

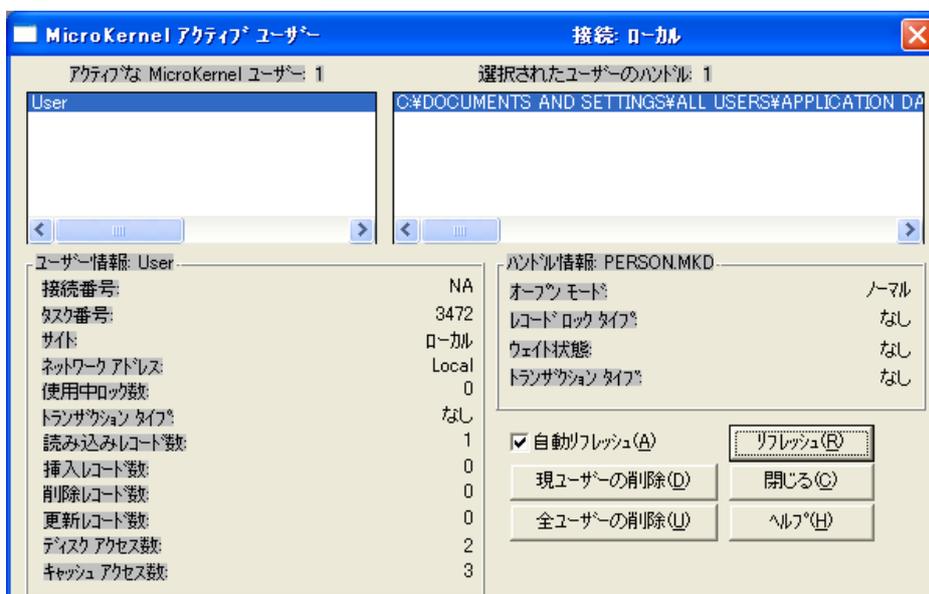
データに対してデータベース キャッシュは十分ですか？

Pervasive PSQL Summit v10 データベース エンジンには、複数のキャッシュ レベルがあります。L1 キャッシュ(キャッシュ割り当てサイズ)、L2 キャッシュ(Microkernel の最大メモリ使用量)、OS キャッシュ([システム キャッシュ]の使用)、および XtremeIO キャッシュ(使用可能なサーバー RAM が 4GB を超える 32 ビット システム用)が用意されています。お使いの環境に最適な構成を決定することは可能ですが、実際のチューニングを行うには、もっと多くの情報を手元に集める必要があります。

この Pervasive Tech Tip では、データベース エンジン キャッシュの統計情報の簡単な分析から始めます。これを基に、システム パフォーマンスを向上させるためにデータベース エンジンの構成を調整する必要があるかどうかを評価できます。

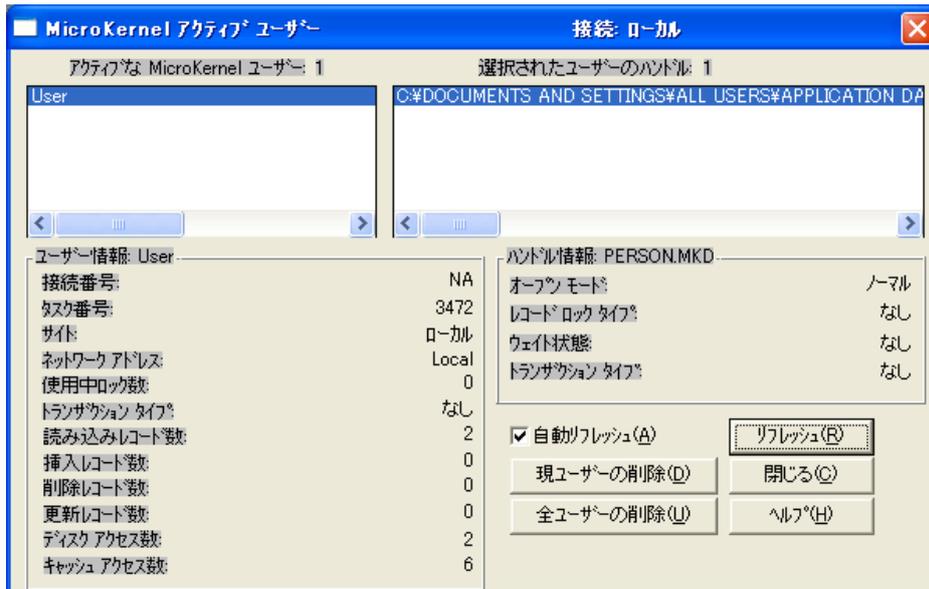
キャッシュの統計データを探す

それでは始めましょう。まず、Pervasive の Monitor ユーティリティを起動し、[Microkernel | アクティブ ユーザー]メニューを選んで画面を開きます。



左下隅の数値に特別な注意を払ってください。[ディスク アクセス数]の統計は、データベースからページが読み取られた回数の合計を示しています。[キャッシュ アクセス数]は、既にメモリ内にあるページがアクセスされた回数の合計を示しています。

この単純な例では、ファイルを開き、キー 0 を指定した GetFirst コマンドを使って単一レコードを読み取っています。これにより、ディスクから 2 ページが読み取られ、メモリ内の既存のページが 3 ページ使用され、合計で 5 ページにアクセスする結果となりました。GetFirst コマンドを 2 度目に実行した場合、どのように変化するかを見てみましょう。



