

5H-4 分散リアルタイムシステムにおける
大域スケジューリング方針に関する性能評価芦沢 真哉 清水 謙多郎
電気通信通信大学

1 はじめに

リアルタイムシステムには、期限内に処理を終了することが絶対的に要求されるハードリアルタイムと、期限内にできるだけ処理を終了させようとするソフトリアルタイムがある。例えば、ソフトリアルタイムシステムでは、電話の交換機、ハードリアルタイムシステムでは、原子力発電所などが挙げられる。

本稿では、ソフトリアルタイム・タスクを対象とした分散リアルタイムシステムにおける、大域的なスケジューリング方針について分類を行ない、代表的な方針について、シミュレーションを用いた性能評価を試みる。特に、移送タスクの決定に関して、すでにあるノードで期限内の処理の終了が保証されている(以下、単に保証されていると略す)タスクを移送の対象とする方針を提案し、到着タスクのみを移送の対象とする既存の方針と比較して、その効果を調べる。

2 システムモデル

対象とするシステムは、複数のノードが共通の伝送媒体で接続された分散システムで、通信の遅延(情報収集、およびタスク移送の遅延)、および負荷による遅延の変動を考慮する。各ノードは単一プロセッサから成り、それぞれの処理能力が同じであると仮定する。

タスク間に依存関係はなく、タスクがノードの到着する時点において、到着時間、開始時間、実行時間、周期(周期タスクのみ)、期限が決定されているとする[2]。本稿では、主に非周期タスクに関して性能評価を行なう。

3 スケジューリング方針

タスクのスケジューリングは、ノード内の局所スケジューリングとノード間に渡る大域スケジューリングの2階層のスケジューリングによって実現される。

局所スケジューリングは各ノードに到着するタスクを各ノード内でスケジューリングするもので、局所スケジューラが処理する。本稿では、期限の小さいタスクを優先的に処理する Earliest Deadline First(EDF)方式を用いる。

大域スケジューリングは、各ノードに到着したタスクをそれぞれのノードだけで保証しようとするものではなく、システム内の他のノードも使って保証しようとするものである。基本的には、あるノードの局所スケジューラでタスクを保証できなかった時、そのタスクを保証できる他のノードを探し、そのノード間でタスクを移送するという方針で行なう。大域スケジューリングの方針は、表1の様に、また、それに伴う情報収集の方針は表2の様に分類することが出来る。これらの方針のうち、主に移送タスク、移送先の決定について比較を行なった。

移送タスクの決定においては、新たなタスクの到着によってすでに保証されているタスクが期限内に実行できなくなる場合、到着タスクをそのまま移送の対象とする(方針1)だけでなく、すでに保証されているタスクも合わせて移送の対象とする(方針2)、さらに、残り実行時間、移送タスクのサイズなどによって、方針1、2のどちらを行なうか随時決定する(方針3)ことを考え、その有効性を調べた。

また、負荷情報の収集の方針においては、負荷情報の内容として、待ち行列内のタスクの数や待ち行列内のタスクの残り実行時間の合計の他に、ある一定時間内のプロセッサ利用率が考え、比較を行なった。

4 シミュレーションによる性能評価

本稿では、ソフトリアルタイム・タスクを対象とし、

$$\text{達成率} = \frac{\text{期限内に実行が完了したタスクの数}}{\text{全タスク数}}$$

を性能指標として用いる。

図1は表1から考えられるいくつかの方針を比較したものである。図2より、方針1より方針2を用いた方が良い結果が得られた。この図は、負荷が最小なノードを移送先とする方針の場合であるが、図1で挙げた他の大域方式でも同様な結果が得られた。これは、方針2を用いた場合、結果として、期限までの時間が大きいタスクを移送することになるので、他ノードで保証され易くなるためと考えられる。また期限までの時間が小さいタス

表 1: 大域スケジューリングの方針

移送判定の主体	移送判定のタイミング	移送タスクの決定	移送先の決定	次の移送先候補
<ul style="list-style-type: none"> 送り手 受け手² 	<ul style="list-style-type: none"> タスク数変動時 タスク到着後、保証できないタスクの発生時 定期的 タスク移送権獲得時² 	<ul style="list-style-type: none"> 到着タスク 既存のタスク^{1,2} 上記のどちらかを <ul style="list-style-type: none"> 残り実行時間 サイズ 弛緩時間 期限 によって決定² 	<ul style="list-style-type: none"> 固定 任意 負荷が最小のもの³ 保証が可能なもの³ 距離が最小のもの 	<ul style="list-style-type: none"> $n(\geq 0)$ 個まで指定

