

オンライン処理とバッチ処理の混在環境において バッチ処理の影響を軽減するディスク I/O 制御方式の評価

田辺雅則^{†1} 乃村能成^{†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毎日実行される。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、両処理の資源の利用量は時間帯で異なるため、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、バッチ処理の入出力要求の割り込み処理の影響を受けるため、プロセス優先度によるプロセッサ割り当て制御だけでは、オンライン処理の処理時間に影響を与えてしまう。著者らは、ディスク I/O、特に両処理の書き込みデータ長が異なることに着目し、ディスクに対する I/O 要求の優先度を制御するディスク I/O 制御方式を提案し、オンライン処理に対するバッチ処理の影響を軽減できることをシミュレーションにより示した。本稿では、プロセッサ数の多い計算機における本制御方式の効果について報告する。

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

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に分類される処理が毎日実行される。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。

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ため、両処理を同じ計算機で実行するシステム構成とすることで、分割損を軽減し、計算機資源の利用効率の向上が期待できる。両処理を同じ計算機で実行する場合、オンライン処理の応答時間に与える影響を軽減する必要がある。このため、バッチ処理よりもオンライン処理を優先して実行するための優先制御が重要である。

計算機資源の利用量を制御する方法として、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。本制御法では、割り込み処理は、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に比べて優先して実行されるため、ディスクへの I/O 要求が多く発生している場合、応答時間に影響を与える問題がある。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が依頼した割り込み処理より優先して実行される。したがって、入出力要求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。

著者らは、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、ディスク I/O 制御方式の有効性をオンライン処理とバッチ処理の簡易モデルを使用してシミュレーションによって評価した[1]。また、ディスク I/O 制御方式を FreeBSD に実装し、オンライン処理とバッチ処理をモデル化したアプリケーションを使用し評価した[2]。しかし、実システムでは、プロセッサ数の多い計算機が使用されるため、プロセッサ数の多い計算機における評価が必要である。

本稿では、計算機のプロセッサ数が多くなると、本制御方式がより有効になることをシミュレーション評価により示す。具体的には、プロセッサ数の多い計算機において、オンライン処理とバッチ処理のアプリケーションモデルを使用し、シミュレーションによって、オンライン処理の I/O 要求が実行されるまでの待ち時間の短縮と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. データ長を考慮したディスク I/O 制御機構

2.1 課題

オンライン処理とバッチ処理を別の計算機で実行するシステム構成では、計算機資源の利用効率が低い時間帯と高い時間帯が発生し、計算機資源の分割損が発生する。オンライン処理とバッチ処理は、主に実行される時間帯が異なることから、両処理を混在して実行すること、つまり一つの計算機で実行することで、計算機資源の利用効率を向上させることができる。しかし、バッチ処理の増加に伴い、オンライン処理が多い昼間帯にもバッチ処理が実行することが必要になっている。両処理を同じ計算機で実行する場合、オンライン処理の応答時間に与える影響を軽減する必要がある。このため、以下の課題に対処する必要がある。

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。このため、バッチ処理の I/O 要求が実行されると、オンライン処理の I/O 要求はバッチ処理の I/O 要求の完了まで待たされる。この待ち時間は、バッチ処理

^{†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} 高知工科大学情報学群
School of Information, Kochi University of Technology