ディスク アクセス数は変わりませんが、キャッシュ アクセス数は 3 増えています。これは、3 ページ(おそらく、2 インデックス ページと 1 データ ページ)がアクセスされたことを示します。インデックスを使用する GET オペレーションの代わりにインデックスを使用しない StepNext コマンドを実行した場合には、キャッシュ アクセス数のカウンターは 1 しか増えません。このようになるのは、インデックス ページが参照されず、データ ページだけがエンジンによって実際にアクセスされるからです。

ディスク アクセス数:	2
キャッシュ アクセス数:	7

ご覧のように、アプリケーションが起動し、指定ユーザーのためにデータを要求するにつれ、どのページ アクセスについてもこれらのカウンターは増え続けます。多数のユーザーが参加している実行中のシステムを調べる場合は、各ユーザーを順にクリックして結果を観察することにより、アクセス数の部分部分を見ることができます。

ディスク アクセス数:	146
キャッシュ アクセス数:	3030

ディスク アクセス数:	390
キャッシュ アクセス数:	368517

ディスク アクセス数:	505
キャッシュ アクセス数:	1322201

キャッシュの統計データを使用する

さて、この情報をどのように利用できるでしょうか？簡単なことです。「定常状態」つまり、例外的な操作が行われない状態で実行されているシステムでは、キャッシュとディスク間の比率は最低でも 100:1、できれば 1000:1 以上になる必要があります。これは、キャッシュのヒット率 99.9% を表します。上の 3 人のユーザーを調べてみると、1 人目は約 20:1 と非常にヒット率が悪いようです。2 人目は 1000:1 でなかなか良さそうですが、3 人目は約 2600:1 とさらにすばらしい結果になっています。

もちろん、個々の接続を調べることはできません。全ユーザー接続のおおよその平均をとる必要があります。さらに、"範囲外のサンプル"は無視してください。このようなサンプルは、同一レコードを頻繁に再読み取りする自動処理や、メモリ内の正常でない古いデータを絶えず探し回るレポート システムが原因で生じます。要するに、一覧はザッと目を通すだけにして、結果をじっくり見てください。

低いキャッシュ ヒット率で正常な活動をしているユーザー接続を頻繁に目にする場合、これは、もっと多くのキャッシュを提供するためにサーバーのメモリを再割り当てする必要があるという、重要な指標かもしれません。その上に、サーバーのメモリが不足している場合は、物理的な RAM を増やした方が、大幅なパフォーマンスの向上を享受するための解決法として、あまり費用がかからないかもしれません。

有用なショートカット

何百ものユーザーがいる場合、これらの統計情報を集めることは、マウス操作をする手にとって骨の折れる仕事でしょう。データの収集作業をもっと簡単にする重要なショートカットがいくつかあります。

下矢印を使用する

各ユーザーをクリックする代わりに、まず一覧内の最初のユーザーをマウスでクリックします。次に、下矢印キーを押して次のユーザーへ移り、一覧の下に着くまでずっとこれを続けます。通常、ユーザーが接続を確立した後、システムに留まるような場合には、これは大いに役立ちます。

ユーザーが一時的な接続をもっと多く作成する場合は、この方法でも問題が発生する可能性があります。1 つ目の問題は、一覧内で現在選択している接続が閉じられてしまった場合には、再び一覧の先頭まで戻されてしまうということです。これは、自動リフレッシュ レートを下げる(または完全に無効にする)ことで最小限に抑えられます。もう 1 つの問題は、下矢印キーを押したとき、次の接続が切断されていると、単純にカーソルが移動を拒否するということが起こります。マウス操作に戻って 2 つ下の接続をクリックしてから、作業を続ける必要があります。これは、自動リフレッシュ レートを下げることで最小限に抑えられます。

BMON を使用する

Pervasive Software は PSQLv10 で、Monitor/DTI データを表示するための BMON という無料の Java ツールを提供しています。BMON を使用するには Java が有効になっている必要があります。また、構成ファイル monconfig.txt を編集して、接続情報と表示したい項目の一覧を提供する必要があります。しかし、一度設定すれば、"bmon -f monconfig.txt -runonce" のような単純なコマンドを使って、ユーザー接続のブロック全体をログ ファイルへすばやくダンプできるようになります。その後、ログをスクロールして、見たい番号を探すことができます。以下に、BMON のユーザー接続の 1 例を示します。

```
*****
***START RECORD 10/03/10 19:05***
*****
ACTIVE USERS: 1
*****
      *****
      ACTIVE USER INFO: #1
      *****
      User Name:      User
      Client ID:
      Connection Number:  NA
      Task Number:    4032
      Site:           Local
      Network Address: Local
      Locks Used:     0
      Transaction State: None
      Records Read:   252
      Records Inserted: 0
      Records Deleted: 0
      Records Updated: 0
      Disk Accesses:  0
      Cache Accesses: 505
```

Distributed Tuning Interface を使用する

必ずしも既存のソリューションを使用する必要はありません。Pervasive Monitor は公開されている API 呼び出しを使用してデータ表示を行っています。本気で意欲があるならば、独自のプログラムを書いて、このデータに直接アクセスし、独自の統計情報を収集することもできます。詳細については、DTI すなわち Distributed Tuning Interface のオンラインマニュアルを参照してください。

著者情報:

Bill Bach 氏は Goldstar Software Inc. の設立者であり社長です。この会社はシカゴにある Pervasive のリセラーで、北米および海外の顧客へ Pervasive 製品、サービス、およびトレーニングを提供することを専門としています。Bill は、Pervasive データベース環境で作業するデータベース開発者やシステム管理者の役に立つ、ツールおよびユーティリティを多数作成しており、彼が開催する Pervasive PSQL および DataExchange のトレーニング講習は最も包括的な講習です。詳細については、<http://www.goldstarsoftware.com> をご覧ください。