

可変なプロセッサ性能を提供するスケジュール法によるサービス時間の調整の試み

谷口, 秀夫
情報工学専攻

<https://doi.org/10.15017/1485127>

出版情報：九州大学大学院システム情報科学紀要. 2 (1), pp.105-110, 1997-03-26. 九州大学大学院システム情報科学研究所
バージョン：
権利関係：

可変なプロセッサ性能を提供するスケジュール法による サービス時間の調整の試み

谷口秀夫*

Experiment of Regulating the Service Time by a Schedule Mechanism for Supplying Variable Performance of Processor

Hideo TANIGUCHI

(Received December 24, 1996)

Abstract: Improvement of processor performance cuts down software processing time, as the result it is possible for software to execute complex processing. By the way, improvement of processor performance influences software processing time. To obtain the suitable processing time of service, some service programs are coded to adjust processing time to suitable service. This paper suggests schedule mechanism to supply processor performance for suitable processing time of service. Suggested schedule mechanism can adjust processing time of service. But, in some cases of adjusting processing time of service that includes input/output processing, this schedule mechanism can not adjust processing time of service because of the input/output buffer facility of operating system.

Keywords: Schedule, Operating system, Process, Processor performance, Service time

1. はじめに

ハードウェアの性能向上とともに、計算機のソフトウェア処理時間は短縮の一途をたどっている。これにより、複雑な処理を行うことができるようになり、いろいろな使いやすいサービスが実現されている。しかし、ソフトウェアの処理はハードウェアの性能に依存するため、ハードウェアの性能向上はいくつかの問題を生んでいる。例えば、プロセッサ性能が高過ぎ、プログラムの処理時間が速過ぎるため、文字の表示速度を人間に合わせて調整する必要がある。或るいは、文字表示の利用者インターフェースそのものを変更しなくてはならない。このように、ハードウェア性能が高くなるにつれて、様々な問題が発生する。

したがって、今後は、ハードウェア性能に惑わされず、ソフトウェアが提供するサービス内容に合わせた速度で処理を実行する必要がある。このためには、性能を自由に変更できる可変性能計算機が必須である。可変性能計算機の実現には、ハードウェアによる実現とソフトウェアによる実現が考えられる。しかし、1つの計算機上で走行する複数のサービスプログラムについて、各々に自由な性能を提供するためには、ソフトウェアによる実現でなくてはならない。

可変性能計算機により、サービスプログラムは、その

実現において計算機ハードウェア性能の違いから開放される。これにより、サービスプログラムの作成において、サービス内容そのものの実現に注力できるようになる。また、サービスプログラムは、計算機に依存した部分が減少するため、その利用期間(ソフトウェアの寿命)が長くなると推察できる。さらに、計算機が提供する速度ではなく、各個人に合わせて人間が求める速度で処理を行うことも可能になる。

本稿では、可変性能計算機の実現に向けた研究の1つとして、サービスの要求に応じたプロセッサの性能を提供するスケジュール法を提案する。具体的には、可変なプロセッサ性能を提供する演算スケジュール法を示し、その試作と評価により、スケジュール法の有効性と限界を示す。

2. 既存スケジュール法と縮小化

既存オペレーティングシステム(以降、OSと略す)のスケジュール法は、サービス処理の均等実行や実時間性の保証、あるいはハードウェアの有効利用を目的に設計されている。演算スケジュール法の研究としては、特に、処理の時間制約が厳しい実時間処理についての研究^{1),2),3)}が多い。また、連続メディアを扱うための研究⁴⁾もされている。このように、既存OSのスケジュール法が実時間性の保証などを目的にしていることは、従来、サービスが求める処理速度に比べプログラムの実行時間が長いことや、ハードウェアが高価であったことに起因している。

サービス処理に対しプロセッサの性能を調整するには、

平成8年12月24日受付

* 情報工学専攻

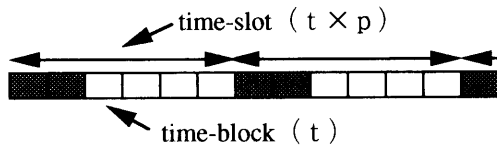


Fig.1 Time-slot and time-block.

