

オンライン処理とバッチ処理のファイル I/O の特徴を生かして 処理負荷を分散制御する入出力制御法

田辺雅則^{†1} 横山和俊^{†2} 長尾 尚^{†1} 谷口秀夫^{†1}

概要：銀行のオンラインシステムに代表される業務システムでは、オンライン処理とバッチ処理が別計算機で実行される。しかし、計算機の処理能力向上とともに、計算機資源を十分に利用しない時間帯が生じ、分割損が生じている。そこで、両処理を同じ計算機で実行するシステム構成が可能である。しかし、このシステム構成においては、2つの処理の負荷を長期だけではなく短期にも時間的に分散させ、バッチ処理の負荷がオンライン処理の応答時間に悪影響を与えないようにすることが強く求められる。サービスの形態から、長期の時間的な負荷分散は可能であるが、短期の負荷分散は困難であり、プロセッサ利用優先度だけでは不十分である。そこで、本稿では、オンライン処理とバッチ処理のファイル I/O の特徴として入出力データ長に着目したディスク I/O 制御方式を述べ、評価結果を報告する。

キーワード：ディスク、I/O、スケジューリング、オンライン処理、バッチ処理

1. はじめに

オンライン処理とバッチ処理が行われるシステムの代表例として、銀行のオンラインシステムがある。オンライン処理では応答時間への要求が厳しいため、オンライン処理とバッチ処理は別計算機で実行される。しかし、両処理の計算機資源の利用量は時間帯で異なるため、計算機の処理能力向上とともに、計算機資源を十分に利用しない時間帯が生じ、分割損が生じている。そこで、両処理を同じ計算機で実行するシステム構成が可能である。

しかし、このシステム構成においては、両処理の負荷を長期だけではなく短期にも時間的に分散させ、バッチ処理の負荷がオンライン処理の応答時間に悪影響を与えないようにすることが強く求められる。オンライン処理は昼間帯に実行され、バッチ処理は主に夜間帯に実行することで、長期の時間的な負荷分散は可能である。しかし、バッチ処理の増加とともに、バッチ処理を昼間帯で実行することも必要になっており、短期の時間的な負荷分散をできることが強く求められる。

計算機資源の利用量を処理の特徴にあわせて負荷分散できる方法として、プロセス優先度によるプロセッサ割り当て制御法がある。しかし、プロセスの処理やプロセスが発行したシステムコール処理は、この制御法によって負荷分散できるが、割り込み処理はこれらの処理に比べ優先して実行されるため負荷分散できない。つまり、例えば、低優先度のプロセスが依頼した入出力要求による割り込み処理は、高優先度のプロセスの処理より優先して実行される。したがって、入出力処理が比較的多いオンライン処理やバッチ処理においては、不十分である。具体的には、低優先度のバッチ処理が I/O 要求を実行した場合、高優先度のオンライン処理はバッチ処理の I/O 割り込み処理の影響を受

け、応答時間が遅くなってしまふ。したがって、短期の時間的な負荷分散を実現するには、入出力処理を制御する方法に工夫が必要である。

文献[1]では、オンライン処理の書き込みデータ長は 1,000 バイト以下が大半を占める一方で、バッチ処理の書き込みデータ長は数 MB から 100MB のファイルサイズが多いことに着目し、書き込みデータ長を考慮したディスク I/O 制御方式を提案し、提案制御方式の有効性をシミュレーションによって評価している。このディスク I/O 制御方式では、バッチ処理の I/O 要求の実行を短期の時間的な負荷分散を行い、オンライン処理の I/O 要求を優先的に実行する。

本稿では、上記の制御方式を FreeBSD に実装し、評価し、オンライン処理とバッチ処理が混在する環境において、ディスク I/O 制御方式によって、バッチ処理によるオンライン処理の応答時間への影響を軽減し、オンライン処理の I/O 要求の処理時間およびオンライン処理の処理時間の改善が可能であることを述べる。

2. オンライン処理とバッチ処理の混在環境

2.1 オンライン処理とバッチ処理の特徴

オンライン処理は、利用者からの要求にもとづいて実行され、応答時間を重視することから、処理時間が短いものが多い。このため、プロセッサ利用時間が短く、ディスクへの I/O 要求も少ない。一方、バッチ処理は、オンライン処理に比べ、プロセッサ利用量も多く、多量なデータの I/O 要求を実行する。銀行オンラインシステムに代表されるシステムについて、オンライン処理とバッチ処理の特徴を述べる。

(1) トランザクション数とバッチ処理数

オンライン処理とバッチ処理の計算機資源の利用量は時間帯で異なる。オンライン処理は、主に昼間帯（6時から18時）に実行される。これに対し、バッチ処理の時間帯ごとのバッチ処理数を図1に示す。昼間帯（6時から18時）のバッチ処理数は、時間あたり 10～100 程度であり、夜間

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

^{†2} 高知工科大学
Kochi University of Technology

深夜帯（18時から6時）に比較して少ないが、一定数存在している。

オンライン処理の計算機資源の利用量として、一日のピーク時間帯である午前中のオンライン処理のトランザクション数を図2に示す。9時から10時のピーク時間帯におけるオンライン処理の負荷は約13TPS（トランザクション数は約48,000）である。なお、この時のバッチ処理数は約10である。つまり、13TPSのオンライン処理と10のバッチ処理が同時に実行される。

