Vol.2011-DPS-147 No.10
Vol.2011-MBL-58 No.10
2011/6/2

入出力時間を調整する制御法の評価

一 井 晴 那[†1]　谷 口 秀 夫[†1]

計算機が様々な用途で利用されるようになり，マルチメディアを扱う処理が増加している．マルチメディア処理は実時間性を要求する処理であるため，計算機を占有して実行することが望ましい．一方，利便性を向上させる観点から，1 つの計算機で同時に多くの処理を行えることが好ましい．しかし，1 つの計算機で同時に多くの処理を行うことにより，マルチメディア処理が他の入出力処理の影響を受け，実時間性が損なわれてしまう．マルチメディアの実時間性を保証するには，入出力時間を調整つまり保証あるいは制限できる機能が必要である．我々は，ハードウェア性能の範囲で，利用者の求める速度あるいはソフトウェアが提供するサービス内容にあわせた速度でプログラムを実行する制御法として，入出力要求数を調整する制御法を提案した．ここでは，入出力要求数を調整する制御法の評価し，その有効性を示す．

Evaluation of Mechanism for Regulating the Service Time Based on I/O time

Haruna Ichii[†1] and Hideo Taniguchi[†1]

Computers are used by various ways, and processing about multimedia increases. Multimedia processing is real-time processing, so when they run, it is to be desired that the computer is inclusive, and on the other hand it is better that a computer handle many process concorrently from the point of view of improving usability. For that, multimedia processing is affected by other I/O processing, and we want to prevent real-time processing from getting out of tune. It is necessary for guaranteeing real-time processing of multimedia to regulate, in other words, guarantee or limit I/O time. We proposed mechanism for regulating the sevice time based on controlling the number of I/O request as one of adjusting program execution time to the speed requested by user or adjusted to service within the limits of perfomance of hardware by OS. In this paper, we prove its effectiveness.

1. はじめに

計算機が普及し様々な用途に利用されるようになっている．これに伴い，マルチメディアを扱う処理が増え，例えば，マルチメディア再生処理のように実時間性を要求する処理が増加している．

マルチメディアの収録 (入力) や再生 (出力) の処理は，これらを行う入出力装置を介して行う．したがって，マルチメディアの収録や再生の処理は，オペレーティングシステム (OS) にとっては入出力処理のひとつである．先に述べたように，これらの処理は実時間性を要求する処理であるため，計算機を占有して実行することが望ましい．一方，利便性を向上させる観点から，ひとつの計算機で同時に多くの処理を行えることが好ましい．これは，マルチメディア処理も例外ではない．例えば，動画再生中にプログラムのコンパイルを行うことも求められる．しかし，これらの処理は，どちらも入出力処理を伴うため，その実行では相互に影響を与える可能性がある．しかし，マルチメディア処理の立場から見ると，他の入出力処理の影響を受けて実時間性が損なわれることは好ましくない．

プロセス優先度制御機能は，処理の優先度に応じてプロセスのプロセッサスケジュールを工夫する機能である．この機能により，プロセッサを利用する優先度を制御できるが，入出力の優先度制御はうまくできないため，ひとつの入出力処理が他の入出力処理に影響を与えてしまう．これは，次の理由による．1 つは，デバイスドライバは入出力要求順で入出力の処理を行うためである．例えば，重要でないプロセスの優先度を低く設定しても，重要なプロセスが CPU 処理を行わない時，重要でないプロセスの CPU 処理が実行され入出力要求が発行される．このため，重要なプロセスが入出力要求を発行しても，重要でないプロセスの入出力要求の処理終了を待つことになってしまう．したがって，この待ち時間により，重要なプロセスの入出力時間は長大化してしまう．なお，この問題は，プロセス優先度に基づき入出力要求の順序を入れ替えることで対処可能であるが，次の問題は解決できない．もう 1 つは，低優先度の入出力要求に基づく入出力終了割り込み処理が高優先度プロセスより優先して実行されるためである．