サービス処理の実行と停止を適度な間隔で繰り返すことが必要になる。このように、プロセッサの性能を調整し制御するには、プロセッサの実行を単位時間(これをタイムスロットと名付ける)に分割し、サービス処理にタイムスロットを割り当てる割合を調整することで可能である。タイムスロットの割当処理を簡単化するため、連続する複数のタイムスロット(例えば p 個の連続したタイムスロット。これをタイムブロックと名付ける)内で、サービス処理を実行するタイムスロット数を必要な個数だけ割り当てる方式を考える。この方式は、プロセス優先度による複雑な制御は必要なく、タイムスロットを割り当てる簡単なものである。つまり、スケジュール法を縮小化(Reduced)する。

タイムスロットとタイムブロックの関係を Fig. 1 に示す。Fig. 1 では、タイムスロットは時間 t であり、タイムブロックは時間 t x p である。ここで、プロセッサの実性能を A、ブロック内でのタイムスロット割当個数を m とすると、要求プロセッサ性能 a は、

$$a = A \cdot (m/p)$$

となる。Fig. 1 は、p = 6, m = 2 であり、サービス処理は実プロセッサ性能の3分の1の性能で実行される。

3. 実現と評価考察

3.1 実現内容と測定環境

タイムスロットの大きさが10ミリ秒でタイムブロックの大きさが1秒の縮小化したスケジュール法を、Pentium (90MHZ)プロセッサの計算機に実現した。

要求プロセッサ性能に基づきタイムスロットを割り当てるため、複数のプログラムが走行しても1つのプログラムの処理時間は他の影響を受けない。そのため、評価は、評価プログラム以外が走行しない環境で行った。評価プログラムは、実プロセッサ性能において約10秒以上の処理を行うものである。評価のための実測は、10ミリ秒単位のタイマを利用した。

特に説明のない限り、以降の各図の処理時間は、タイムスロットの大きさが10ミリ秒の時で、かつ要求プロセッサ性能が100%の場合の評価プログラムの処理時間の値を1とした相対値である。また、予想の処理時間は、処理時間1の処理を要求プロセッサ性能で実行した時に要す

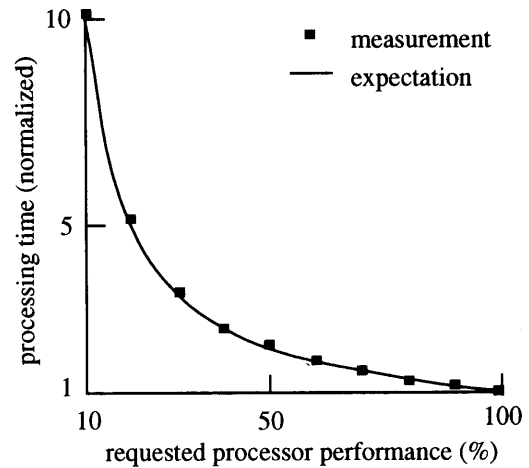


Fig.2 Relation of requested processor performance vs. processing time.

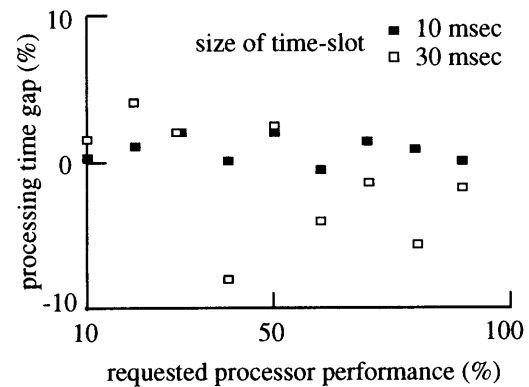


Fig.3 Relation of requested processor performance vs. processing time gap.

