Tech Tip: Is my Database Cache Big Enough for my Data?

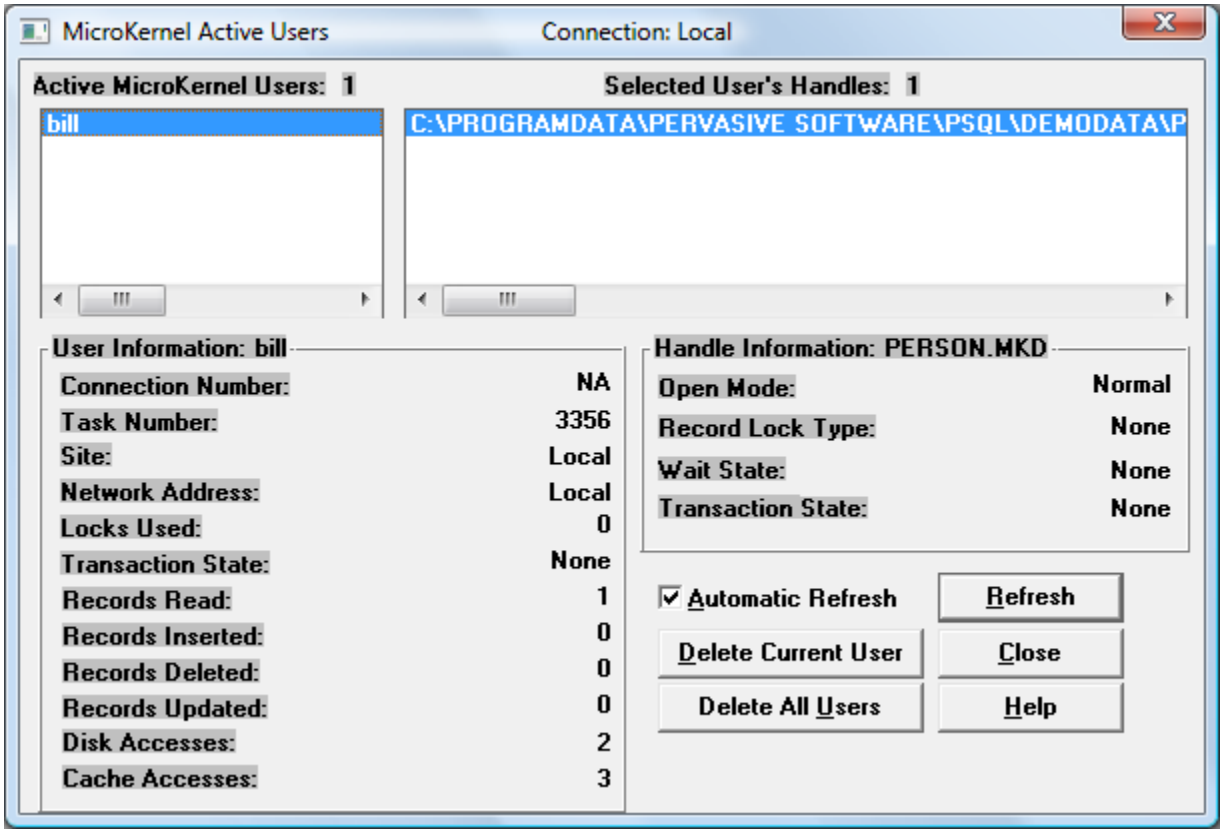
Written by Bill Bach, President of Goldstar Software Inc.

The Pervasive PSQL Summit v10 database engine has multiple levels of cache, including L1 cache (Cache Allocation Size), L2 cache (Max Microkernel Memory Usage), OS cache (Use System Cache), and XtremeIO cache (for 32-bit systems with more than 4GB of available server RAM). Determining the best configuration for your environment is possible, but requires having a LOT more information on hand before you can do any real tuning.

In this tech tip, we are going to start with a simple analysis of your database engine cache statistics so that you can evaluate whether or not your database engine configuration needs to be adjusted to improve your system performance.