したがって，マルチメディア処理の実時間性を保証するには，入出力時間を調整つまり保証あるいは制限できる機能が必要である．この機能により，マルチメディア処理を保証す

†1 岡山大学大学院自然科学研究科
Graduate School of Natural Science and Technology, Okayama University

© 2011 Information Processing Society of Japan

Vol.2011-DPS-147 No.10
Vol.2011-MBL-58 No.10
2011/6/2

る，あるいは非マルチメディア処理を制限する．

我々は，ハードウェア性能の範囲で，利用者の求める速度あるいはソフトウェアが提供するサービス内容にあわせた速度でプログラムを実行する制御法として，入出力要求数を調整する制御法[1] を提案した．ここでは，プロセッサ処理と入出力処理の比率が異なるサービスについて，制御の有効性を示す．また，入出力ドライバに一度に要求できる入出力数を示す許可数 (許容係数) に着目して，入出力要求数を調整する制御法の評価を行う．

2. 関 連 研 究

入出力スケジューリングについては，シーク時間が短くなるように入出力要求の順番を入れ替えることで磁気ディスクの入出力性能を向上させる研究[2] がある．しかし，重要なプロセスと重要でないプロセスを区別なく扱うため，重要なプロセスの入出力時間の長大化を抑制できない．また，磁気ディスクのスループットを向上させつつ，重要なプロセスの入出力性能を保証する手法として，SCAN-EDF[3]，GSR[4]，RBED[5]，Fahrrad[6]，および Redline[7] が提案されている．これらの手法は，利用者が設定するデッドラインまでに重要なプロセスの入出力処理が完了することを保証する．SCAN-EDF[3] および GSR[4] は，デッドラインを超過しない範囲で，シーク時間が短くなるように入出力要求の順番を入れ替える．RBED[5] および Fahrrad[6] は，上記に加え，各重要なプロセスに設定されたデッドラインの比と入出力要求の発行頻度に応じ，各重要なプロセスの入出力要求の順番を入れ替える．これにより，デッドラインが短い，あるいは，入出力要求の発行頻度の低いプロセスを優先し，入出力性能の公平に分配する．Redline[7] は，デッドラインに基づくスケジューリングを CPU スケジューリングにも適用し，システム全体で重要なプロセスの実行速度を保証する．これらの手法は，デッドラインが短いプロセスを優先することにより，一定時間内に重要なプロセスの入出力処理を完了できる．しかし，重要でないプロセスによる入出力処理を制限しないため，実行中の入出力処理の完了待ち時間による重要なプロセスの入出力時間の長大化を抑制できない．また，プロセスに付与した優先度に基づき，入出力要求の順番を入れ替える研究[8],[9] は，重要なプロセスの入出力要求を優先的に実行できる．これらの研究は，多数のプロセスが入出力要求を発行する際，重要なプロセスの入出力要求の実行開始までの待ち時間の長大化を抑制できる．しかし，入出力性能を保証する手法と同様，重要なプロセスの入出力時間の長大化を抑制できない．また，DBMS への要求数を制限することにより，重要なプロセスの入出力性能を保証する研究[10] がある．この研究は，DBMS への要求を保留する外部キューを用意することにより，デバイスドライバを変更することなく，重要なプロセスの入出力性能を保証できる．しかし，入出力性能を保証する手法と同様，重要なプロセスの入出力時間の長大化を抑制できない．

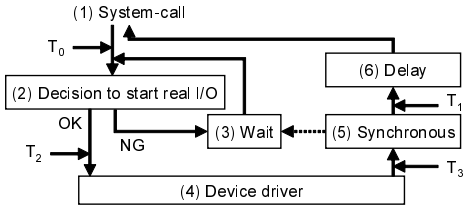


図 1 基本方式
Fig. 1 Basic method.

