

OLTP 環境におけるディスクキャッシュ制御方法の モデル化と考察

3T-4

太田 義人* 佐伯 慎一* 松村 康伸**

*松下電器産業(株) メディア研究所 ** (株) 松下ソフトリサーチ

1 はじめに

昨今の情報化の進展にともない、オンライン・トランザクション処理(OLTP)は、座席予約システムなどに幅広く利用されている。また今後、その用途、利用者はますます増加していくと予想され、これらのシステムの処理性能の向上が強く望まれている。

ここでは、OLTP 性能の評価を目的とした、ディスクキャッシュ制御方法のモデル化について述べる。なお、ここで評価の対象としたシステムは、当社で開発したUNIX系のワークステーションである。

評価対象システムのOLTP性能の向上を目的として、我々はまず、システムにおける処理性能のボトルネックの検出を行なった。一般に、OLTPシステムのように大量のデータに頻りにアクセスするシステムでは、入出力処理がシステム性能のボトルネックになりがちであるが、我々が評価対象としたシステムについてもベンチマークの実行時に収集したデータの解析から、その事実を確認した。

そこで、入出力処理性能の向上を目的とし、その評価手段を提供するために入出力処理性能を左右する要因の一つであるディスクキャッシュの制御方法のモデル化を行なった。OLTP環境における入出力処理をシミュレートするためのモデルへの入力データは、評価対象システム上でベンチマークを実行し、その間に発生する入出力要求を検出して作成した。このモデル、および入力データの正当性は、評価対象システム上でTP1ベンチマークを実行し、収集したデータと、モデルに入力データを与えて得た結果とを比較することにより検証した。

2 OLTP システムにおけるボトルネックの検出

当社で開発したワークステーションを評価対象システムとし、OLTP性能のボトルネックの検出するために、システム上でTP1ベンチマークを実行し、その間のCPU使用率、ディスクキャッシュのヒット率等のデータを収集し、その解析を行なった。

ここで使用したTP1ベンチマークは大規模な銀行のOLTPをシミュレートするものであり、表1で示される4種類のテーブルで構成されるデータベースを使用する。

ただし、ディスクスペースやベンチマーク実行時間を考慮して、実際には各テーブルのサイズを1/10にスケールして使用した。

ディスクへのファイルの割り当てを図1に示す。収集したデータの解析を容易するために、データファイル、ログファイルを2つのハードディスクに別々に配置した。

この環境で、トランザクションを実行するプロセス数の増減によってシステムの負荷を変化させ、データの収集を行なった。データの収集には、OSが提供するプログラムに手を加え、機能を拡張したものを使用した。

テーブル名	1レコードサイズ	レコード数	総バイト数
アカウント	100B	10,000,000	1.0GB
出納係	100B	10,000	1.0MB
支店	100B	1,000	0.1MB
履歴	50B	90 days	10.0GB

表1: TP1 で使用されるデータベースの実モデル

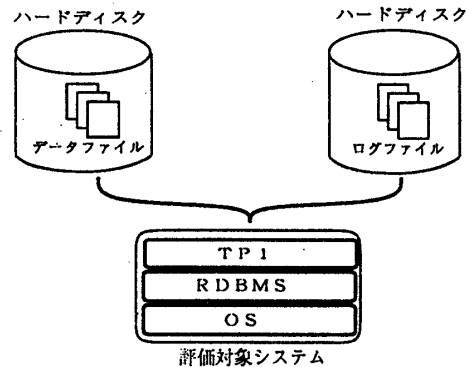


図1 ディスク構成とファイルの配置

こうして収集したデータから、

- CPUが入出力処理完了待ちになっている比率が高い。
- ハードディスクはほぼ100%動作している。

ことが明らかになり、入出力処理がボトルネックになっていることが確認できた。入出力処理を高速化するために、我々はキャッシュのヒット率を向上させることにより、ハードディスクへのアクセス回数を減少させる、という方策を検討することにした。そこで、入出力性能の評価を行なうために、ディスクキャッシュ制御部のモデル化を行なった。

Modelling of Disk Cache Control of OLTP System
Yoshihito OHTA*, Shin'ichi SAEKI*, Yasumobu MATSUMURA**
*Matsushita Electric Industrial Co., Ltd.
**Matsushita Soft-Research Inc.