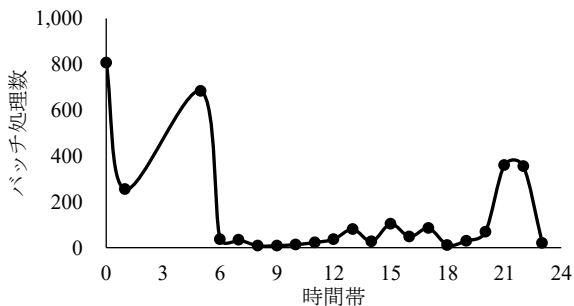


図1 一日のシステム運用におけるバッチ処理数

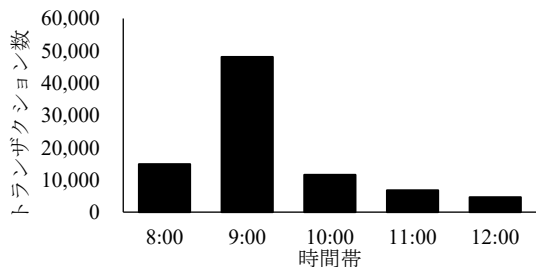
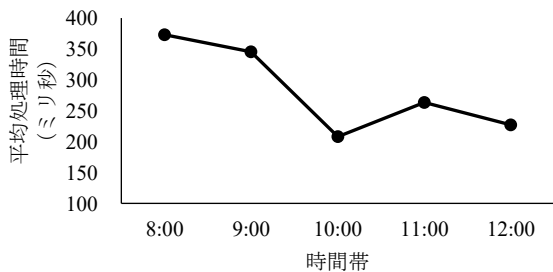


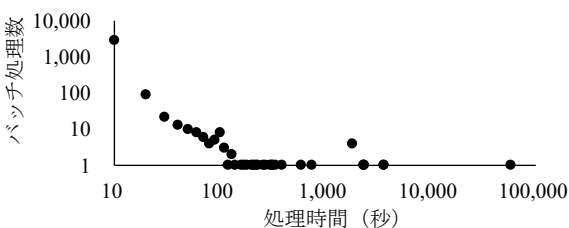
図2 一日のピーク時間帯のトランザクション数

(2) 処理時間

図3にオンライン処理の平均処理時間とバッチ処理の処理時間の分布を示す。図3(A)より、オンライン処理の平均処理時間は、約200ミリ秒から370ミリ秒である。オンライン処理の負荷が大きい9時から10時の時間帯では処理



(A) オンライン処理の時間帯の処理時間



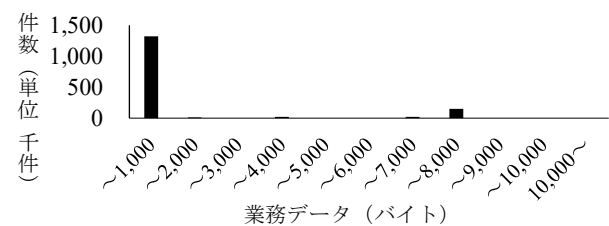
(B) バッチ処理の処理時間の分布

図3 オンライン処理とバッチ処理の処理時間

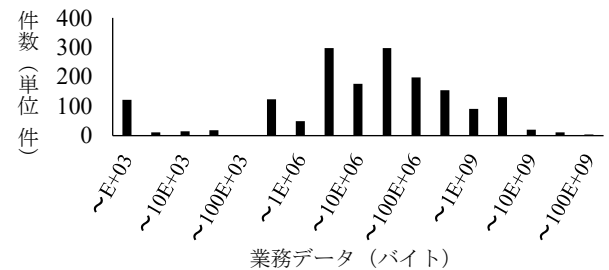
時間は約350ミリ秒である。なお、図3(A)のオンライン処理の処理時間には、通信にかかる処理時間も含まれている。また、図3(B)より、バッチ処理の処理時間は、短いものから長いものまで、様々である。一日に実行されるバッチ処理の処理時間を単純平均した場合、バッチ処理の平均処理時間は約30秒である。

(3) 書き込みデータ長

オンライン処理とバッチ処理は、ディスクに書き込むデータ長に大きな違いがある。オンライン処理とバッチ処理の業務データのデータ長の分布を図4に示す。図4(A)より、オンライン処理の書き込みデータ長は、1,000バイト以下が大半を占める。一方、図4(B)より、バッチ処理の書き込みデータ長は、数MBから100MBである。



(A) オンライン処理



(B) バッチ処理

図4 書き込みデータ長の分布

2.2 課題

オンライン処理とバッチ処理の計算機資源の利用量の大半は、時間帯で異なるため、長期の負荷分散は可能である。しかし、オンライン処理が多い昼間帯にもバッチ処理が実行される。このため、以下の課題に対処し、短期的な負荷分散を実施する必要がある。

計算機資源の利用量を処理の特徴にあわせて負荷分散できる方法として、プロセス優先度によるプロセッサ割り当て制御法がある。しかし、プロセスの処理やプロセスが発行したシステムコール処理は、この制御法によって負荷分散できるが、割り込み処理はこれらの処理に比べ優先して実行されるため負荷分散できない。このため、オンライン処理の応答時間を保証するには、プロセス優先度による優先制御だけでは不十分であり、入出力処理を制御する方法に工夫が必要である。

また、ディスクドライブはI/O要求を並行して実行することができないため、I/O要求の処理時間が長いバッチ処理が実行された場合、オンライン処理のI/O要求は待たさ

れてしまう。これは、ディスクドライバは、ディスク装置の I/O 要求の完了による割り込み制御によって、I/O 要求の実行が完了した後の処理を開始するためである。このため、オンライン処理の I/O 要求は、バッチ処理の I/O 要求の完了を待たされる。この様子を図 5 に示す。図 5 では、オンライン処理 (CPU) とバッチ処理 (CPU) は、プロセッサ資源を利用し、計算処理を実行している。バッチ処理 (CPU) は、I/O 要求をディスクドライバに依頼し、ディスクドライバが I/O 要求を実行する。少し遅れてオンライン処理 (CPU) も同様に I/O 要求をディスクドライバに依頼する。しかし、このときディスクドライバはバッチ処理の I/O 要求を実行しているため、オンライン処理 (CPU) の I/O 要求は待たされる。バッチ処理 (CPU) の I/O 要求が完了した後、オンライン処理 (CPU) の I/O 要求が実行される。オンライン処理の I/O 要求が、先に実行されているバッチ処理の I/O 要求と重なった場合、オンライン処理に対するプロセス優先度によるプロセッサ割り当て制御によって、オンライン処理が優先的に処理されていた場合でも、バッチ処理の I/O 要求の完了までは、オンライン処理の I/O 要求は実行されない。

