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

3Z−3

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研究科

1 はじめに

我々は、計算機ハードウェアの性能に依存することなく、プログラムの実行速度を調整する機構を実現した[1]。本稿では、その機構において、プログラムの実行速度を調整したときの処理の均一性を向上させる手法について述べる。

2 プログラム実行速度調整機能

2.1 基本方式

プログラムの実行速度の調整は、ある単位時間(これをタイムスロットと名付ける)で、プログラムの実行と停止を繰り返すことで実現できる。タイムスロットとタイムブロックの関係を図1に示す。タイムスロットの一定の数の連続をタイムブロックと名付ける。タイムブロック中で、プログラムの実行速度を調整する要求性能(1〜100 %, プロセッサそのものの性能を100 %とする)に見合う時間分のタイムスロットをプログラムに割り当て、そのタイムスロット時間だけプログラムを走行させることで、プログラムの実行速度を調整できる。

この基本方式では、プログラムはタイムスロット単位で連続実行されるので、プログラム実行速度を調整された処理の均一性は、タイムスロットの大きさおよびタイムスロットを割り当てる方式の影響を受ける。また、タイムスロットの分割はタイマ割り込みを利用するため、タイムスロット分割の精度は、タイマ以外の割り込み処理の影響を受ける[2]。

2.2 タイムスロット割当方式

タイムスロット割当方式として、必要な周期でタイムスロットを割り当てる方式と、タイムブロックの中で必要なタイムスロット数を割り当てる方式がある。周期割当方式は、複数のプログラムについて実行速度を調整しようとすると、割り当てるべきタイムスロット位置が衝突しタイムスロットの割当てができなくなる欠点がある。そのため、後者について、以下の4つの方式がある。

(1) 非周期割当方式：処理を実行するプログラムのタイムスロット数を必要な個数だけタイムスロット内先詰めで割り当てる方式である。

Method of Improving Uniformity of Processing on Program Speed Control Mechanism
Toshihiro TABATA, Hideo TANIGUCHI and Kazuo USHIJIMA
Graduate School of Information Science and Electrical Engineering, Kyushu University
Email:tabata,tani,ushijima@csce.kyushu-u.ac.jp

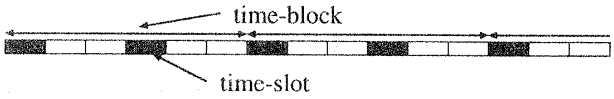


図1 タイムブロックとタイムスロットの関係

(2) 半周期割当方式：要求性能をn %とするとき、1タイムブロック内において100/n個のタイムスロット間隔で[p(n/100)]個のタイムスロットを割り当て、既に割り当てられている場合には、次の最も近いタイムスロットを割り当てる方式である。

(3) 疑似周期割当方式：要求性能が50 %以下の場合は半周期割当方式と同様にし、要求性能が50 %を超える場合は空きタイムスロットを半周期割当方式で確保する方式である。

(4) 改良疑似周期割当方式：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。

3 処理の均一性

3.1 速度調整度

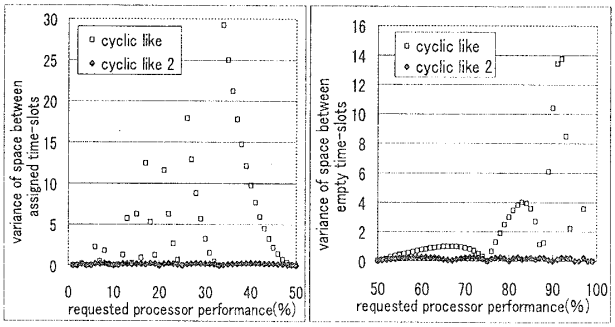
プログラムの実行速度は、プログラムの実行時間のみで評価することはできない。それは、実行時間はプログラムの開始時間と終了時間の差で決定されるからである。このため、プログラムの実行時間を評価するほかにプログラムの実行速度が滑らかに調整されているかについても評価する必要がある。

利用者から見たプログラムの速度は、入出力により認識される。そこで、プログラムの入出力に着目し、処理の均一性の評価尺度のために、速度調整度(η)を定義する。プログラムの実行速度を調整しない場合の入出力時刻の間隔をt、要求性能n %のときの入出力時刻の間隔をtnとするとき、速度調整度(η_i)を以下に示す。

$$\eta_i = tn_i/t_i/(n/100) = (tn_i/t_i) \times (n/100) \qquad (式1)$$

速度調整度は、1に近いほど処理を要求性能で調整していることを示し、1を超えると要求性能以下、1未満では要求性能を超える性能に調整していることを示す。つまり、η_iが1に近いほど、調整がうまくいっているといえる。また、速度調整度の分散が小さいほど、処理の均一性は高い。したがって、処理の均一性の評価尺度として、速度調整度の分散を利用する。つまり、各時間での処理の均一性は、隣接した入出力についての速度調整度の分散が小さいほど高いといえる。