る時間の計算値である。さらに、処理時間差とは、実測した処理時間と予想した処理時間の差を予想した処理時間で割算した値(%)である。

3.2 要求プロセッサ性能と処理時間

3.2.1 プロセッサ処理

プロセッサ処理のみを行う場合について、要求プロセッサ性能とプロセッサ処理時間の関係を Fig. 2 に示し、要求プロセッサ性能と処理時間差の関係を Fig. 3 に示す。Fig. 2 と Fig. 3 から、次のことがわかる。

(1) Fig. 2 より、要求プロセッサ性能の大きさに関係なく、実測の処理時間と予想の処理時間はほぼ一致する。

Fig. 3 から、タイムスロットの大きさが10ミリ秒の場合、処理時間差は、最大約2%である。

(2) Fig. 3 より、要求プロセッサ性能を変化させた時、タイムスロットの大きさが10ミリ秒の場合に比べ30ミリ秒の場合、処理時間差がバラつく。これは、タイムスロットの大きさが大きいと、割り付けられた最初のタイムス

ロットで実行されるまでの待ち時間がバラつくためである。

上記2つのことから、タイムスロットの大きさを小さくすることにより、処理時間のバラツキが小さく、かつ要求プロセッサ性能を満足するスケジュールが可能である。

3.2.2 入出力処理

プロセッサ処理と入出力処理を繰り返し行う処理について、測定した。

入出力処理が磁気ディスク装置からのデータ入力処理である場合について、プロセッサ処理と入出力処理の処理時間の割合をパラメータとした時、処理時間と要求プロセッサ性能の関係をFig. 4に示し、処理時間差と要求プロセッサ性能の関係をFig. 5に示す。Fig. 4とFig. 5から、次のことがわかる。

(1) Fig. 4より、プロセッサ処理と入出力処理の処理時間の割合に関係なく、要求プロセッサ性能に見合った処理時間になる。これにより、プロセッサ性能の調整により処理の時間を調整できるといえる。

(2) Fig. 5から、実測した処理時間と予想した処理時間の差を分析すると次のことがわかる。

(A) 処理時間差は最大2.5%程度であり、上記(1)で述べたように、要求プロセッサ性能に見合った処理時間になっている。

(B) 処理時間差が大きい場合(1.5%以上)は負値であり、

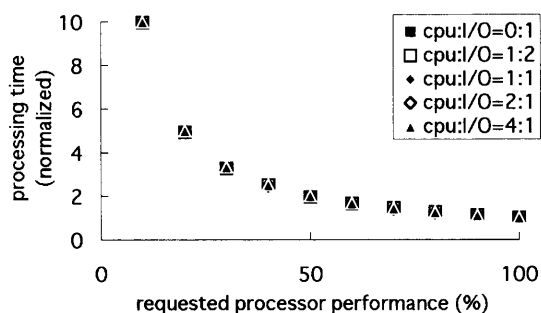


Fig. 4 Processing time vs. requested processor performance. (read data from disk)

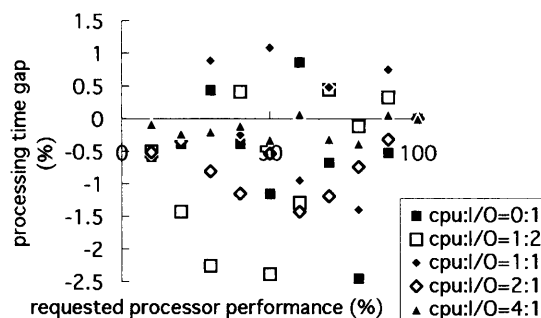


Fig. 5 Processing time gap vs. requested processor performance. (read data from disk)

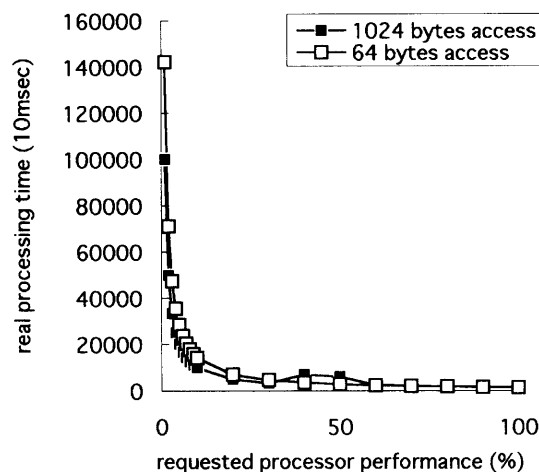


Fig. 6 Processing time vs. requested processor performance. (display)