したがって、バッチ処理の I/O 要求に基づく割り込み処理の影響やディスクドライバ処理 (実 I/O 時間も含む) の影響を抑制する入出力制御が必要である。

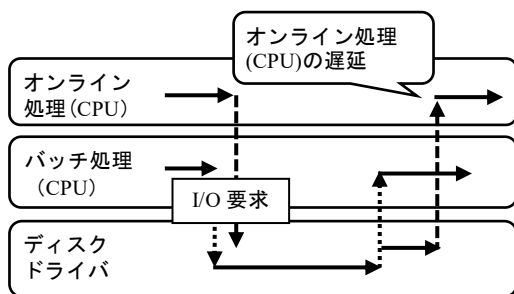


図 5 I/O 要求と処理の遅延

3. データ長を考慮したディスク I/O 制御機構

3.1 基本機能

図 4 に示したように、書き込みデータ長は、オンライン処理がバッチ処理に比べ非常に小さい。そこで、書き込みデータ長が小さい I/O 要求を優先的に実行するディスク I/O 制御方式を述べる。この方式の基本構造を図 6 に示し、以下に説明する。

(1) I/O サイズ分割機能

書き込みデータ長が非常に大きいものについて、分割し (図 6 では「大サイズ」としている)、一つ一つの I/O 要求の処理時間を短くする。これにより、書き込みデータ長が非常に大きいバッチ処理の I/O 要求の処理時間を分割して小さくできるため、オンライン処理への影響を抑制できる。

(2) I/O 処理キュー制御機能

I/O サイズの違い毎に実行待ちキューを用意して管理す

る。各キューは I/O 要求到着順である。小サイズキューは、I/O サイズが小さい I/O 要求のキューであり、オンライン処理の I/O 要求に相当する。一方、大サイズキューは、I/O サイズ分割機能によって分割された I/O 要求のキューであり、バッチ処理の I/O 要求に相当する。

(3) I/O 優先制御機能

小サイズキューの I/O 要求を大サイズキューの I/O 要求より優先させて、ディスクドライバに I/O 要求の実行を依頼する。つまり、I/O サイズが小さい I/O 要求を優先して実行する。なお、ディスクドライバ処理中に新たな I/O 要求が発生しても、優先制御を確実にを行うために、I/O 優先制御がディスクドライバに依頼する処理は一度に一つである。

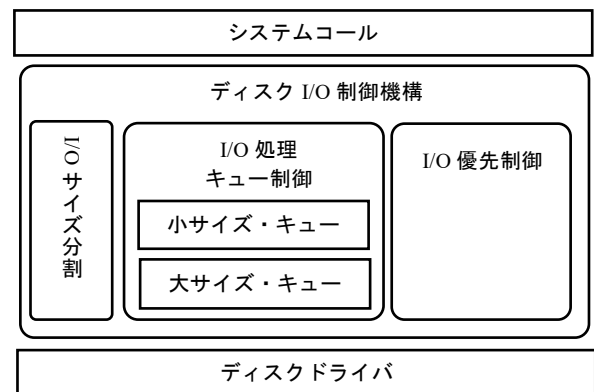


図 6 ディスク I/O 制御方式の基本構造

3.2 処理流れ

I/O 制御機構の処理流れを図 7 に示し、以下に説明する。

① システムコール発行により、OS に I/O 要求の実行が

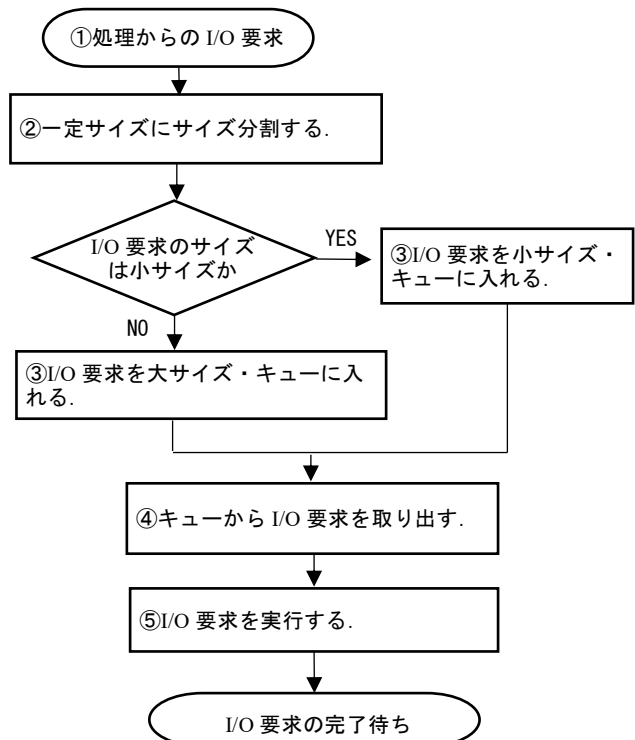


図 7 ディスク I/O 制御機構の処理流れ

要求される。

- ② I/O 要求の書き込みデータ長が大きい場合、一定サイズにサイズ分割する。
- ③ I/O 要求のサイズにもとづきキューに入れる。
- ④ キューから I/O 要求を取り出す。優先度の高い小サイズ・キューから優先的に I/O 要求を取り出す。
- ⑤ I/O 要求を実行し、処理結果を待つ。I/O 要求は、その割り込み処理により、呼び出し元に戻る。

4. 評価

4.1 観点

ディスク I/O 制御方式の有効性を以下の観点で評価する。
(評価 1) プロセス優先度によるプロセッサ割り当て制御ではオンライン処理の I/O 要求の処理時間が短くならないこと、すなわち I/O 要求の優先制御の方がオンライン処理の I/O 要求の処理時間が短くなること。
(評価 2) オンライン処理の負荷 (TPS) とオンライン処理の処理時間の関係を明らかにすること。I/O 要求の優先制御を使用したオンライン処理の処理時間が、プロセス優先度によるプロセッサ割り当て制御を使用したオンライン処理の処理時間よりも短くなること。
また、評価対象とする優先制御方式は、以下の3つである。