3 モデルによる性能評価

従来 OLTP システムの性能は、TP1、EPI といったベンチマークを用いて評価されてきた。しかし、OLTP システムのように大規模なシステムの開発段階では、システムを構成する個々の要素単位での性能評価が必要となる。

例えば入出力処理に変更を加えた場合、OS の変更、リメイク、ベンチマークの実行、データの収集と解析、といった一連の処理を繰り返すことによってその効果を確認することも可能ではあるが、明らかに効率が悪い。ここで、変更を加えた部分の性能のみを評価できれば開発効率を向上することができる。

この際、モデル化の手法が有効になる。一般に、OLTP システムのような大規模なシステム全体をモデル化するのは困難であるが、システムの一部をモデル化することは比較的容易である。また、モデルを用いて性能評価を行なうことにより、機能の変更、拡張の効果を容易に確認することができる。

こうした理由から、我々は入出力処理の性能評価にモデルを用いた。

4 ディスクキャッシュ制御方法のモデル化

4.1 モデルの作成

ディスクキャッシュの制御方法を以下のようにモデル化した。後に機能の拡張を行なうことを考慮して、初期モデルはできるだけシンプルなものにした。

ディスクキャッシュとして使用されるバッファには、その内容が有効なもの(アクティブバッファ)とフリー状態のもの(フリーバッファ)の2種類があり、各々のバッファのサイズは可変とする。初期状態では、すべてフリーバッファであり、また、バッファを使用したい時にフリーバッファがない場合は、アクティブバッファが使用される。

4.2 モデルへの入力データの作成

評価対象システム上で TP1 ベンチマークを実行中に発生する入出力関連のシステムコール、およびその引数を、作成したツールを用いて検出し、これをモデルへの入力データとすることにより、OLTP システムにおける入出力処理をシミュレートする(図2)。

4.3 モデルの検証

モデルが評価対象システムの入出力制御方法を正しく再現し得ることを検証する。まず、モデルを評価対象システムに合わせて、以下のように実現した。

- アクティブバッファ、フリーバッファをリストで管理する。
- バッファのサイズは2K バイトとする。

- 先読み機構を採り入れる。

先読みとは、読み出し要求で指定される読み出し位置に何らかの規則が見い出せる時、次の読み出し位置を予測して、前もってその読み出しを行なう機能である。ここでは、連続する読み出し要求で指定される読み出し位置が2K バイト間隔であった場合、さらに次の2K バイトの読み出しを行なうことにした。

このように実現したモデルに先の入力データを与え、ログファイル、データファイルへの入出力要求のキャッシュヒット率を求めた。これを、評価対象システム上で TP1 ベンチマークを実行した際に収集した両ファイルへの入出力要求のキャッシュヒット率と比較した。比較はキャッシュサイズが1M、4M の2通りの場合について行なった。

この結果から、若干の相違は見られるものの、このモデルは評価対象システムを正しく再現し得ると考えられる。

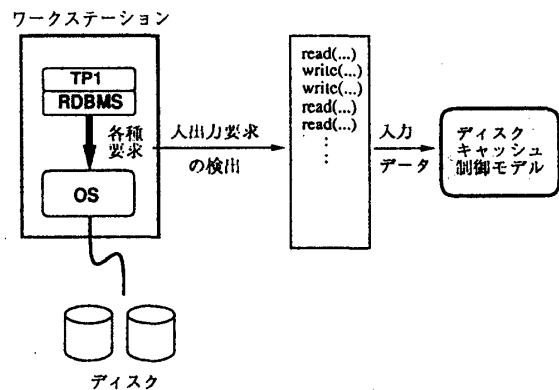


図2: ディスクキャッシュ制御モデルと入力データ

	1M		4M	
	評価システム	モデル	評価システム	モデル
ログ	96.7	96.4	96.7	96.4
データ	88.8	89.3	92.4	93.2

表2: キャッシュヒット率の比較

5 おわりに

OLTP 環境におけるディスクキャッシュの制御方法のモデル化について述べた。モデルを用いることにより、従来のように、OS のリメイク、長時間にわたるベンチマークの実行、多量のデータの収集と解析、といった多大の労力と時間を必要とすることなく、短時間に、かつ容易に性能の評価を行なうことができる。

今後は、ディスクキャッシュヒット率向上のための方策を検討し、このモデルを用いてその有効性を調査したい。