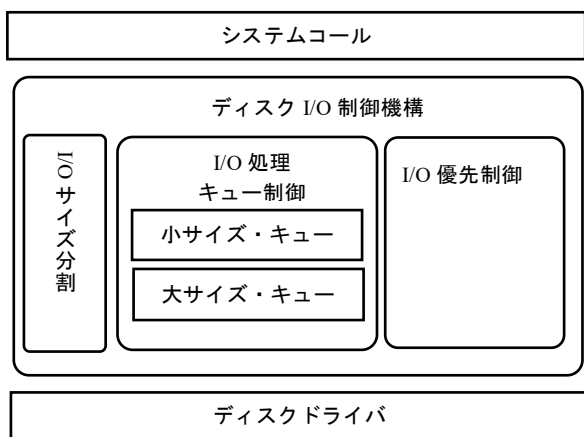


図1 ディスク I/O 制御方式の基本構造

の実 I/O 時間と I/O 要求完了後処理（割り込み処理）時間の和である。この待ち時間によるオンライン処理への影響を軽減するという課題がある。著者らは、書き込みデータ長を考慮したディスク I/O 制御方式を提案し評価した[1][2]。しかし、実システムでは、プロセッサ数の多い計算機が使用されるため、プロセッサ数の多い計算機における評価が必要である。

2.2 ディスク I/O 制御機構の基本構造

著者らが提案するデータ長を考慮したディスク I/O 制御方式の基本構造を図1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1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。

3. シミュレーションによる評価モデル

3.1 観点

I/O 要求が多数ある状態は、I/O 要求の実行待ちが多数発生するため、ディスクビジー率は高くなる。そこで、デ

表1 測定環境の計算機の諸元

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 パス	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	

スクビジー率が高い状況において、以下の2点を評価し、ディスク I/O 制御方式の有効性を示す。

（評価1）オンライン処理の負荷量と処理時間

（評価2）オンライン処理の負荷量と I/O 要求開始の待ち時間

なお、シミュレーションにおけるディスクビジー率は、処理時間を分析する評価対象時間に対して I/O 要求が割り当てられている時間の割合とする。

シミュレーションでは、以下の3つの制御方法に対して、前述の評価1と評価2を実施する、

- ①優先制御なし
- ②CPU 優先制御（オンライン処理優先）
- ③I/O 優先制御（オンライン処理優先）

3.2 計算機資源の割り当てモデル

3.2.1 I/O サイズ分割と計算機の諸元

I/O サイズ分割後の書き込みデータ長は、I/O 要求の処理時間が、1,000 バイトの I/O 要求の処理時間（書き込み）と同程度の処理時間となり、なるべく大きいデータサイズとする。具体的には、表1に示す測定環境において、1,000 バイトのデータ書き込み時間が 0.926 ミリ秒、10,000 バイトのデータ書き込み時間が 1.26 ミリ秒であったため、I/O サイズ分割による書き込みデータのデータサイズを 10,000 バイトとする。つまり、書き込みデータ長が大きい I/O 要求について、書き込みデータを 10,000 バイト単位に分割して、10,000 バイトずつ書き込む。

また、I/O サイズ分割を行わない場合、バッチ処理のデータ書き込み時間は、30,000 バイトあたり約 2.1 ミリ秒であった。そこで、I/O サイズ分割を行わない場合のデータ書き込み時間は、30,000 バイトあたり 2 ミリ秒とする。I/O 要求のデータ書き込み時間は、書き込みデータ長に比例したデータ書き込み時間とする。つまり、I/O サイズ分割を行わないため、例えば、書き込みデータ長が 1,000,000 バイトの場合は、データ書き込み時間は 66 ミリ秒である。

シミュレーションに用いる計算機モデルの諸元を表2に示す。計算機モデルは、プロセッサ数の少ない計算機（プロセッサ数が4個）とプロセッサ数の多い計算機（プロセッサ数が8個）の2種類とする。両計算機の I/O パス本数

表2 計算機モデル

			計算機 1	計算機 2
プロセッサ数			4	8
I/O パス本数			1	1
I/O 要求の処理時間（書き込み）	I/O サイズ分割あり	1,000 バイト	1 ミリ秒	1 ミリ秒
		10,000 バイト	1 ミリ秒	1 ミリ秒
		30,000 バイト	2 ミリ秒	2 ミリ秒
	I/O サイズ分割なし	1,000 バイト	1 ミリ秒	1 ミリ秒
		10,000 バイト	1 ミリ秒	1 ミリ秒
		30,000 バイト	2 ミリ秒	2 ミリ秒