① I/O 要求優先 (オンライン処理優先)

ディスク I/O 制御方式を使用した優先制御によって、オンライン処理の I/O 要求を優先的に実行する。

② CPU 優先 (オンライン処理優先)

プロセス優先度によるプロセッサ割り当て制御によって、オンライン処理を優先的に実行する。オンライン処理の I/O 要求は優先的に実行しない。

③ 優先なし

I/O 要求の優先制御およびプロセス優先度によるプロセッサ割り当て制御を行わない。

4.2 評価環境

ディスク I/O 制御方式の評価に使用する計算機の諸元を表 1 に示す。オンライン処理およびバッチ処理がデータを書き出すために使用するデータ用ディスクは、ファイルシステムのキャッシュの影響をなくするため、UFS のキャッシュを使用しない設定とする。

表 1 測定に使用した計算機環境の諸元

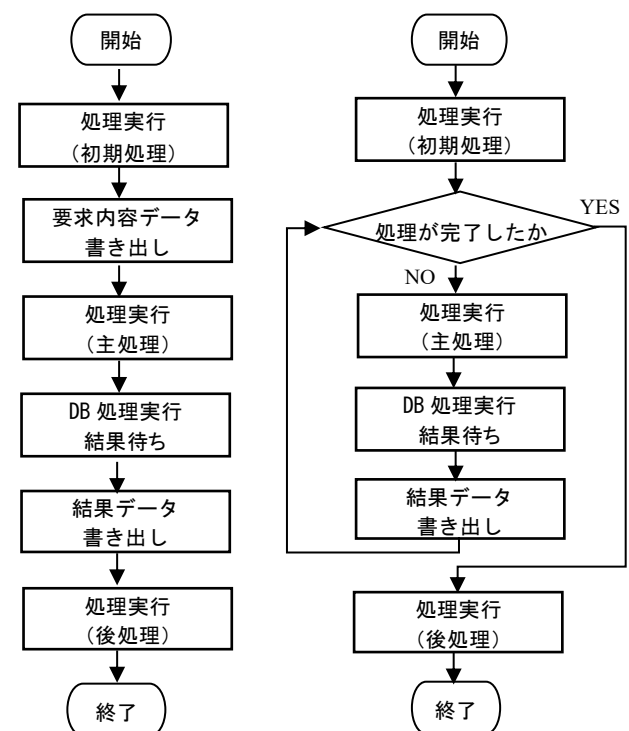
CPU	Intel i5 3.2GHz 4 コア ハイパースレッドなし	
データ用 ディスク	I/O バス	SATA3.0
	DISK	7400 回転 32MB キャッシュ
	ファイルシステム	UFS
	ファイルシステム のキャッシュ	使用しない
OS	FreeBSD 11.0-RELEASE-p1	

4.3 評価プログラム

4.3.1 処理内容

ディスク I/O 制御方式を評価するため、オンライン処理とバッチ処理の評価用プログラムを使用する。評価用プログラムの処理流れを図 8 に示す。図 8(A)は、オンライン処理の 1 トランザクション処理を擬似した処理 (オンライン処理) であり、処理実行 (初期処理)、要求内容データ書き出し、処理実行 (主処理)、DB 処理実行結果待ち、結果データ書き出し、処理実行 (後処理) からなる。図 8(B)は、バッチ処理を擬似した処理 (バッチ処理) であり、処理実行 (初期処理)、処理実行 (主処理)、DB 処理実行結果待ち、結果データ書き出し、処理実行 (後処理) からなり、処理実行 (主処理)、DB 処理実行結果待ち、結果データ書き出しの処理を繰り返す。

書き出しデータ長は、オンライン処理を 1,000 バイト、バッチ処理を 1,000,000 バイトとする。オンライン処理の処理流れは、オンライン処理の処理時間実行 (初期処理) と処理実行 (後処理) をそれぞれ 10 ミリ秒、処理実行 (主処理) を 50 ミリ秒、DB 処理実行結果待ちを 50 ミリ秒とし、全体の処理時間を 120 ミリ秒とする。表 1 の計算機を使用した処理時間は、オンライン処理の同時起動数 20 の場合、オンライン処理の平均処理時間は 193 ミリ秒、平均 CPU 使用率は 40% であり、図 4(A) のオンライン処理の処理時間相当である。バッチ処理の処理流れは、処理実行 (初期処理) を 100 ミリ秒、処理実行 (主処理) を 50 ミリ秒、



(A) オンライン処理 (1 トランザクション)

(B) バッチ処理

図8 評価プログラムの処理流れ

DB 処理実行結果待ちを 100 ミリ秒、処理実行（後処理）を 100 ミリ秒とし、全体の処理時間が 30 秒程度となるように、繰り返し回数を 150 回とした。表 1 の計算機において実行した処理時間は、バッチ処理の同時起動数 4 の場合、バッチ処理の平均処理時間は 28 秒、平均 CPU 使用率は 30% であり、図 4(B) のバッチ処理の平均処理時間相当である。

4.3.2 起動条件

オンライン処理とバッチ処理の起動は、オンライン処理の負荷、および、バッチ処理数が多い場合と少ない場合の組み合わせとする。また、両処理の同時起動により、プロセッサ資源がボトルネックとなることを避けるため、最大の同時起動数は、CPU 使用率が 90% 程度になるように決める。4.3.1 節の評価用プログラムの処理時間を考慮したオンライン処理とバッチ処理の起動条件を以下に示す。

(1) オンライン処理

- (A) オンライン処理の到着は、0~3 秒の間隔でランダムに到着するものとする。
- (B) 同時起動数は 40, 30, 20, 10 とする。

(2) バッチ処理

- (A) バッチ処理は、同時に起動したバッチ処理の完了後、次のバッチ処理を同時に起動する。
- (B) バッチ処理の同時起動数は、多い場合として 4、少ない場合として 2 とする。