3. 入出力要求数を調整する制御法[1]

3.1 基 本 方 式

本制御法は，入出力デバイスの I/O 処理 (以降，実 I/O 処理と名付ける) 終了後に，要求入出力性能に合わせて遅延処理を実施し，被調整プロセスの入出力性能を調整する．さらに，デバイスドライバへの実 I/O 要求数 (実 I/O 処理を行うプロセス数) を制御する．これにより，調整精度への他プロセスの影響を抑制する．この制御法を「入出力要求数を調整する制御法」と名付ける．入出力要求数を調整する制御法の基本方式を図 1 に示し，以下に説明する．

(1) プロセスは入出力システムコールを発行する．
(2) 実 I/O 可否判断規則 (詳細は，3.2 節で述べる) に基づき，実 I/O 処理を許可するか否か判断する．許可しない場合，(3) の処理を行う．許可する場合，(4) の処理を行う．
(3) 入出力優先度に基づき，プロセスを待ちキューにつないで待ち状態にする．要求入出力性能が高いものを高入出力優先度とし，同入出力優先度は LRU で管理する．同期処理からの同期により，入出力優先度が最も高いプロセスを起床する．
(4) 実 I/O 処理を行う．
(5) 待ち処理に同期を送信する．
(6) 要求入出力性能に基づき，遅延処理を実施する．遅延時間は，要求入出力性能より算出する理想の調整後の入出力時間 (以降，理想の入出力時間と名付ける) から入出力処理に要した時間 $(T_1 - T_0)$ を減算した値である．遅延時間 T_S の算出式を以下に示す．

© 2011 Information Processing Society of Japan

Vol.2011-DPS-147 No.10
Vol.2011-MBL-58 No.10
2011/6/2

$$T_S = \frac{100}{\text{要求入出力性能}} \times \text{実 I/O 時間} - (T_1 - T_0) \quad (1)$$

ここで，実 I/O 時間とは，実 I/O 処理に要する時間である．

実 I/O 時間は，デバイスドライバ外で計測を行う．デバイスドライバ内で実 I/O 開始待ちが発生しない場合，図 1 の T_2 と T_3 との差分が実 I/O 時間となる．一方，デバイスドライバ内で実 I/O 開始待ちが発生する場合，T_3 の間隔 (実 I/O 終了時刻の間隔) が実 I/O 時間である．したがって，以下の式により実 I/O 時間を決定する．

$$\text{実 I/O 時間} = MIN((T_3 - T_2), T_3\text{の間隔}) \quad (2)$$

本制御法を OS カーネル内に実現することにより，デバイスドライバに共通的な機能として組み込む．また，制御したい入出力デバイスごとに本制御法を実現する．ここで，入出力要求から入出力デバイスの情報を取得することにより，ひとつのデバイスドライバが複数の入出力デバイスを管理する場合においても，入出力デバイスごとに制御できる．

3.2 実 I/O 可否判断規則

実 I/O 可否判断規則ではデバイスドライバへの実 I/O 処理を要求するプロセス数を制御する．具体的には，被調整プロセスの実 I/O 処理が要求入出力性能から算出する理想の入出力時間内に終了するように，実 I/O 要求プロセス数の最大値 (以降，許容値と名付ける) を決定する．

被調整プロセスの理想の入出力時間は，占有状態における入出力時間を要求入出力性能で割った値となる．したがって，被調整プロセスが 1 つの場合，デバイスドライバ内の実 I/O 要求の数 (被調整プロセスを含む) が被調整プロセスの要求入出力性能の逆数以下であれば，被調整プロセスの実 I/O 処理は，理想の入出力時間内に終了できる．