入出力処理の割合が高い処理の場合である。これは、入出力処理の割合が高い処理の場合、入出力処理による実I/Oがプロセッサを割り付けられていない時間に行われる可能性が高くなるためである。この結果、実測した処理時間は予想した処理時間より短いものになる。

入出力処理がディスプレイへのデータ表示処理である場合について、要求プロセッサ性能と処理時間の関係をFig. 6に示し、以下に説明する。

(1) アクセス単位(表示の単位)が1024バイトの場合、要求プロセッサ性能が30%と40%の間および50%と60%の間で、実処理時間が増減する。具体的には、要求プロセッサ性能を33%から34%に上げると処理時間が急に長くなり、要求プロセッサ性能を50%から51%に上げると処理時間が急に短くなった。一方、アクセス単位が64バイトの場合、要求プロセッサ性能の増加とともに実処理時間は減少する。アクセス単位が512バイトや256バイトでは1024バイトと同様の現象が実測でき、128バイトでは64バイトと同様の現象が実測できた。これは、表示制御部が保有する表示バッファの制御方式に影響されたものである。

(2) 要求プロセッサ性能が低い場合、実処理時間はアクセス単位1024バイトの方がアクセス単位64バイトの方より短い。これは、同量のデータ表示において、アクセス単位の大きい方がシステムコール発行回数を少なくできるためである。また、アクセス単位の大きい場合、表示データが表示バッファに蓄積され、実表示処理がプロセッサを割り付けられていない時間に行われる可能性が高くなるためである。