4.4 結果と考察

4.4.1 I/O 要求の処理時間

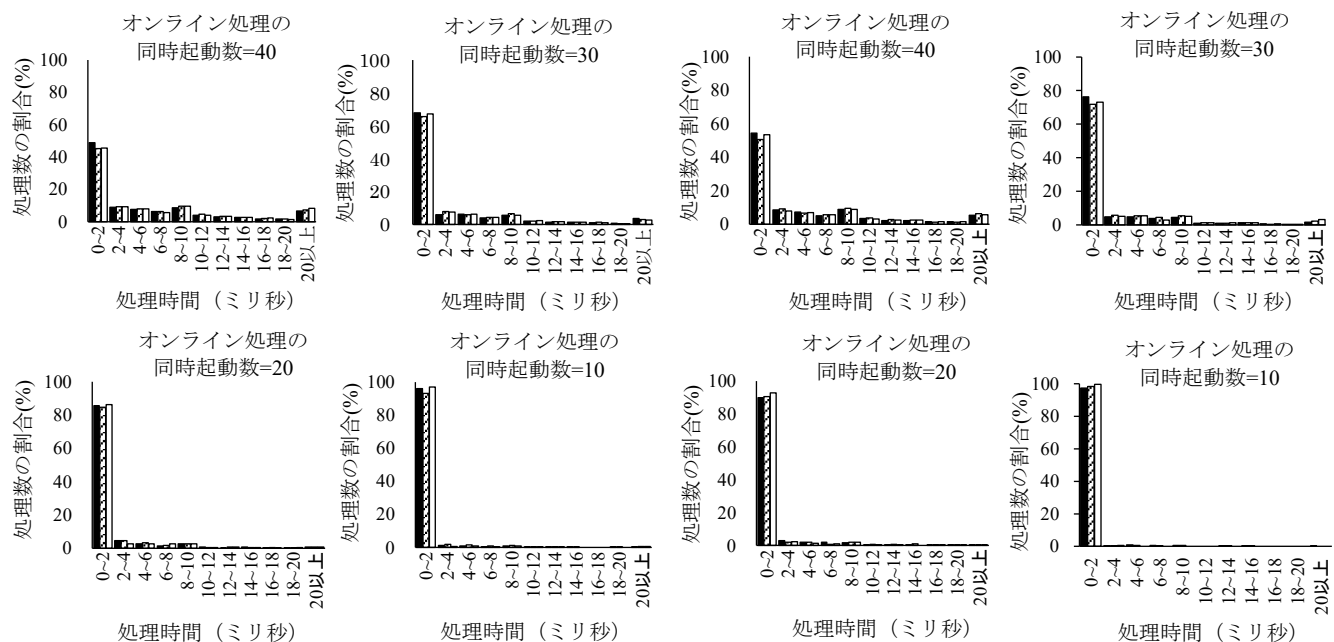
オンライン処理の I/O 要求の処理時間の分布を図 9 に示す。I/O 要求優先、CPU 優先、優先なしに関係なく、かつ

バッチ処理の同時起動数やオンライン処理の同時起動数に関わらず、I/O 要求の処理時間の大半は 2 ミリ秒未満である。なお、オンライン処理の同時起動数が多い時 (40, 30) は、I/O 要求の処理時間が 2 ミリ秒から 10 ミリ秒の間である場合も見られるものの、I/O 要求の処理時間が 10 ミリ秒以上である場合は非常に少ない。そこで、I/O 要求の処理時間を、2 ミリ秒未満 (区分 1)、2 ミリ秒から 10 ミリ秒 (区分 2)、10 ミリ秒以上 (区分 3) に区分し、その割合を図 10 に示す。また、各起動条件 (バッチ処理やオンライン処理の同時起動数) と各区分における I/O 要求の処理時間の平均値 (以降、平均 I/O 処理時間と呼ぶ) を表 2 に示す。なお、同じ条件と区分において、平均 I/O 処理時間が最も短いものを網掛けしている。

図 10 より、以下のことがわかる。

(1) オンライン処理の同時起動数が少ない、つまりオンライン処理の負荷 (TPS) が小さいほど、オンライン処理の I/O 処理時間は区分 1 の割合が大きい。この現象は、I/O 要求優先、CPU 優先、優先なしに関係ない。このことは、バッチ処理の同時起動数が異なっても同様である。例えば、オンライン処理の同時起動数 30 かつバッチ処理の同時起動数 2 の時、区分 1 は 71% 以上である。

(2) オンライン処理の同時起動数が多い、つまりオンライン処理の負荷 (TPS) が大きいと、オンライン処理の I/O 要求の処理時間は区分 1 の割合が減少し、区分 2 の割合が増える。この現象は、I/O 要求優先、CPU 優先、優先なしに関係ない。このことは、バッチ処理の同時起動数が異なっても同様である。例えば、オンライン処理の同時起動数 40

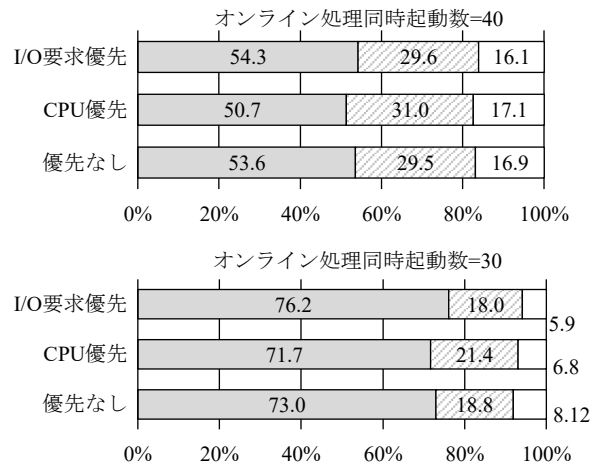
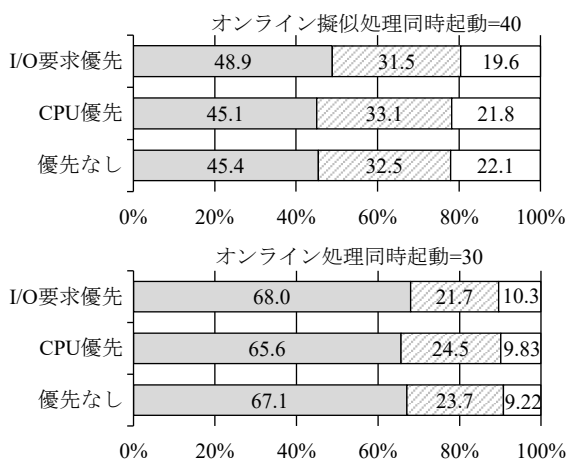