被調整プロセスが複数存在する場合，全ての被調整プロセスの調整精度を保証するためには，許容値の決定において全ての被調整プロセスの要求入出力性能を考慮する必要がある．しかし，要求入出力性能の高い被調整プロセスを考慮すれば，要求入出力性能の低い被調整プロセスを考慮しなくても，両方の調整精度は保証される．このため，要求入出力性能が高い上位 k 番目までの要求入出力性能の和を利用する．以降，この k を許容係数と名付ける．なお，共存する他プロセスが入出力処理を実行できなくなることを防ぐため，許容値は少なくとも 1 である．以上より，実 I/O 処理を行っていない被調整プロセスの上位 i 番目の要求入出力性能を P_i とし，許容値を以下のように決定する．

$$MAX(1, \frac{100}{\sum_{i=1}^{k} P_i} - 1) \quad (3)$$

ここで，許容係数は求められる調整精度に応じて設定される．求められる調整精度があまり高くない場合，スループット低下を防ぐために，許容係数は小さい方が良い．一方，多くの被調整プロセスについて要求入出力性能の調整精度を高めるには，許容係数は大きい方が良い．

4. 評　　価

4.1 評 価 項 目

入出力要求数を調整する制御法を FreeBSD 6.3-RELEASE(以降，FreeBSD と呼ぶ) の OS カーネル内に実装し，Celeron(1.8GHz) プロセッサを搭載した計算機で走行させた．入出力性能の調整精度に着目して評価した．

4.2 評 価 条 件

多くのプログラムは，CPU 処理と I/O 処理を繰り返しながら処理を進める．そこで，文献 1) と同じ評価プログラムを利用した．具体的には，評価プログラムは，CPU 処理として特定のメモリ領域の値のインクリメント処理，および I/O 処理として read() システムコールにより磁気ディスク装置から 512 バイト読み込む入力処理を繰り返す．なお，入力位置はランダムである．また，read() システムコールの前後で取得したハードウェアクロックの差を測定し，入出力時間とする．評価では，インクリメント処理の回数を変更することにより，CPU 処理の時間を変更し，CPU 処理時間と I/O 処理時間の比率 (以降，処理比率と名付ける) を変化させた．

調整精度の評価尺度として，調整比を用いる．調整比は，

$$\text{調整比} = \frac{\text{入出力時間の実測値}}{\text{理想の入出力時間}} \quad (4)$$

である．ここで，入出力時間の実測値とは，read() システムコール発行から戻り値が返却されるまでの時間である．また，理想の入出力時間とは，実 I/O 時間を要求入出力性能で割った値である．以降の評価では，評価プログラムによる CPU 処理と I/O 処理を 1002 回繰り返し，入出力時間を測定し，最初と最後を除いた 1000 回の入出力時間を評価に利用する．

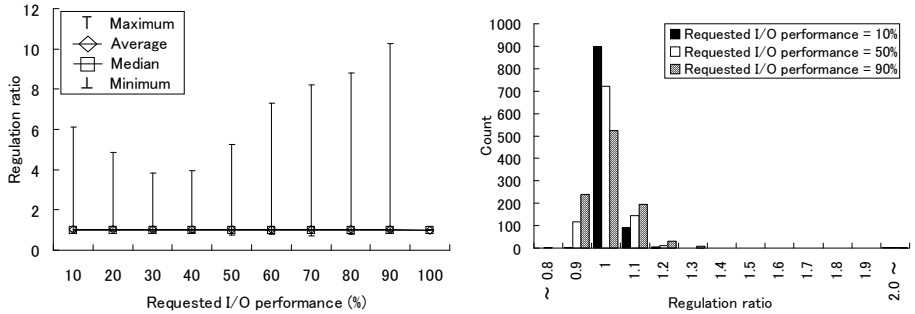
4.3 占有状態の調整精度

占有状態における被調整プロセスの調整比と頻度分布について，被調整プロセスの処理比率が 0∶1 の場合と 2∶1 の場合を図 2 と図 3 に示す．図 2 と図 3 から，以下のことがいえる．

(1) いづれの場合も，調整値は最大値のみが平均値や中央値より大きく外れている．例えば，要求入出力性能が 50 %のとき，処理比率が 0:1 の場合，最大値は平均値の約