したがって、プログラム処理全体における処理の均一性は、プログラム処理の全入出力についての速度調整度



（A）割当タイムスロット間隔の分散（要求性能50%以下）　（B）空きタイムスロット間隔の分散（要求性能50%以上）

図 2　要求性能とタイムスロット間隔の分散

の分散により判断できる。また、プログラム処理全体における要求性能の調整程度は、プログラム処理の全入出力についての速度調整度の平均値により判断できる。

3.2　タイムスロット割当方式の比較

各タイムスロット割当方式の処理について、処理の均一性の評価をアルゴリズムの面から行なう。非周期割当方式は、タイムスロットの割当は容易であるが、処理の均一性が悪い。半周期割当方式も、要求性能が 50 %以上の場合は非周期割当方式と同様なので、処理の均一性が悪い。そこで、処理の均一性の比較的高い、疑似周期割当方式と改良疑似周期割当方式を比較する。疑似周期割当方式は、要求性能が 50 %以下の場合と 50 %より大きい場合で割当方法が異なるため、この二つの場合について比較する。

50 %以下の場合、疑似周期割当方式は、タイムスロット間隔を計算し、計算結果の小数点以下を切捨て、タイムスロットを割り当てる。このため、切り捨てる値が大きい場合には、処理の均一性が低くなる。図 2(A) は、割り当てられたタイムスロットの間隔の分散を求めたものである。図 2(A) から、疑似周期割当方式のいくつかの点で、分散が大きくなっていることがわかる。特に、要求性能が 34 %、26 %、21 %、17 %の 4 点で大きい。この場合のタイムスロット間隔の計算結果の値は、2.94、3.84、4.76、5.88 であり、小数点以下の切捨て値が大きい。これに対し、改良疑似周期割当方式は、割当タイムスロットの位置を各々計算することにより、小数点以下の値の切捨てを最小限に抑えている。このため、分散の値は 0.25 以下と小さい。

次に、要求性能が 50 %以上の場合を比較する。両割当方式とも、空きタイムスロットが連続しないようにタイムスロットを割り当てる。このため、割り当てられたタイムスロットの間隔の分散は一致する。そこで、空きタイムスロットの間隔の分散に着目して、評価する。空きタイムスロットが均等に分散していれば、処理の均一性は高いといえる。空きタイムスロットの間隔の分散を図 2(B) に示す。図 2(B) より、疑似周期割当方式の分散は、改良疑似周期割当方式の分散と同じか、それより

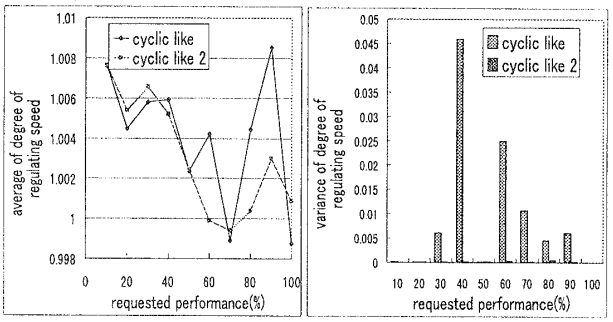


図 3　疑似周期割当方式と改良疑似周期割当方式の比較

大きい値である。このことから、改良疑似周期割当方式が優れているといえる。また、改良疑似周期割当方式の分散は最大で 0.25 であるが、疑似周期割当方式の分散は、多くの場合 0.25 より大きく、要求性能が大きくなるほど分散が大きくなる傾向がある。しかし、要求性能が大きい場合はタイムスロットが多く確保されるため、空きタイムスロット位置による処理の均一性への影響は小さくなる。

4　実装評価

処理の均一性が比較的高い疑似周期割当方式と改良疑似周期割当方式について比較する。BSD/OS に実装し、プロセッサ Pentium 233MHz の計算機で評価した。結果を図 3に示す。図 3において、改良疑似周期割当方式の速度調整度の平均は、疑似周期割当方式と同様で 1 から最大のズレでも約 0.007(要求性能 10 %) と非常に小さい。また、その分散は、疑似周期割当方式 (最大約 0.045: 要求性能 40 %) よりも小さく、最大でも約 0.00039(要求性能 80 %) である。したがって、疑似周期割当方式より、さらに改良疑似周期割当方式は処理の均一性が高いプロセッサ割当方式であるといえる。このことは、先に述べたアルゴリズムによる比較の結果を裏付けている。

5　まとめ

プログラムのプロセッサ割当を制御し、プログラムの実行速度を調整する基本方式について述べ、プログラムの処理の均一性を向上させるタイムスロットの割当法について述べた。改良疑似周期割当方式のアルゴリズムを評価し、実装評価した結果、分散の値が最大で約 0.00039 であり、従来のタイムスロット割当法に比べ、処理の均一性が高い。

今後の課題としては、複数プログラムの実行速度を調整する方式の検討がある。

参考文献

[1] 谷口 秀夫：” サービス処理時間を調整するプロセスのスケジュール法”, 電子情報通信学会 論文誌 Vol.J81-D-I, No.4, pp.386-392 (1998).

[2] 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”, 情報処理学会論文誌, Vol.40, No.6, pp.2523-2533 (1999).