(A) バッチ処理の同時起動数=4

(B) バッチ処理の同時起動数=2

- I/O 要求優先
- ▨ CPU 優先
- 優先なし

図 9 オンライン処理の I/O 要求の処理時間分布



(A) バッチ処理の同時起動数=4

(B) バッチ処理の同時起動数=2

- 区分1 0~2 ミリ秒
- ▨ 区分2 2~10 ミリ秒
- 区分3 10 ミリ秒以上

図 10 オンライン処理の I/O 要求の処理時間区分ごとの割合

かつバッチ処理の同時起動数 4 の時、区分 1 が 45%以上である。しかし、区分 2 も含めると約 80%であり、区分 1 と区分 2 の和は大きい。

さらに、表 2 も含め、以下のことがわかる。

(3) オンライン処理の同時起動数 40 かつバッチ処理の同時起動数 4 の時、I/O 要求優先の平均 I/O 処理時間は、以下の状況にある。

(A) 区分 1 では、優先なしに比べ 10.8% (0.074 ミリ秒) 短く、CPU 優先に比べ 13.2% (0.093 ミリ秒) 短い。なお、CPU 優先は、優先なしよりも長くなり悪化している。

(B) 区分 2 では、優先なしに比べ 0.67% (0.004 ミリ秒) であり、CPU 優先に比べ 0.67% (0.004 ミリ秒) であり、同等である。

したがって、オンライン処理の約 8 割 (区分 1 と区分 2 を合わせると約 80%) において、I/O 要求優先の平均 I/O 処理時間は優先なしや CPU 優先より短いか同等である。以上のことから、オンライン処理の負荷 (TPS) が大きい時、提案する I/O 要求優先の制御を行うことでオンライン処理の平均 I/O 処理時間を短くできるといえる。一方、CPU 優先の制御では I/O 処理時間が悪化する (長くなる) ことがある。

(4) 上記 (3) に比べバッチ処理の同時起動数が少ない時は、オンライン処理の同時起動数が 40 だけではなく 30 の時も、オンライン処理の約 8 割から 9 割強 (区分 1 と区分 2 を合わせると約 81%から 94%) において、I/O 要求優先の平均 I/O 処理時間は、優先なしや CPU 優先より短い。つまり、バッチ処理が少ない時、提案する I/O 要求の優先制御を行うことでオンライン処理の平均 I/O 処理時間を小さくできるといえる。

4.4.2 オンライン処理とバッチ処理の処理時間

(1) オンライン処理の処理時間

図 11 にオンライン処理の負荷と処理時間の関係を示す。オンライン処理の負荷が大きい時、I/O 要求優先のオンライン処理の処理時間は、以下の状況である。

(A) バッチ処理数が多い時、オンライン処理の負荷 22.4TPS (I/O 要求優先), 22.4TPS (CPU 優先) では、CPU 優先に比べ 2.2% (6.6 ミリ秒) 短い。しかし、オンライン処理の負荷 17.7TPS (I/O 要求優先), 17.2TPS (CPU 優先) では、CPU 優先に比べ 2.9% (6.8 ミリ秒) 長い。

(B) バッチ処理数が少ない時、オンライン処理の負荷 22.8TPS (I/O 要求優先), 22.6TPS (CPU 優先) では、CPU

表 2 オンライン処理の負荷と I/O 要求の処理時間

バッチ処理の同時起動数		4						2					
		40			30			40			30		
オンライン処理の同時起動数		40			30			40			30		
オンライン処理の負荷 (TPS)		22.4	22.4	22.5	17.7	17.2	17.5	22.8	22.6	22.5	17.4	17.9	17.6
優先制御方式		I/O要求優先	CPU優先	優先なし	I/O要求優先	CPU優先	優先なし	I/O要求優先	CPU優先	優先なし	I/O要求優先	CPU優先	優先なし
オンライン処理の平均 I/O 処理時間 (ミリ秒)	(区間1) 0~2ミリ秒	0.613	0.706	0.687	0.544	0.604	0.591	0.508	0.531	0.519	0.430	0.459	0.458
	(区間2) 2~10ミリ秒	5.93	5.97	5.97	5.79	5.74	5.67	5.97	6.00	6.07	5.80	5.85	5.83
	(区間3) 10ミリ秒以上	20.6	20.4	21.1	20.0	18.6	21.7	20.8	21.5	20.6	20.6	19.4	21.1