表 2: 負荷情報の収集の方針

情報収集の主体	情報収集のタイミング ³	負荷情報の内容
<ul style="list-style-type: none"> システム内の全ノードが互いに情報交換 	<ul style="list-style-type: none"> 定期的 	<ul style="list-style-type: none"> 待ち行列内のタスク数 待ち行列内のタスクの残り実行時間の合計 $[t, t + \delta t]$ 内のプロセッサ利用率²
<ul style="list-style-type: none"> タスク保証ノード、またはタスク消滅ノードが通知 	<ul style="list-style-type: none"> タスク数変動時 <ul style="list-style-type: none"> 随時 閾値を越えた時 	

クを保証することは、残り実行時間の短いタスクを保証する傾向にあり、従って、実行時間の大きなタスクを移送することになるので、より多くのタスクを保証することができるためと考えられる。

方針3で良い結果が得られたのは、サイズを方針1、2の決定基準とした時だけである。残り実行時間を決定基準とした場合が、最も低い達成率を示した。これは、残り実行時間を決定基準とすることによって、他のノードで保証され易くすることを意図したものだが、結果的に1つの残り実行時間が大きいタスクを保証することにより、後から来る複数の残り実行時間が小さいタスクを保証できなくなったために、達成率が低下したものと考えられる。

5 おわりに

今後、応答時間を指標とする負荷分散のように受け手を主体とする移送、また、タスクを統一的に管理、割り当てを行なうような大域スケジューリング方針について

も検討していきたいと考えている。

参考文献

- [1] J.A. Stankovic, K. Ramamritham and S. Cheng. Evaluation of a Flexible Task Scheduling Algorithm for Distributed Hard Real-Time Systems, *IEEE Transactions on Computers*, Vol.34, No.12, pp.1130-1143, 1985.
- [2] K. Schwan and H. Zhou. Dynamic Scheduling of Hard Real-Time Tasks and Real-Time Threads, *IEEE Transactions on Software Engineering*, Vol.18, No.8, pp.736-748, 1992.
- [3] K.G. Shin and C. Hou. Design and Evaluation of Effective Load Sharing in Distributed Real-Time Systems, *IEEE Transactions on Parallel and Distributed Systems*, Vol.5, No.7, pp.704-719,1994.

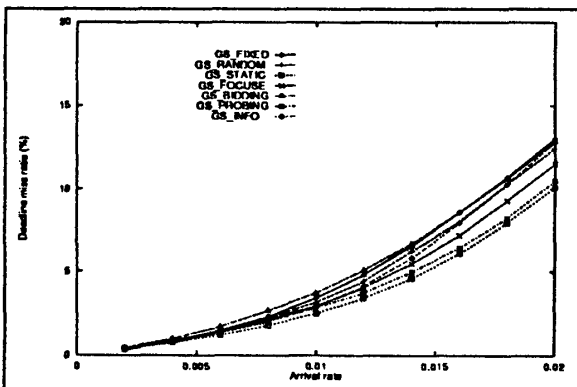


図 1: Performance of global scheduling policies

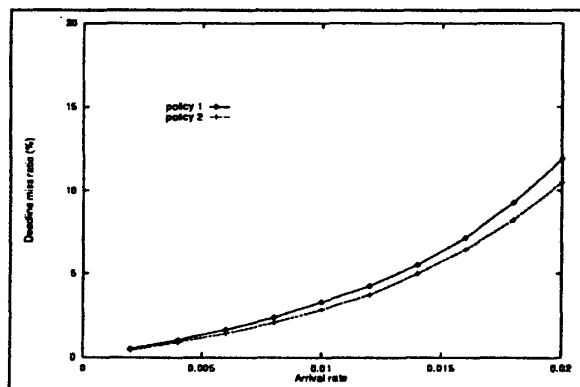


図 2: Performance comparison of policy 1 and policy 2

¹到着タスクによって、期限内に実行不可能になるタスク

²従来の負荷分散方針に対応し、新たに考えることができるもの

³移送先の決定においては、移送判定時まで収集した負荷情報を用いるものと、移送判定時にオンデマンドで収集するものとに分類できる