は同じ本数とし、1本とする。また、I/O サイズ分割を行う場合の I/O 要求の処理時間（書き込み）は、1,000 バイトおよび 10,000 バイトあたり 1 ミリ秒とする。また、I/O サイズ分割を行わない場合の I/O 要求の処理時間（書き込み）は、1,000 バイトあたり 1 ミリ秒、30,000 バイトあたり 2 ミリ秒とする。

3.2.2 CPU と I/O 要求の割り当て

シミュレーションにおける CPU 割り当てと I/O 要求割り当ての規則を以下に示す。

(1) CPU 割り当ての規則

(A) 複数の処理が CPU の割り当てを待っている場合は、時間的に早く要求された処理を CPU に割り当てる。

(B) CPU のタイムスライスが 10 ミリ秒とする。10 ミリ秒単位で、次の実行待ちの処理に CPU を割り当てる。

(C) 処理の割り当てられていない空き CPU が複数ある場合、CPU の割り当てを待つ処理の CPU 割り当てが 1 回目の場合は、いずれかの空き CPU に処理を割り当てる。2 回目以降の場合は、1 回目に割り当てた CPU に割り当てる。つまり、2 回目以降の CPU 割り当ては、1 回目に割り当てた CPU に固定とする。

(2) I/O 要求割り当ての規則

(A) 同じ時間に複数の処理による I/O 要求があった場合、時間的に早く要求された I/O 要求を実行する。

(B) I/O のタイムスライスは 1 ミリ秒とする。1 ミリ秒単位で、次の実行待ちをしている処理の I/O 要求があれば、その I/O 要求を割り当てる。

3.2.3 優先度制御による CPU と I/O 要求の割り当て

CPU および I/O 割り当ての優先制御をシミュレーションに適用する。優先制御の規則を以下に示す。

(1) CPU 割り当ての優先制御の規則

(A) 同じ時間に複数の処理が CPU の割り当てを待つ場合、優先度の高い処理を CPU に割り当てる。同じ優先度の場合は、時間的に早く要求された処理を先に CPU に割り当てる。

表3 オンライン処理モデル

計算処理	初期処理時間	10 ミリ秒
	主処理時間	100 ミリ秒
	終了処理時間	10 ミリ秒
DB 処理時間 (主処理 1 回あたり)		50 ミリ秒
書き込みデータ長		1,000 バイト
主処理繰り返し回数		1 回
I/O 要求の実行間隔		初期処理後と終了処理前に各 1 回
総 I/O 要求数		2 回

表4 バッチ処理モデル

計算処理	初期処理時間	50 ミリ秒
	主処理時間	50 ミリ秒
	終了処理時間	50 ミリ秒
DB 処理時間 (主処理 1 回あたり)		100 ミリ秒
書き込みデータ長		1,000,000 バイト
主処理繰り返し回数		100 回
I/O 要求の実行間隔		DB 処理後に 1 回（主処理繰り返しごとに実行）
総 I/O 要求数		100 回

表5 オンライン処理の負荷量

	ケース 1	ケース 2	ケース 3
オンライン処理の負荷量 (TPS)	5	15	25

(2) I/O 割り当ての優先度制御の規則

(A) 同じ時間に複数の処理が I/O 要求があった場合、優先度の高い処理の I/O 要求を割り当てる。

(B) 実行待ちの I/O 要求の優先度を比較し、同じ優先度の場合は、直前まで実行中の I/O 要求を割り当てる。高い優先度の場合は、高い優先度の I/O 要求を割り当てる。

3.3 評価環境モデル

オンライン処理とバッチ処理の処理モデルは、評価用プログラムと同等の処理モデル[2]とする。表 3 にオンライン処理モデル（以降、オンライン処理と呼ぶ）を示し、表 4 にバッチ処理モデル（以降、バッチ処理と呼ぶ）を示す。また、評価に使用するオンライン処理の負荷量は、負荷量の多いケース、少ないケース、その中間のケースの 3 通りとする。表 5 にオンライン処理の 3 通りの負荷量を示す。また、バッチ処理モデルの同時起動数は 4 とする。