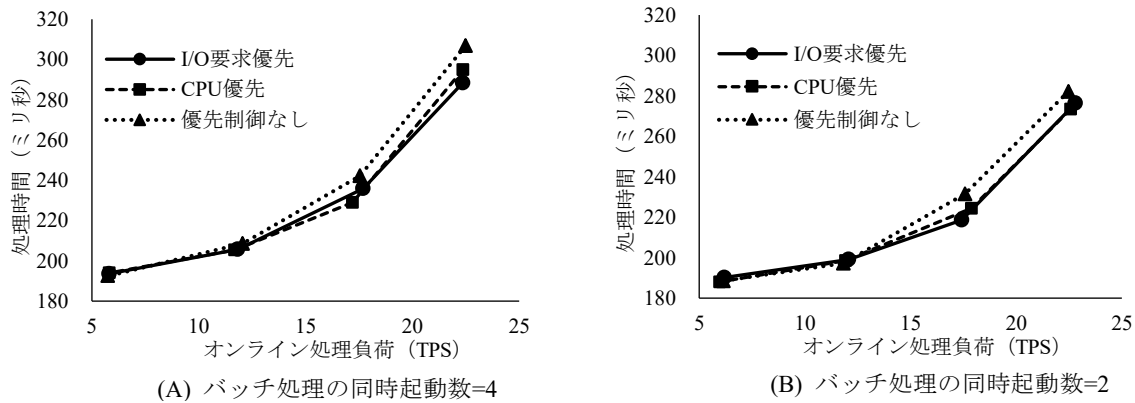


図 11 オンライン処理の負荷に対する処理時間

優先に比べ 1.0% (2.9 ミリ秒) 長い。しかし、オンライン処理の負荷 17.4TPS (I/O 要求優先), 17.9TPS (CPU 優先) では、バッチ処理が少ない時 (同時起動数 2), CPU 優先に比べ 2.6% (5.9 ミリ秒) 短い。つまり、バッチ処理が少ない場合、I/O 要求優先と CPU 優先の間に明確な差はない。

オンライン処理の負荷が小さい時、I/O 要求優先のオンライン処理の処理時間は、以下の状況である。

(C) バッチ処理が多い時、および、少ない時において、I/O 要求優先と CPU 優先の処理時間の差は、0.17~1.1% (0.34~2.2 ミリ秒) であり、ほとんど差はない。

つまり、オンライン処理の負荷が大きく、バッチ処理数が多いとき、I/O 要求優先により、オンライン処理の処理時間を短くできるといえる。また、オンライン処理の負荷が小さく、バッチ処理数が少ないとき、I/O 要求優先と CPU 優先のいずれも処理時間に対して、オンライン処理の処理時間は優先制御なしとほとんど変わらない。

(2) バッチ処理の処理時間

表 3 に同時起動数 4 と 2 のバッチ処理の処理時間および平均 I/O 処理時間を示す。すべてのオンライン処理の同時起動数において、I/O 要求優先の処理時間および平均 I/O 処理時間が、CPU 優先および優先なしと比べて大きい。これは、I/O 要求優先によって、オンライン処理の I/O 要求が優先的に実行された結果、バッチ処理の I/O 要求の短期の時間的な負荷分散が行われたためである。また、処理時間が長くなった割合は、概ね 5~6% である。時間に換算すると、例えば処理時間が 1 時間のバッチ処理の場合、概ね 3.6 分

に相当し、計画的に実行されるバッチ処理に許容される処理時間において、問題となる時間ではない。

5. 関連研究

文献[2][3][4][5]は、I/O 要求に対するシーク時間や回転待ち時間により、I/O 要求の処理時間が大きくなることに着目し、書き込みデータのディスク上の書き込み位置などの情報と I/O 要求の処理時間を予測し、I/O 要求を効率的に実行するスケジューリング手法を提案している。文献[6]は、入出力スループットの向上を目的として、入出力要求を並び替えて HDD のシーク時間を最小とする I/O 要求のスケジューリング手法を提案している。また、近年普及している SSD は入出力要求の処理時間が短いため、OS と入出力デバイス間の入出力要求の授受に要する時間が入出力時間に影響を与えやすい。文献[7]は入出力要求をまとめて発行することで授受の回数を削減している。文献[8][9]は、SSD ではメモリセルへの読み書きにおいて排他が必要となるため、読み書き対象の領域が分散するように入出力要求を並び替えて排他の発生を防いでいる。これらの方法では、オンライン処理の I/O 要求をバッチ処理の I/O 要求より優先させて実行することはできない。

文献[10]は、アプリケーションの I/O 要求の実行にあたって、I/O 要求の優先度、I/O 要求の開始時間と終了時間をもとにして、優先的に実行する I/O 要求を決定し、それらの I/O 要求を複数まとめ、スループットの向上するために I/O 要求の実行順番を入れ替える方式を提案している。こ

表 3 バッチ処理の処理時間と I/O 要求の処理時間

バッチ処理の同時起動数	4						2					
	40			30			40			30		
オンライン処理の同時起動数												
オンライン処理の負荷(TPS)	22.4	22.4	22.5	17.7	17.2	17.5	22.8	22.6	22.5	17.4	17.9	17.6
優先制御方式	I/O要求優先	CPU優先	優先なし	I/O要求優先	CPU優先	優先なし	I/O要求優先	CPU優先	優先なし	I/O要求優先	CPU優先	優先なし
バッチ処理の処理時間(秒)	61.1	58.0	57.3	40.6	38.3	38.4	47.1	47.8	45.6	34.0	33.6	34.1
バッチ処理の平均 I/O 処理時間 (ミリ秒/回)	7.67	6.74	6.64	3.41	2.87	2.90	4.69	4.80	4.35	2.11	1.98	2.09

の方法では、オンライン処理の I/O 要求をまとめて実行することになり、オンライン処理の応答時間の最大値が大きくなり好ましくない。

リアルタイム処理と QoS 保証を実現する場合として、I/O 要求の優先度制御が提案されている。文献[11]は、リアルタイムシステムにおいて、サービス時間を短縮することとデッドラインを守ることを実現するため、I/O 要求の優先度とシーク時間を基にした評価関数を用いて、実行する I/O 要求を決定することで、リアルタイム処理のデッドラインを守りながらサービス時間の短縮を図る手法を提案している。また、文献[12]は、リアルタイム処理とベストエフォートな処理が混在する環境において、I/O 要求を実行するスケジューラが、リアルタイム処理のデッドラインなどの情報にもとづき、各 I/O 要求の処理時間を予測し、その予測にもとづいて、ベストエフォート処理およびリアルタイム処理の I/O 要求の優先度を決定する手法を提案している。文献[13]は、デッドライン制御に関して Deferrable Schedule と組み合わせることで平均応答時間を改善している。文献[14][15]は、ストレージスタック（ファイルシステムレベル、ブロックレベル、デバイスレベルなど）における各階層の I/O 要求の優先度制御に関して、各階層にある I/O 要求の情報（処理時間や優先度、デッドラインなど）をそれぞれの階層の優先度制御で利用する制御方法を提案している。しかし、いずれの提案もデッドラインを規定しスケジュールできることが前提となっており、負荷変動が大きく、かつデッドラインを規定し難いオンライン処理には適さない。

