4D-4 複数プロセス制御におけるプログラム実行速度調整法の実現と評価

田端 利宏† 谷口 秀夫‡ 牛島 和夫‡
†九州大学大学院システム情報科学府 ‡九州大学大学院システム情報科学研究院

1 はじめに

近年、計算機ハードウェアの性能は大幅に向上している。一方、ソフトウェアの処理速度は、ハードウェア性能に大きく依存するため、計算機ハードウェアが異なれば、処理速度が異なる。この結果、ソフトウェアの利便性が低下し、最適な実行速度で利用できなくなる。したがって、将来に向け、オペレーティングシステムが、ソフトウェアの実行速度を、計算機ハードウェア性能の範囲で自由に制御する方式の確立が必要である。これまでに提案したプロセスのスケジュール法を工夫する方式[1]は、一つのプロセスの実行速度を滑らかに調整すること主眼としているため、同時に複数プロセスの実行速度を制御する方式の確立が必要である。

本稿では、同時に複数プロセスの実行速度を制御する方法について述べる。

2 プログラム実行速度調整法

2.1 基本方式

プロセスのスケジュール法を工夫した方式では、プログラムの実行と停止を繰り返すことでプログラムの実行速度を調整できる。具体的には、プログラム実行の単位時間をタイムスロットと名付け、タイムスロットの一定の連続した時間 (タイムブロックと名付ける) 内で、要求された実行速度に必要な個数分のタイムスロットの時間だけプログラムを実行し、実行速度を調整する[1]。

2.2 タイムスロット割当方式

プログラムの実行速度を調整したときのプログラムの動きは、タイムスロットの割当位置により決まる。プログラム実行速度調整法では、要求された実行速度で調整でき、処理の均一性が高いことが重要である。処理の均一性とは、プログラムの実行速度を調整したときの処理の滑らかさを表すものである。

我々は、一つのプロセスの実行速度を調整する場合においては、改良疑似周期割当方式が、最も処理の均一性が高いことを明らかにした[2]。改良疑似周期割当方式とは、要求性能に基づき n 個のタイムスロットを割り当てる場合、i 番目に割り当てるタイムスロットの位置を、(総タイムスロット数) $*(i-1)/n$ とする方式である。本稿では、改良疑似周期割当方式を用い、複数プロセスの実行速度を滑らかに調整できる方式を提案する。

3 複数プロセス制御の問題点と対処

複数プロセスの実行速度を調整する場合は、タイムスロットの割当位置が他のプロセスと衝突する問題がある。タイムスロットの割当位置が衝突した場合は、一つのプロセスだけが要求を満たされる。この結果、要求を満たされないプロセスは、他のタイムスロットが割り当てられるため、処理の均一性が低下することが懸念される。以下で二つの問題点と対処を述べる。

3.1 割当位置衝突時の処理

割当位置衝突時の問題点として、要求を満たされないプロセスに割り当てる空きタイムスロットの検索方式問題がある。検索方式は以下の三つがある。

(1) 割当衝突位置の前方検索方式
(2) 割当衝突位置の後方検索方式
(3) 割当衝突位置からの最短距離検索方式

タイムスロットの割当を再現するソフトウェアを作成し、この三つの方式を評価した。このソフトウェアは、入力された条件でのタイムスロット割当状態を再現し、割当タイムスロット間隔の標準偏差を算出する。入力条件として、各プロセスの性能の大きさとプロセスを生成する順番がある。割当タイムスロット間隔の標準偏差とは、処理の均一性の評価尺度の一つで、特定のプロセスに割り当てられた個々のタイムスロット間隔の標準偏差を求めたものである。評価は、1 タイムブロック内のタイムスロット数を 1000 個とし、その長さを 1 秒とした。プロセッサ PentiumII 450MHz の計算機上で測定した。

利用者が同時に利用するプログラム数とそのプログラムに割り当てる性能を考え、プロセスが 5 個で 60 %程度の性能を確保している場合に、6 個目のプロセスを生成した場合の 6 個目のプロセスの割当タイムスロット間隔の標準偏差を求めた。5 個のプロセスで 60 %程度の性能を確保する場合には、一つの当りの期待値が 12 となるように、ランダムで生成される値の最大値を 24 とし、13,5,8,13,23 (合計 62) を選択した。

各検索方式について処理の均一性を評価した結果を図 1 に示す。図 1 は、三つの空きタイムスロット検索方式で、6 番目に生成したプロセスの割当タイムスロット間隔の標準偏差を測定した。図 1 から、前方検索、後方検索、最短距離検索の割当タイムスロット間隔の標準偏差の平均値が、0.99, 1.04, 0.86 で、その標準偏差が、0.30,

---

Implementation and Evaluation of Mechanism of Regulating Program Execution Speed for Controlling Multi processes
Toshihiro TABATA, Hideo TANIGUCHI and Kazuo USHIJIMA
Graduate School of Information Science and Electrical Engineering, Kyushu University

表 1 タイムスロットの割当順番の影響

性能 (%)	割当タイムスロット間隔の標準偏差		
	ランダム	性能小	性能大
13	0.46	0.72	0.72
5	0.63	0.00	1.41
8	0.50	0.50	1.32
13	0.72	1.07	0.72
23	1.20	1.09	0.48
5	0.89	0.00	2.19