3.3 タイムスロットの大きさと処理時間

プロセッサ処理と磁気ディスク装置からのデータ入力処理を繰り返す場合について、各処理の割合を変化させて測定した。タイムスロットの大きさをパラメータとし

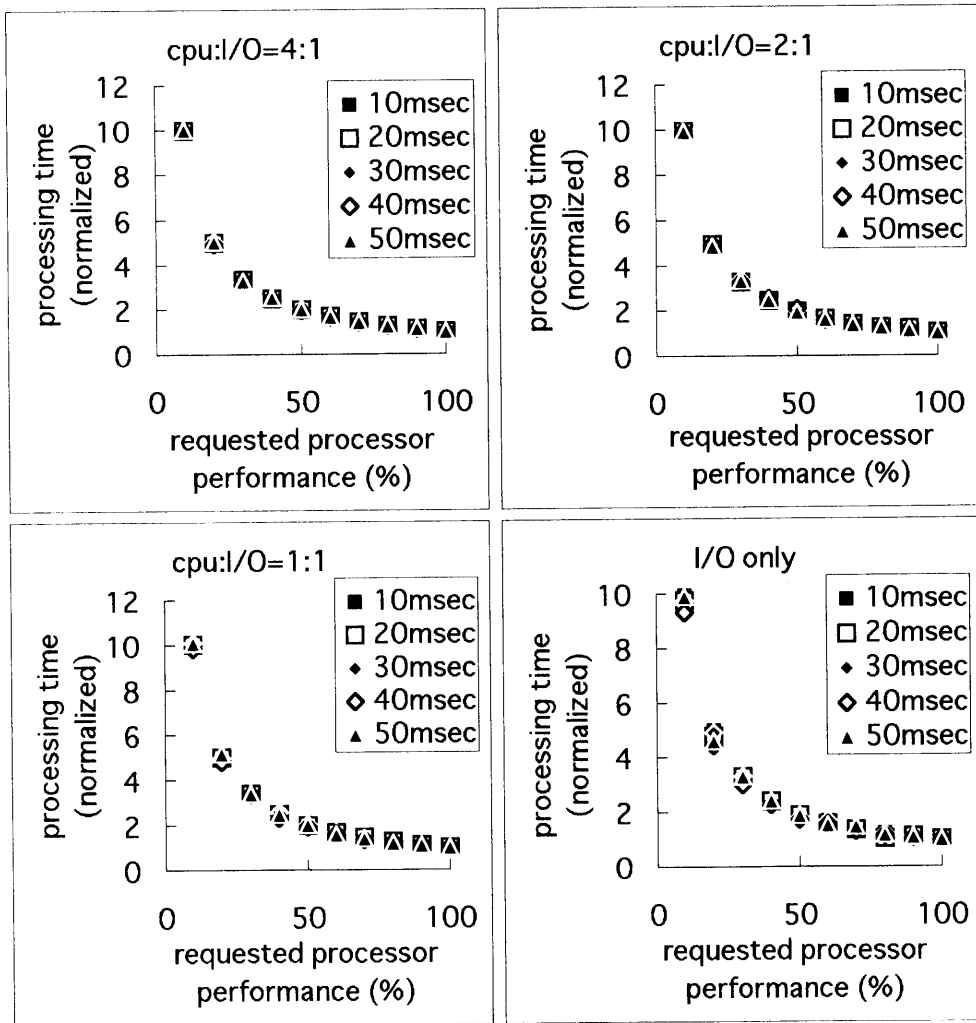


Fig.7 Processing time vs. requested processor performance (parameter: size of time-slot).

た時の処理時間と要求プロセッサ性能の関係をFig. 7に示し、処理時間差と要求プロセッサ性能の関係をFig. 8に示す。Fig. 7とFig. 8から、次のことがわかる。

(1) 入出力処理の割合が高い処理の場合、タイムスロットの大きさの影響を受けやすい。例えば、プロセッサ処理と入出力処理の処理時間の割合が4:1の場合、タイムスロットの大きさが10ミリ秒の処理時間差の絶対値の最大は0.4%であり、タイムスロットの大きさが50ミリ秒の処理時間差の絶対値の最大は1.8%である。この差は1.4%である。これに対し、入出力処理のみの場合、タイムスロットの大きさが10ミリ秒の処理時間差の絶対値の最大は2.5%であり、タイムスロットの大きさが50ミリ秒の処理時間差の絶対値の最大は8.7%である。この差は6.2%である。このように、入出力処理の割合が高い処理の場合、各タイムスロットの大きさの処理時間差が大きく、かつタイムスロットの大きさの増加に対する処理時間差の割合も大きい。

(2) タイムスロットの大きさが大きい場合、処理時間差が

大きくなる。例えば、プロセッサ処理と入出力処理の処理時間の割合が2:1の場合、タイムスロットの大きさが10ミリ秒の処理時間差の絶対値の最大は1.5%であり、タイムスロットの大きさが50ミリ秒の処理時間差の絶対値の最大は4.7%である。

上記については、先に述べたように、入出力処理による実I/Oがプロセッサを割り付けられていない時間に行われる可能性に深く関係している。

3.4 既存プログラムでの評価

評価プログラムとして、既存の2つのプログラムを利用した。1つは、指定されたファイルのデータを磁気ディスク装置から読み出し、そのままディスプレイに表示するプログラム(コマンド名“cat”)である。もう1つは、指定されたファイルのデータを磁気ディスク装置から読み出し、加工し、ディスプレイに表示するプログラムである(コマンド名“od”)である。