文献[16]は、I/O 要求の処理時間を調整する制御法を提案している。この提案では、高い優先度を持つ I/O 要求が優先的に実行されるように I/O 要求の処理時間を調整することで、優先度の高い処理の I/O 要求の完了待ちによる待ち時間を短くできることを示している。しかし、この方法はスループットが低い問題がある。

6. おわりに

オンライン処理とバッチ処理が混在して実行する環境において、オンライン処理の I/O 要求の書き込みデータ長が小さいことに着目し、書き込みデータ長を考慮したディスク I/O 制御方法を提案した。バッチ処理の起動数が多く、オンライン処理の負荷が大きい場合、プロセス優先度によるプロセッサ割り当て制御法では、I/O 要求の処理時間が悪化することがあることを示し、ディスク I/O 制御方式によって I/O 要求の処理時間を短くできることを示した。

ディスク I/O 制御方式によって、バッチ処理の I/O 要求に対して、短期の時間的な負荷分散を行うことで、バッチ処理によるオンライン処理の応答時間への影響を軽減し、バッチ処理の起動数が多く、オンライン処理の負荷が大きい場合に、オンライン処理の処理時間を改善できることを示した。

参考文献

- [1] 田辺雅則, 横山和俊, 長尾尚, 乃村能成, 谷口秀夫, “オンライン処理とバッチ処理が混在する環境におけるディスク I/O 制御方式,” 情報処理学会研究報告システムソフトウェアとオペレーティングシステム Vol. 2018-OS-143, No.4, 2018/5/14.
- [2] Lan Huang and Tzi-cker Chiueh, “Implementation of a Rotation-Latency-Sensitive Disk Scheduler,” Technical Report ECSL-TR81, SUNY, Stony Brook University at New York, Mar. 2000.
- [3] Sitaram Iyer and Peter Drushel, “Anticipatory scheduling: A disk scheduling framework to overcome deceptive idleness in synchronous I/O,” Proc. of 18th ACM Symposium on Operating Systems Principles, Oct. 2001.
- [4] Paolo Valente and Fabio Checconi, “High Throughput Disk Scheduling with Fair Bandwidth Distribution,” IEEE Transactions on Computers, Vol. 59, No. 9, pp.1172-1186, Sep. 2010.
- [5] 田邨 優人, 中島 耕太, 山本 昌生, 前田 宗則, “CPU 処理と I/O 処理の高速性を両立する I/O 制御機構,” 情報処理学会研究報告, システムソフトウェアとオペレーティングシステム, Vol. 2017-OS-141, No. 22, 2017/7/19.
- [6] G. Basavaraju, P. K. Kondugari and Shankaraiah, “Accelerating the performance of Disk with Re-Ordering of An Input/Output Requests at Virtual Machine Monitor Level,” Proceedings of IEEE International Conference on Computer Communication and Systems (ICCCS14), pp.245-247, Feb. 2014.
- [7] Y. Son, H. Y. Yeom and H. Han, “Optimizing I/O Operations in File Systems for Fast Storage Devices,” IEEE Transactions on Computers, Vol.66, No.6, pp.1071-1084, June 2017.
- [8] Myung Hyun Jo and Won Woo Ro, “Dynamic Load Balancing of Dispatch Scheduling for Solid State Disks,” IEEE Transactions on Computers, Vol.66, No.6, pp.1034-1047, June 2017.
- [9] S. He, Y. Wang, X. Sun, C. Huang and C. Xu, “Heterogeneity-Aware Collective I/O for Parallel I/O Systems with Hybrid HDD/SSD Servers,” IEEE Transactions on Computers, Vol.66, No.6, pp.1091-1098, June 2017.
- [10] J. Bruno, J. Brustoloni, E. Gabber, M. McShea, B. Ozden and A. Silberschatz, “Disk Scheduling with Quality of Service Guarantees,” Proceedings IEEE International Conference on Multimedia Computing and Systems, 7-11 June 1999.
- [11] S. Chen, John A. Stankovic, James F. Kurose and D. Towles, “Performance Evaluation of Two New Disk Scheduling Algorithms for Real-Time Systems,” Real-Time Systems, Vol. 3, Issue 3, pp 307-336, Sep. 1991.
- [12] Damien Le Moal, Donald Molaro and Jorge Campello, “A Real-Time File System for Constrained Quality of Service Applications,” IPSJ Transactions on Advanced Computing Systems, Vol. 3, No. 1, 61-76 (Mar. 2010).
- [13] S. Han, D. Chen, M. Xiong, Kam-Yiu Lam, Aloysius K. Mok and K. Ramamritham, “Schedulability Analysis of Deferrable Scheduling Algorithms for Maintaining Real-Time Data Freshness,” IEEE Transactions on Computers, Vol.63, No.4, pp.979-994, Apr. 2014.
- [14] S. Kim, H. Kim, J. Lee and J. Jeong, “Enlightening the I/O Path: A Holistic Approach for Application Performance,” the 15th USENIX Conference on File and Storage Technologies (FAST '17), pp.345-358, Mar. 2017.
- [15] S. Yang, T. Harter, N. Agrawal, S. S. Kowsalya, A. Krishnamurthy, S. Al-Kiswany, R. T. Kaushik, A. C. Arpaci-Dusseau and R. H. Arpaci-Dusseau, “Split-level I/O Scheduling,” Proceeding SOSP '15 Proceedings of the 25th Symposium on Operating Systems Principles p.474 - 489, Monterey, California, Oct. 4-7, 2015.
- [16] 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, D, Vol. J94-D No.7 pp.1047-1057, 2011-07-01.