4. シミュレーション結果と考察

4.1 処理時間の評価 (評価 1)

オンライン処理とバッチ処理が混在して実行される区間を評価区間とする。シミュレーションの開始直後は、I/O 要求がオンライン処理に偏り、シミュレーションの終了間際はバッチ処理に偏る傾向があるため、シミュレーション開始後の 5 秒後から 15 秒までの 10 秒間を評価区間とする。

計算機 1 および計算機 2 の各ケースのオンライン処理とバッチ処理の処理時間をそれぞれ図 2 と図 3 に示し、ディスクビジー率と CPU 使用率を表 6 と表 7 に示す。図 2 と図 3 より以下のことがわかる。

(A) オンライン処理について、計算機 1 および計算機 2 の I/O 優先制御では、すべてのケースでオンライン処理の処理時間は、優先制御なしおよび CPU 優先制御より短い。計算機 1 の I/O 優先制御では、ケース 1、ケース 2、ケース 3 の順番で、ケース 1 が最も処理時間が短い。一方で、計算機 2 の I/O 優先制御では、すべてのケースで処理時間の差は小さい。また、計算機 1 のケース 3 では、I/O 優先制御と優先制御なしの処理時間の差と、I/O 優先制御と CPU 優先制御の処理時間の差は、ケース 1 およびケース 2 のときに比べて小さい。この理由は、表 6 よりケース 3 の CPU 使用率はケース 1 およびケース 2 に比べて高い CPU 使用率であることから、オンライン処理のプロセッサ競合が発生する割合が大きくなったためである。その結果として、ケース 3 のディスクビジー率は他のケースより小さくなっている。一方で、表 7 より計算機 2 のケース 3 の CPU 使用率は、表 6 の計算機 1 のケース 3 の CPU 使用率より低くなっている。プロセッサ数が増えると、プロセッサ競合が発

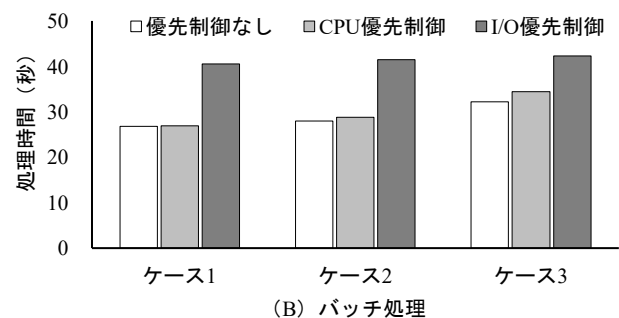
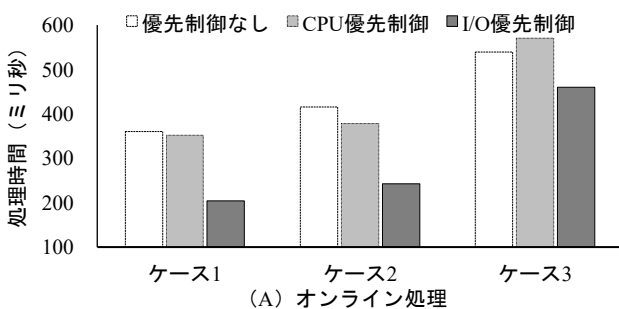


図 2 オンライン処理とバッチ処理の処理時間 (計算機 1)

表 6 ディスクビジー率と CPU 使用率 (計算機 1)

	ケース 1 単位 (%)		ケース 2 単位 (%)		ケース 3 単位 (%)	
	DISK	CPU	DISK	CPU	DISK	CPU
優先制御なし	100.0	33.7	97.6	63.1	87.2	90.8
CPU 優先制御	99.9	33.8	96.3	62.9	80.7	89.0
I/O 優先制御	100.0	27.4	100.0	57.3	100.0	86.9

表 7 ディスクビジー率と CPU 使用率 (計算機 2)

