タスクが実行を開始するが、その実行中に周期タスクが起動されるとプリエンプションが発生し、非周期タスクの実行は中断される。このように通常の EDF スケジューリングでは、非周期タスクの応答時間が長くなる。

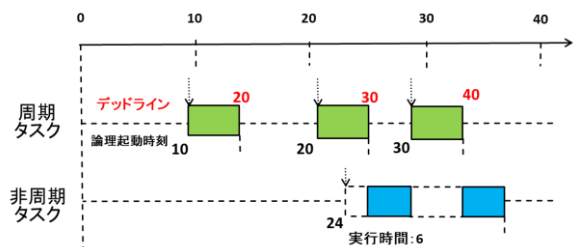


図 2: EDF スケジューリングに基づく動作例

3. スケジューリング方式

3.1 方針

本論文では、通常の EDF スケジューリングに比べて、非周期タスクの応答時間を短縮できるスケジューリングアルゴリズムである TBS(Total

A Real-Time Operating System for Distributed Control Systems with Periodic Tasks and Aperiodic Tasks

[†]Tomonori Sasaki, Takanori Yokoyama and Myungryun Yoo
[‡]Tokyo City University

Bandwidth Server)[3]を導入し、それを論理時間に基づくタスク管理を可能とするよう拡張する。これにより、論理時間を導入した時間駆動分散処理環境において、周期タスクのデッドラインを守りながら非周期タスクの応答時間を短縮することができる。

3.2 論理時間に基づく TBS

TBS では周期タスク、非周期タスクにそれぞれ式(1)、(2)に示す絶対デッドラインを与える。

$$d_i = r_i + t \quad (1)$$

$$d_k = \max(r_k, d_{k-1}) + \frac{C_k}{U_s} \quad (2)$$

d は絶対デッドライン、 r は起動時刻、 t は相対デッドライン、 C は実行時間で添字 i, k はそれぞれ i, k 番目のジョブを示す。また、 U_s は非周期タスクに与えるCPU使用率(バンド幅)である。そして上式で求めたデッドラインに基づいて EDF でスケジューリングする。ただし、提案するスケジューリング機構では、論理時間に基づいたタスク管理を行うため、起動時刻には論理起動時刻を用いる。

図 3 に論理時間に基づく TBS の動作例を示す。図 2 と同様に周期タスクの周期は 10、実行時間は 4 で、周期タスク 2 番目のジョブの実行中に非周期タスクが起動され、式(2)によりデッドラインが与えられる。この時は、周期タスクの方がデッドラインが近いため、プリエンプションは発生しない。その後、周期タスクの 2 番目のジョブの実行終了後に非周期タスクは実行を開始し、その実行中に周期タスク 3 番目のジョブが起動されるが、この時は非周期タスクの方がデッドラインが近いため、プリエンプションは発生しない。このように論理時間に基づく TBS により、応答時間を短縮することができる。

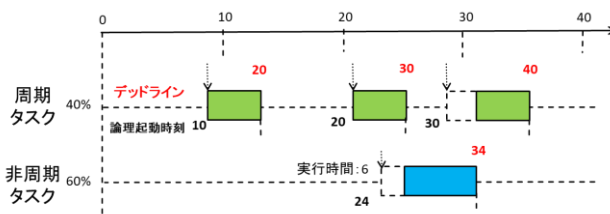


図 3 : 論理時間に基づく TBS の動作例

4. リアルタイム OS

4.1 スケジューリング機構

開発したリアルタイム OS のスケジューリング機構を図 4 に示す。レディキューには実行可能状態のタスクを格納するが、先頭にいくほどデッドラインが近くなるように格納する。次にスケジューリング機構の動作について説明する。休止状態のタスクの起動要求が発生すると、式(1)、(2)によりそのタスクのデッドラインを算出するとともに、実行可能状態へと遷移させる。そして実行中のタスクとデッドラインを比較し、実行中のタスクよりもデッドラインが近い場合は、実行中のタスクをプリエンプトしレディキューの先頭に入れ、起動したタスクを実行

する。実行中のタスクの方がデッドラインが近い場合は、レディキュー中のタスクのデッドラインと比較しながら、デッドラインの近い順に並ぶようにレディキューに格納する。また実行中のタスクが終了した場合は、そのタスクを休止状態にし、レディキュー中の先頭のタスクを実行する。

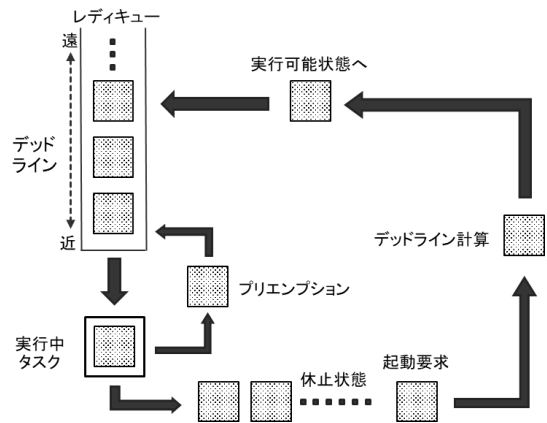


図 4 : スケジューリング機構

4.2 ベースとするリアルタイム OS

本リアルタイム OS は OSEK OS 仕様[4]の TOPPERS/ATK1[5]をベースに、その固定優先度に基づくスケジューリング機構を、提案したスケジューリング機構に置き換えることで実現する。実装にはマイクロコントローラ H8S/2638F を搭載した評価ボードを用いた。H8S/2638F は 256KB の ROM、16KB の RAM を内蔵しており、クロック周波数は 20MHz である。

5. おわりに

周期タスクのみではなく非周期タスクも扱える分散処理環境を実現するために、論理時間に基づく TBS スケジューリング機構を提案するとともに、リアルタイム OS を実装した。本リアルタイム OS を用いることで非周期タスクの応答時間を短縮できる。

本研究で使用した TOPPERS/ATK1 の開発者に感謝する。本研究は JSPS 科研費 18K11225 の助成を受けたものである。

参考文献

- [1]. Ayumu Ichimura, Takanori Yokoyama and Myungryun Yoo :A Time-Triggered Distributed Computing Environment for Cyber-Physical Systems Based on Physical Time and Logical Time, Proceedings of TENCON 2018-2018 IEEE Region 10 Conference, pp.1510-1515(2018)
- [2]. Liu C.and Layland J.W.: Scheduling Algorithms for Multiprogramming in a Hard-RealTime Environment, Journal of the ACM , Vol.20, No.1, pp.46-61 (1973)
- [3]. Spuri, M. and Buttazzo, G. C.: Efficient Aperiodic Service under Earliest Deadline Scheduling, Proc. 15th IEEE Real-Time Systems Symposium, pp.2-11(1994)
- [4]. OSEK/VDX, OSEK/VDX Operating System Version 2.2.3, (2005)
- [5]. TOPPERS/ATK1, <https://www.toppers.jp/atk1.html>