処理時間と要求プロセッサ性能の関係をFig. 9と

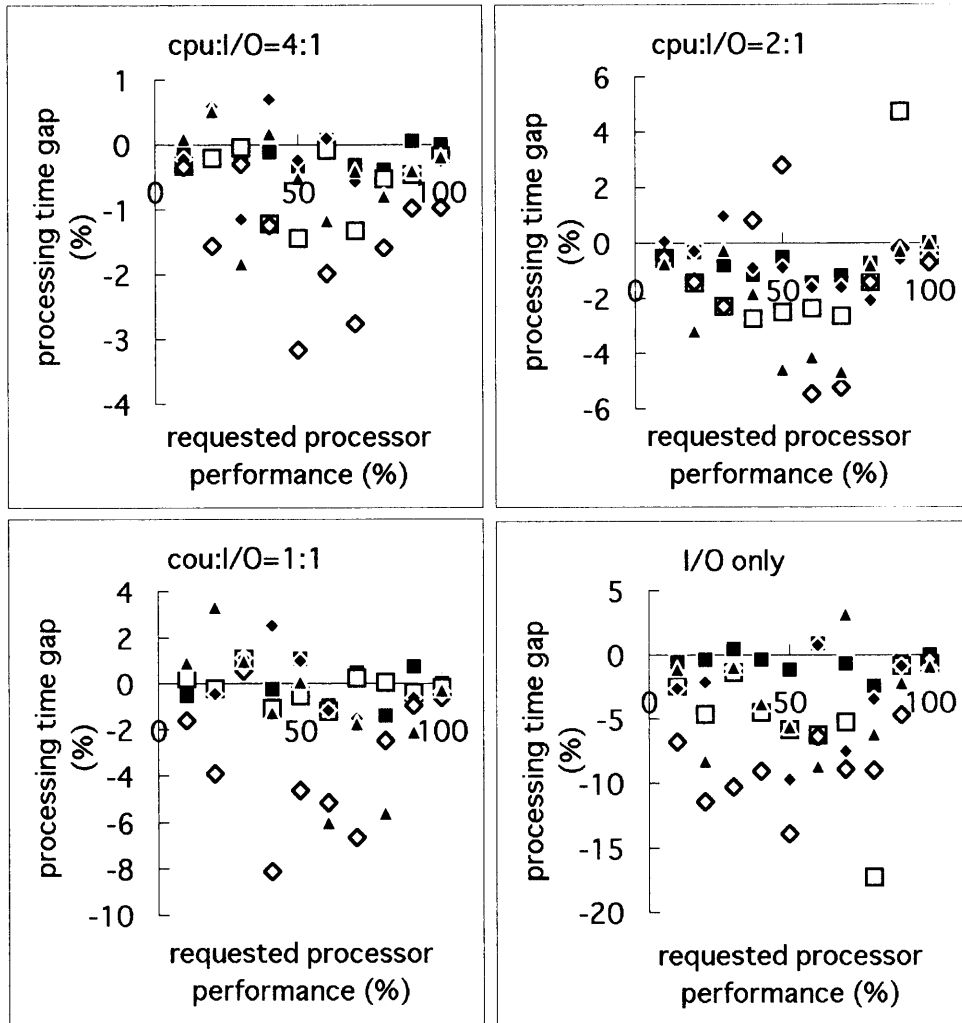


Fig.8 Processing time gap vs. requested processor performance. (parameter: size of time-slot, pattern as same as pattern of Fig. 7)

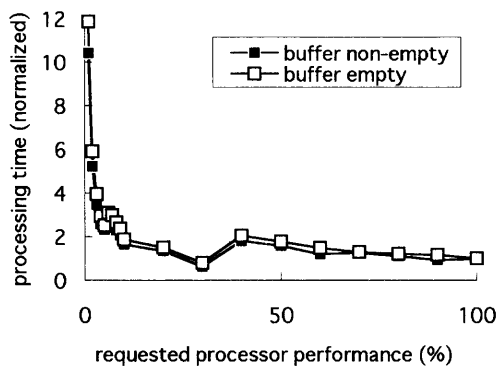


Fig.9 Processing time vs. requested processor performance. (“cat” command)

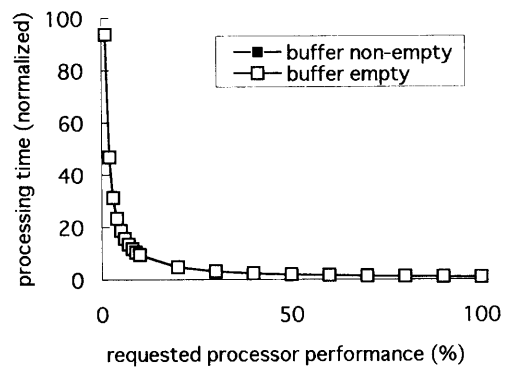


Fig.10 Processing time vs. requested processor performance. (“od” command)

Fig. 10に示す。各図において、バッファ空処理とは、当該コマンドの走行前に、当該コマンドが利用するデータがOS内のバッファに存在しないようにする処理である。これにより、バッファのキャッシュに関する影響を調べる

ことができる。各図から、以下のことがわかる。
 (1) 要求プロセッサ性能の増加に合わせて、“od”の処理時間は減少している。一方、“cat”の処理時間は、必ずしも減少していない。これは、入出力の単位の大きさが、