	ケース 1 単位 (%)		ケース 2 単位 (%)		ケース 3 単位 (%)	
	DISK	CPU	DISK	CPU	DISK	CPU
優先制御なし	100.0	33.2	100.0	46.2	99.5	54.6
CPU 優先制御	100.0	24.4	100.0	37.1	99.3	54.2
I/O 優先制御	100.0	27.4	100.0	53.8	100.0	56.9

生する割合が小さくなり、オンライン処理が実行される単位時間あたりの計算処理の割合が増える。その結果として、オンライン処理の I/O 要求が実行される割合が大きくなるため、計算機 2 のケース 3 のディスクビジー率は、計算機 1 のケース 3 よりも大きくなる。しかし、このケース 3 において、計算機 2 の I/O 優先制御のオンライン処理の処理時間は、ケース 1 およびケース 2 に対して増加していない。つまり、プロセッサ数が多く、オンライン処理の負荷が多い場合では、I/O 優先制御がより有効になると言える。

次に、計算機 1 のケース 3、計算機 2 のケース 2 では、

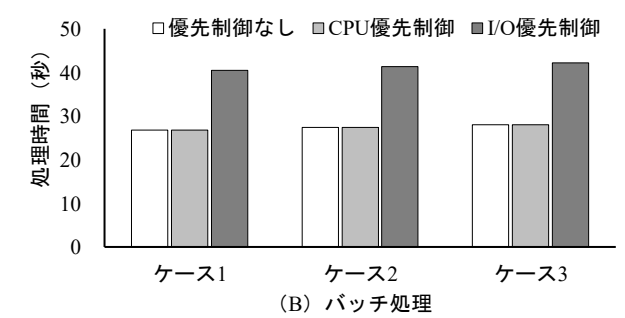
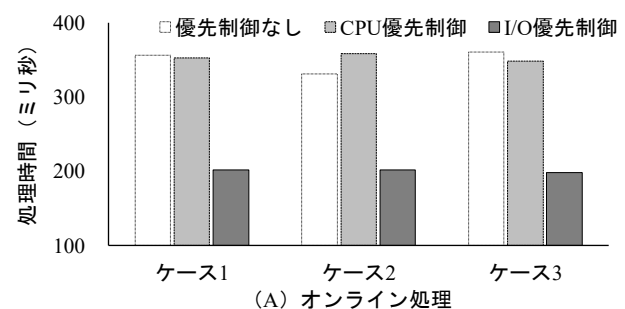


図 3 オンライン処理とバッチ処理の処理時間 (計算機 2)

CPU 優先制御のオンライン処理の処理時間は、優先制御なしより長い。つまり、CPU 優先制御だけでは、優先制御なしより処理時間が悪化する場合があることがわかる。また、計算機 1 の CPU 優先制御では、ケース 1、ケース 2、ケース 3 の順番で、ケース 3 が最も処理時間が長い。すなわち、オンライン処理の負荷量が多くなるにしたがって、オンライン処理の処理時間は長くなっている。一方、計算機 2 の CPU 優先制御では、処理時間は大きく変化していない。この理由は、計算機 1 では、オンライン処理の負荷量が多くなるにしたがって、オンライン処理のプロセッサ競合が発生する割合が高くなるためである。一方で、計算機 2 では、プロセッサ競合が発生する割合は低いためである。

(B) バッチ処理について、I/O 優先制御のバッチ処理の処理時間は、すべてのケースにおいて、優先制御なしと CPU 優先制御より長い。また、オンライン処理の負荷量が多い場合でも、バッチ処理の処理時間は微増である。また、計算機 1 では、CPU 優先制御のバッチ処理の処理時間は、優先制御なしよりも同等もしくは少し悪化している。これは、オンライン処理に対する CPU 優先制御のためである。なお、バッチ処理は、計画的に実行することにより、I/O 優先制御によるバッチ処理の処理時間の影響を軽減することが可能であり、大きな問題とはならない。