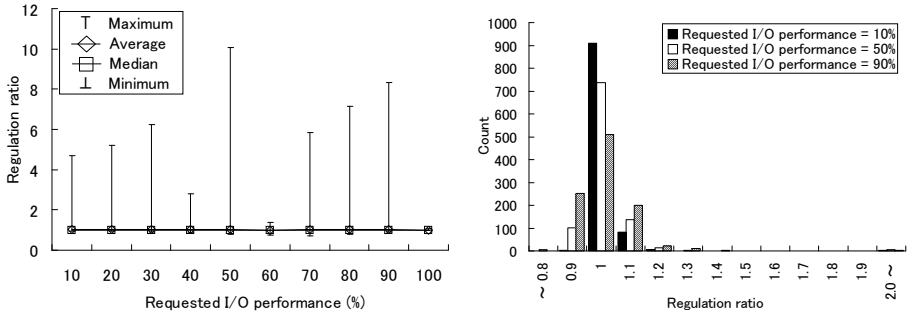
© 2011 Information Processing Society of Japan

Vol.2011-DPS-147 No.10
Vol.2011-MBL-58 No.10
2011/6/2



(A) Ragulation ratio (B) Distriobution of frequency of regulation of ratio

図 2 調整比と調整比の頻度分布 (CPU:I/O=0:1)
Fig. 2 Regulation ratio and distriobution of frequency of regulation of ratio(CPU:I/O=0:1).



(A) Ragulation ratio (B) Distriobution of frequency of regulation of ratio

図 3 調整比と調整比の頻度分布 (CPU:I/O=2:1)
Fig. 3 Regulation ratio and distriobution of frequency of regulation of ratio(CPU:I/O=2:1).

5.2 倍，処理比率が 2:1 の場合，最大値は平均値の約 10 倍であり，要求入出力性能が 20 %のとき，処理比率が 0:1 の場合，約 4.8 倍，処理比率が 2:1 の場合，約 5.2 倍である．

(2) いづれの場合も，調整比の大半は 1 付近であり，大きく外れた値として最大値がある．例えば，要求入出力性能が 50 %の場合，調整比の 99 %以上が 1.2 以下であり，最大値は 2.0 以上と大きい．しかし，調整比が 2.0 以上と大きくなる場合は，全体の 0.3 %前後と非常に少ない．

文献 1)(処理比率 1:1 で評価) と上記の結果より，占有状態において，被調整プロセスの処理比率に関係なく，調整精度は高いといえる．

4.4 他プロセスの影響

文献 1) では，調整しないプロセス (以降，共存プロセスと名付ける) を 1 つだけ同時走行させた場合における被調整プロセスの要求入出力性能と調整比 (平均値) の関係を明らかにした．ここでは，3 つ同時に走行させた場合について評価する．調整比の様子を図 4 に示す．なお，許容係数は 1 である．

図 4 より，被調整プロセスの処理比率に関係なく，被調整プロセスの要求入出力性能が低い場合，共存プロセスの影響は小さく，調整精度が高いことがいえる．また，共存プロセスの処理比率に関係なく，被調整プロセスの要求入出力性能が高くなるほど，調整精度は低下する．

共存プロセスの処理比率と被調整プロセスの要求入出力性能との関係は文献 1) で述べられているように，被調整プロセスの要求入出力性能が高い場合，共存プロセスの処理比率が高くなるにつれ，共存プロセスの影響は大きくなる．これは，共存プロセスの処理比率が高いほど，被調整プロセスと共存プロセスが同時期に入出力要求を発行する可能性が高くなり，デバイスドライバ内で被調整プロセスの実 I/O 開始待ちが発生するためである．しかし，本評価ではその傾向が現れていない．文献 1) では共存プロセス数が 1 である．しかし，ここでは，同時に走行する共存プロセス数が 3 である．このため，文献 1) と比べ，共存プロセス全体の発行する入出力要求数が増加している．つまり，被調整プロセスへ与える影響として，共存プロセスの入出力処理の比率の高さより共存プロセスの数の多さが大半を占め，共存プロセスの入出力処理の比率の高さによる影響が結果に現れていない．