表 2 再割当時のタイムスロット割当時間比

	ランダム	性能小	性能大	割当位置衝突なし
割当処理時間比	1.24	1.23	1.23	1.00

0.35, 0.24 であった。この結果から、最短距離検索が割当タイムスロット間隔の標準偏差の平均値とその標準偏差が最も小さく、処理の均一性が最も高いといえる。

割当タイムスロット間隔の標準偏差の測定と同じ条件で、6 番目に生成したプロセスのタイムスロット割当処理時間を測定した。測定した処理時間と、他に性能を調整するプロセスが存在しない条件で、同じ性能分のタイムスロットを確保した場合の処理時間との比を算出した。この結果から、前方検索、後方検索、最短距離検索のタイムスロット割当処理時間の比の平均を算出すると、5 個のプロセスで 62 %を確保した場合は、1.39, 1.35, 1.31 であった。割当処理時間の最大値は、121μsec であった。各空きタイムスロット検索方式間の処理時間比を比較すると、後方検索に比べ、前方検索と最短距離検索の処理時間は短く、その差は非常に小さい。

これらの結果から、処理の均一性が最も高く、タイムスロット割当処理時間が比較的短い、最短距離検索方式を空きタイムスロット検索方式とする。

3.2 割当位置衝突の影響の分散

各プロセスへの影響の大きさを考えた場合に、各プロセスが確保した性能により受ける影響が異なると考えられる。そこで、新たにタイムスロットを割り当てる場合に、すべてのタイムスロットを再割当することで、割当位置衝突の影響をプロセス間で均等に分散させる。3.1 節と同じ 5 個の整数に加えて、さらに 6 個目の整数を同じ方法で選択した。ランダムに生成した値を性能として持つプロセスを 6 個生成し、各プロセスにタイムスロットを割り当てる順番を変えて、再割当順序の影響を評価した。ランダムに選択した結果、順番に 13, 5, 8, 13, 23, 5(合計 67) を選択した。タイムスロットを割り当てる順番には、ランダム順、性能の小さい順、性能の大きい順がある。ランダムは、利用者のプロセス生成要求順で割り当てる場合を想定する。性能の小さい順番は、性能の小さいものが影響を受けやすいため、性能の小さいものを優先した場合である。評価結果を表 1 に示す。表 1 では、性能の小さい順が最大値が 1.09 と最も小さい。また、ランダム順と性能の小さい順の割当タイムスロット間隔の標準偏差を算出すると 0.73 と 0.56 であり、性能の小さい順の方が処理の均一性が良いことがわかる。この結果から、性能の小さい順が、処理の均一性が最も高いことがわかる。タイムスロット割当処理時間について先ほどの測定と同じ条件で評価を行なった結果を表 2 に示す。表 2 から、各割当方式の差は全体の処理時間の 1 %以下であることがわかる。これらの結果から、性能の小さい順で再割当する。

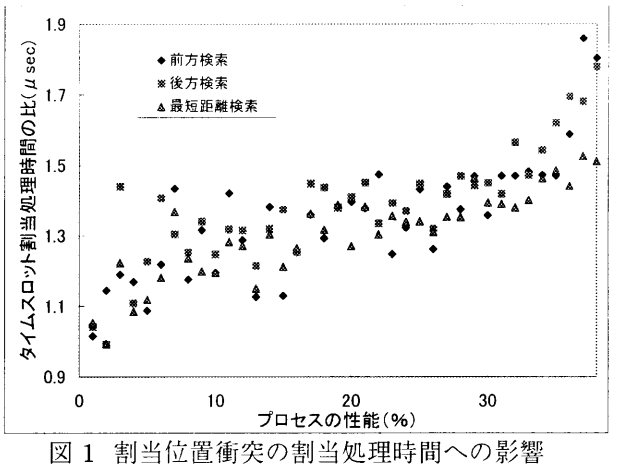

図 1 割当位置衝突の割当処理時間への影響

再割当の契機には、プロセスの生成時とプロセスの性能変更時がある。

4 おわりに

プロセスのスケジュール法を工夫してプログラム実行速度を調整する方式について、基本方式を説明し、複数プロセスの実行速度を調整する際の問題点と対処を示した。タイムスロットの割当位置衝突への対処処理では、衝突位置から最短距離の空きスロットを検索し割り当てる方法が良い。タイムスロットの再割当処理では、割当タイムスロット数が少ないプロセスから割り当てる方法が、処理の均一性が最も高い。

今後の課題として、性能調整されないプロセスへの影響の明確化がある。

参考文献

[1] 田端利宏, 谷口秀夫 : "***Tender*** オペレーティングシステムの資源「演算」によるプログラム実行速度調整機能の実現と評価", 情処学論, Vol.40, No.6, pp.2523-2533 (1999).

[2] 田端利宏, 谷口秀夫, 牛島和夫 : "プログラム実行速度調整機能における処理の均一性向上手法," 情報処理学会 第 59 回全国大会, 3Z-3 (1999).