4.2 I/O 要求開始待ち時間の評価 (評価 2)

計算機 1 および計算機 2 における各ケースのオンライン処理とバッチ処理について、I/O 要求が開始されるまでの I/O 要求開始待ち時間をそれぞれ図 4 および図 5 に示す。図 4 および図 5 より以下のことがわかる。

(A) オンライン処理について、I/O 優先制御の I/O 要求開始待ち時間は、計算機 1 のケース 1、ケース 2、ケース 3 で

は、それぞれ 0.04 ミリ秒、0.3 ミリ秒、0.2 ミリ秒である。計算機 2 のケース 1、ケース 2、ケース 3 では、それぞれ 0.04 ミリ秒、0.7 ミリ秒、1.6 ミリ秒である。いずれのケースにおいても待ち時間の大きな増加は見られない。また、I/O 優先制御の I/O 要求開始待ち時間は、計算機 1 および計算機 2 のすべてケースにおいて、優先制御なしと CPU 優先制御より短い。

CPU 優先制御の I/O 要求開始待ち時間は、計算機 1 では、ケース 3 で優先制御なしより長く、計算機 2 では、ケース 1 とケース 2 で優先制御なしより長い。つまり、プロセッサ数が多い計算機 2 では、オンライン処理の負荷量が少ないとき、CPU 優先制御の I/O 要求開始待ち時間は、優先制御なしより長くなると言える。この理由は、計算機 2 では、CPU 優先制御によってオンライン処理を優先的に実行した結果、バッチ処理との I/O 要求の競合が発生した割合が多くなり、CPU 優先制御の I/O 要求開始待ち時間が優先制御なしの I/O 要求開始待ち時間より長くなったためである。一方、プロセッサ数の少ない計算機 1 では、オンライン処理の負荷量が多い場合に、同様な結果である。

(B) バッチ処理について、I/O 優先制御の I/O 要求開始待ち時間は、計算機 1 および計算機 2 のすべてのケースで、優先制御なしおよび CPU 優先制御より長い。計算機 1 では、I/O 優先制御の I/O 要求開始待ち時間は、ケース 3、ケース 2、ケース 1 の順番でケース 3 が最も短い。この理由は、オンライン処理の負荷量が多くなるにしたがって、オンライン処理のプロセッサ競合が発生する割合が増えた結果、オンライン処理の I/O 要求の単位時間あたりの実行数が減少し、バッチ処理の I/O 要求が実行される割合が増えたためである。一方で、計算機 2 では、ケース 1、ケース

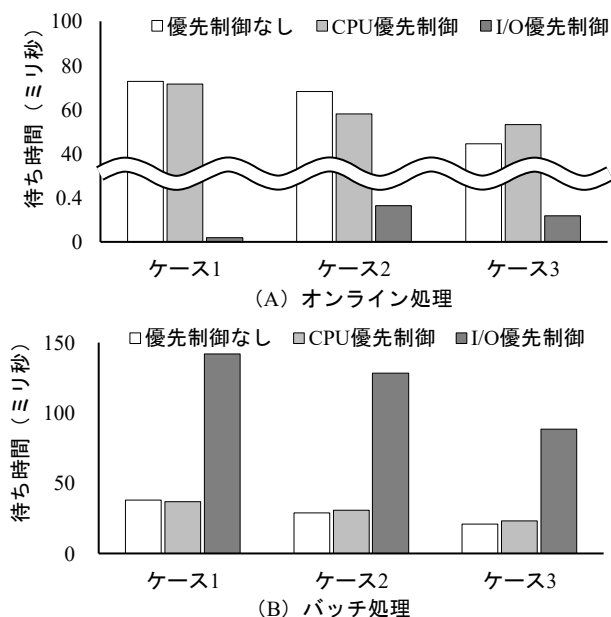


図 4 オンライン処理とバッチ処理の I/O 要求開始待ち時間 (計算機 1)