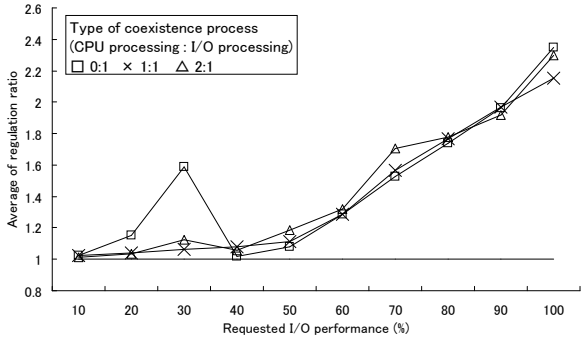
文献 1)(共存プロセス数が 1 の評価) と上記の結果より，被調整プロセスの要求入出力性能が低い場合，共存プロセスの影響は少ない．被調整プロセスの要求入出力性能が高い場合，共存プロセスの入出力処理の比率の高さと同様に，共存プロセス数の多さも被調整プロセスに与える影響は大きい．

4.5 許容係数とスループットの関係

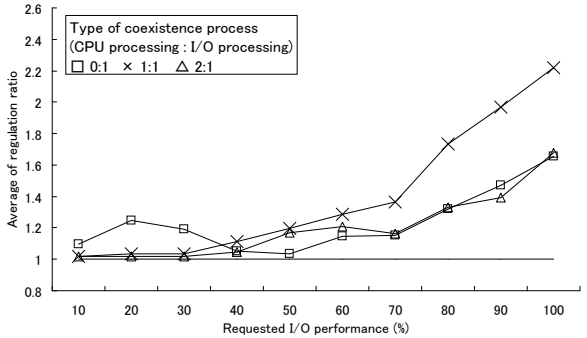
許容係数とスループット (1 秒あたりの実 I/O 回数) の関係を明らかにする．要求入出力

© 2011 Information Processing Society of Japan

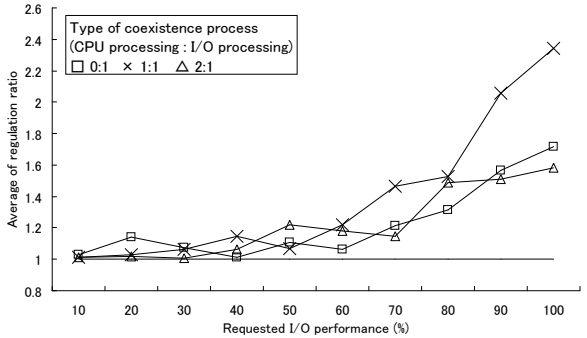
Vol.2011-DPS-147 No.10
Vol.2011-MBL-58 No.10
2011/6/2



(A) Type of regulated process(CPU:I/O=0:1)

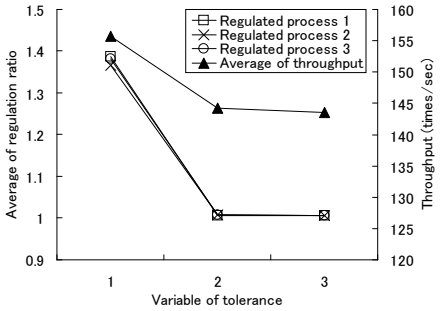


(B) Type of regulated process(CPU:I/O=1:1)

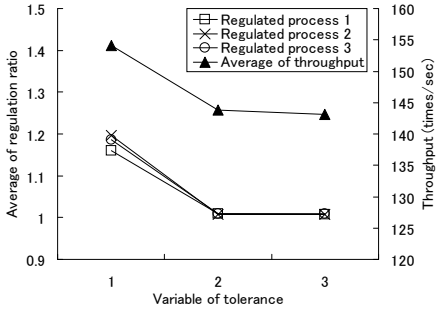


(C) Type of regulated process(CPU:I/O=2:1)

図 4 共存プロセスを 3 つ同時走行させた場合の調整比
Fig. 4 Regulation ratio when three coexistence processes run.