“cat”は1024バイトであり，“od”はそれに比べ小さいためである。

(2) “cat”の処理時間は、要求プロセッサ性能が30%付近で非常に小さく、1以下である。具体的には、要求プロセッサ性能が26%から33%の間では、処理時間が0.55%～0.68%であった。つまり、評価プログラムを要求プロセッサ性能を指定しない状態で走行させた時の処理時間より短かくなっている。これは、以下の理由による。

(A) 1つは、システムコール処理についてのOSの制御方式ためである。プロセスがデータ表示のためのシステムコールを発行し、データをOS内のバッファに複写するため「待ち状態」にある場合、磁気ディスク装置からのデータ転送終了しても、「待ち状態」であるため直ぐに次の読み込みのシステムコールを発行できない。要求プロセッサ性能を指定しない状態で走行させた場合は、要求プロセッサ性能30%付近で走行させた場合に比べ、磁気ディスク装置からのデータ転送終了時にプロセスが「待ち状態」である可能性は高い。

(B) もう1つの理由は、磁気ディスク装置からのデータ転送終了割込とディスプレイへのデータ表示終了割込の衝突によるものである。要求プロセッサ性能を指定しない状態で走行させた場合、ディスプレイへのデータ表示終了割込が定期的に発生し、磁気ディスク装置からのデータ転送終了割込が待たされる可能性が高い(これは、ディスプレイへのデータ表示終了割込レベルが磁気ディスク装置からのデータ転送終了割込レベルより高いことによる)。一方、要求プロセッサ性能30%付近で走行させた場合、割込が衝突する可能性は低くなる。

(3) バッファ空処理が、“od”の処理時間には影響せず、“cat”の処理時間には影響を与えている。これは、次の要因による。“od”の処理では、バッファ空処理なしでも、ファイルの先読み処理がデータ加工を行うプロセッサ処理の間に終了するためである。これに対し、“cat”の処理では、プロセッサ処理が非常に短いため、バッファ空処理なしの時のファイルの先読み処理が処理時間に影響を与える。

4. おわりに

サービスの要求に応じたプロセッサ性能を提供するス

ケジュール法について述べた。

プロセッサ処理や入出力を伴う処理の処理時間について実測し、評価した。入出力の割合が少ない処理ほど、またタイムスロットの大きさが小さいほど、要求プロセッサ性能に基づく予想処理時間に近い処理時間で処理を行える。しかし、プロセスのプロセッサ割当を調整するだけでは、入出力を伴う処理の処理時間を調整するには充分でない。主な結果を以下にまとめる。

(1) 応用プログラム処理とOS処理(割込処理を含む)が順序良く行われる処理においては、プロセスのプロセッサ割当を調整するだけで、入出力を伴う処理の処理時間を調整できる。

(2) 入出力を頻繁に行う処理においては、入出力による実I/Oがプロセッサを割り付けられていない時間に行われる可能性が高いため、要求プロセッサ性能に基づく予想処理時間より短い処理時間になる。

(3) 実I/Oの際にバッファを利用する入出力を頻繁に行う処理においては、処理時間がバッファの制御方式に大きく影響される。これを防ぐには、1回の入出力要求のデータ量を少なくすることが有効である。

(4) 2種類以上の入出力を頻繁に行う処理では、割込の衝突が発生し、割込レベルの低い処理が待たされる。このため、結果的に処理全体が遅くなることがある。

今後は、可変な入出力性能のための入出力スケジューリング法の検討を進める予定である。

参考文献

- 1) 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).
- 2) T. Shepard and J. A. M. Gagne, A Pre-Run-Time Scheduling Algorithm For Hard Real-Time Systems, IEEE tr. on soft. eng., Vol. 17 No. 7, pp. 669-677 (1991).
- 3) M. G. Harbour, M. H. Klein, and J. P. Lehoczky, Timing Analysis for Fixed-Priority Scheduling of Hard Real-Time Systems, IEEE tr. on soft. eng., Vol. 20, No. 1, pp. 13-28 (1994).
- 4) R. Govindan and D. P. Anderson, Scheduling and IPC Mechanisms for Continuous Media, 13th ACM Sym. on Operating Systems Principles, pp. 68-80 (1991).