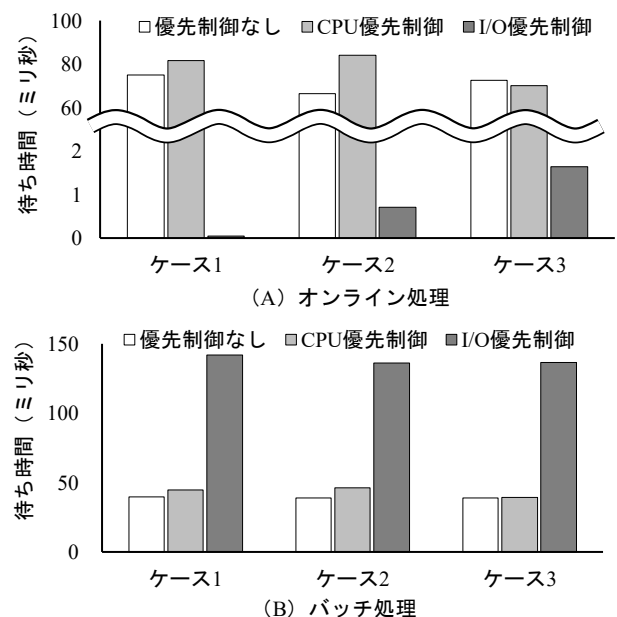


図 5 オンライン処理とバッチ処理の I/O 要求開始待ち時間 (計算機 2)

2, ケース 3 で, バッチ処理の I/O 要求開始待ち時間の差は小さい. この理由は, オンライン処理の I/O 要求が優先的に実行されたためである.

5. 関連研究

文献[3][4][5][6]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[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]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, スループットを向上するために 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. I/O 要求の処理時間を調整する制御法が提案されている[9]. 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, 応答時間およびスループットを重視する業務システムでは, I/O 要求の処理時間を調整することによる性能劣化が問題となる.

ディスクに対する I/O 要求に関して, 仮想環境における課題を解決する研究が行われている[10][11][12]. これらの研究では, 各仮想環境における I/O 要求の優先制御の有効性を明らかにしている. しかし, 仮想環境における I/O 要求の優先制御は, 各仮想環境の I/O の使用量を各仮想環境に対してあらかじめ割り当てた割合に近づけることや仮想環境の I/O 要求が他の仮想環境に与える影響を軽減することを目的としている. このため, アプリケーションに対する優先制御と目的が異なる.

6. おわりに

オンライン処理とバッチ処理の I/O 要求の実行を制御するディスク I/O 制御方式は, プロセッサ数の多い計算機においてもオンライン処理の処理時間を短くできることを示した. また, 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, CPU の優先制御だけではオンライン処理が優先的に処理されない場合があることを示した.

残された課題として, 仮想環境における本制御方式の有効性の評価がある.

参考文献

[1] 田辺雅則, 横山和俊, 乃村能成, 長尾尚, 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, 2018-OS-143, No.4, 2018/5/14.
- [2] 田辺雅則, 横山和俊, 長尾尚, 谷口秀夫, “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”, 情報処理学会研究報告マルチメディア通信と分散処理, 2019-DPS-178, No.12, 2019/02/25.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] 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.
- [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] 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.
- [9] 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.
- [10] Filip Blagojevic, Cyril Guyot, Qingbo Wang, Timothy Tsai, Robert Mateescu and Zvonimir Bandic, “Priority IO Scheduling in the Cloud,” 5th USENIX Workshop on Hot Topic in Cloud Computing, JUNE 25-26, 2013.
- [11] Mukil Kesavan, Ada Gavrilovska, Karsten Schwan, “On Disk I/O Scheduling in Virtual Machines,” The 2nd Workshop on I/O Virtualization (WIOV’ 10), March 13, 2010, USA.
- [12] Ziyi Yang, Haifeng Fang, Yingjun Wu, Chunqi Li, Bin Zhao, H. Howie Huang, “Understanding the Effects of Hypervisor I/O Scheduling for Virtual Machine Performance Interference,” 2012 IEEE 4th International Conference on Cloud Computing Technology and Science.