(A) Type of regulated process(CPU:I/O=0:1)



(B) Type of regulated process(CPU:I/O=1:1)

図 5 許容係数と被調整プロセスの調整比，およびスループット
Fig. 5 Variable of tolerance VS. regulation ratio of regulated process or throughput.

性能が 15 %の 3 つの被調整プロセスと 6 つの共存プロセスを走行させた場合における許容係数，被調整プロセスの調整比 (平均)，およびスループットの関係を図 5 に示す．ここで，入出力性能の調整精度への影響を大きくするため，共存プロセスの処理比率を 0：1 とし，最大の I/O 処理を要求するものとした．図 5 から被調整プロセスの処理比率に関係なく，以下のことがいえる．

(1) 許容係数が大きいほど，調整精度は向上するものの，スループットは低下する．
(2) 許容係数が小さいほど，調整精度は低下するものの，スループットは増加する．

例えば，図 5 の (A) では許容係数が 3 の場合，調整比は約 1.0，スループットは約 143 回/秒であり，許容係数が 1 の場合は約 1.37 と約 155 回/秒である．

文献 1)(処理比率 2:1 で評価) と上記の結果より，被調整プロセスの処理比率に関係なく，許容係数が大きいほど，調整精度は向上するものの，スループットは低下し，許容係数が小さいほど，調整精度は低下するもののスループットは増加するといえる．

4.6 高入出力優先度の優先処理の効果

3.1 節で述べたように，入出力要求数を調整する制御法では，待ち処理において高入出力優先度の要求を優先して処理している．ここでは，その効果を明らかにする．

要求入出力性能が異なる 5 つの被調整プロセスを同時に走行させた場合について，各被調整プロセスの入出力処理開始時刻を図 6 に示す．各プロセスの処理比率はいずれも 0:1 であり，入出力の回数はいずれも 1000 回である．なお，許容係数は 1 である．図 6 より，以下のことがわかる．

(1) 要求入出力性能が高いプロセス (50 %，40 %，30 %) の入出力処理開始時刻は，最初 0 秒から始まっており，連続して入出力が行われている．また，要求入出力性能が高いほど 1000 回の入出力処理が早く終了している．したがって，高入出力優先度の要求は優先的に処理開始されていることがわかる．

© 2011 Information Processing Society of Japan

Vol.2011-DPS-147 No.10
Vol.2011-MBL-58 No.10
2011/6/2

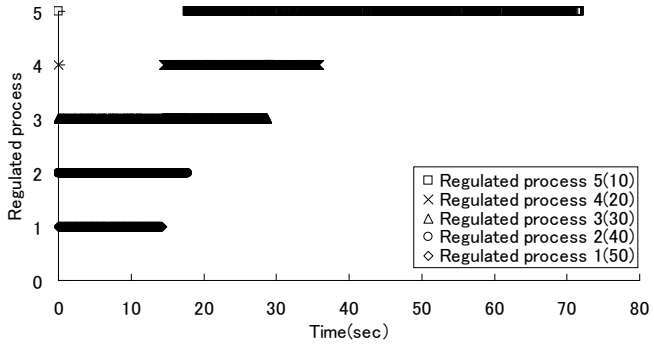


図 6 被調整プロセスの入出力処理開始時刻
Fig. 6 Start time of regulated process.

(2) 要求入出力性能が低いプロセス (20 %，10 %) の入出力処理開始時刻は，最初 0 秒付近で始まるものの，その後の処理は行われず，要求入出力性能が高いプロセスの入出力処理が終了してから本格的に行われている．これは，要求入出力性能が高い 3 つのプロセスの要求入出力性能の和が 100 %を超えるため，磁気ディスク装置の入出力性能がボトルネックとなることに起因すると推察できる．なお，ほぼ同時に本格的な入出力処理が開始されているにもかかわらず，要求入出力性能 20 %のプロセスの入出力処理が 10 %のプロセスの入出力処理より早く終了しているのは，高入出力優先度の要求が優先的に処理開始されているためである．

5. おわりに

入出力要求数を調整する制御法は，入出力デバイスへの実 I/O 要求数を制御し，実 I/O 処理終了後に遅延処理を行うことで，プロセスの入出力時間を理想の入出力時間に調整する．入出力デバイスへの実 I/O 要求を制御することにより，被調整プロセスの実 I/O 処理時間が要求入出力性能から算出する理想の入出力時間内に終了することを可能にしている．このため，入出力デバイスへの実 I/O 要求数を制御する指標として許容値を定める．許容値は，入出力デバイスへの実 I/O 要求プロセス数の最大値であり，被調整プロセスの上位 k 番目 (k:許容係数) までの要求入出力性能の和を基に算出される．

評価により，本制御法は 1 つの被調整プロセスのみが走行する場合，被調整プロセスの CPU 処理と I/O 処理の比率にかかわらず，高い精度で入出力性能の調整を行うことができることを明らかにした．また，共存プロセスが存在する場合，共存プロセスの数の多さが被調整プロセスへ影響を与えることを示した．さらに，許容係数を大きくして実 I/O 要求数を制御することにより，被調整プロセスの処理比率に関係なく，実 I/O スループットは低下するものの高い精度で調整できることを示した．また，要求入出力性能が高い被調整プロセスほど，優先して入出力処理が行われることを検証した．

参 考 文 献

1) 長尾 尚, 谷口 秀夫, ”入出力要求数の制御によりサービス時間を調整する制御法の実現と評価, ” 電子情報通信学会論文誌,Vol.J94-D,No.7,pp.(未定),Jul. 2011.
2) M. Seltzer, P. Chen, and J. Ousterhout, ”Disk scheduling Revisited,” Proc. USENIX Technical Conference, pp.313–324, January 1990.
3) A.L.N. Reddy, J. Wyllie, and K.B.R. Wijayaratne, ”Disk scheduling in a multimedia I/O system,” ACM Trans. Multimedia Computing, Communications, and Applications (TOMCCAP), Vol.1, no.1, pp.37–59, February 2005.
4) H.P. Chang, R.I. Chang, W.K. Shih, and R.C. Chang, ”GSR: A global seek-optimizing real-time disk-scheduling algorithm,” Journal of Systems and Software, vol.80, no.2, pp.198–215, February 2007.
5) S.A. Brandt, S. Banachowski, C. Lin, and T. Bisson, ”Dynamic Integrated Scheduling of Hard Real-Time, Soft Real-Time and Non-Real-Time Processes,” Proc. 24th IEEE Real-Time Systems Symposium (RTSS 2003), pp.396–407, December 2003.
6) A. Povzner, T. Kaldewey, S. Brandt, R. Golding, T.M. Wong, and C. Maltzahn, ”Efficient Guaranteed Disk Request Scheduling with Fahrrad,” In EuroSys ’08, vol.42, no.4, pp.13–25, April 2008.
7) T. Yang, T. Liu, E.D. Berger, S.F. Kaplan, and J.E.B. Moss, ”Redline: First Class Support for Interactivity in Commodity Operating Systems,” In 8th Symp. on Operating Systems Design and Implementation (OSDI), pp.73–86, December 2008.
8) R.K. Abbott and H. Garcia-Molina, ”Scheduling Real-Time Transactions: A Performance Evaluation,” ACM Trans. Database Systems, vol.17, no.3, pp.513–560, September 1992.
9) M.J. Carey, R. Jauhari, and M. Livny, ”Priority in DBMS resource scheduling,” Int. Conf. on Very large data bases (VLDB ’89), pp.397–410, July 1989.
10) B. Schroeder, M. Harchol-Balter, A. Iyengar, E. Nahum, and A. Wierman, ”How to Determine a Good Multi-Programming Level for External Scheduling,” Int. Conf. on Data Engineering (ICDE), pp.60, April 2006.

© 2011 Information Processing Society of Japan