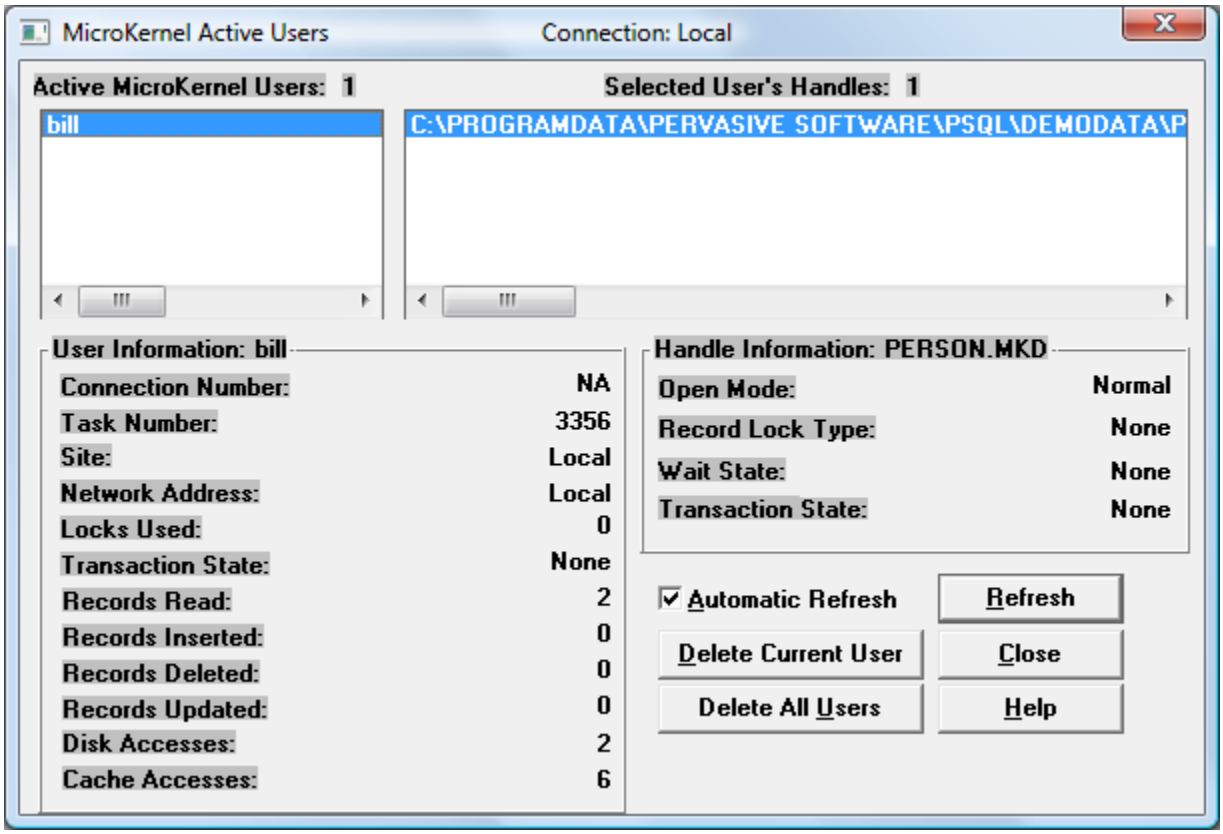
Locating Your Cache Statistics Data

Let's start by going into the Pervasive Monitor tool and looking at the Microkernel/Active Users screen.



Pay special attention to the numbers in the lower left corner. The statistic for "Disk Accesses" indicates the total number of times that a page has been read from the database. The "Cache Accesses" number indicates the total number of times that a page has been accessed that is already in memory.

In this simple example, the file has been opened, and a single record has been read with a GetFirst command on Key 0. This has resulted in 5 page accesses in all, with two coming from disk and three being used that are already in memory. Watch how it changes if we issue the GetFirst command a second time:



Of course, the disk accesses have not changed, but the cache accesses increased by three, indicating that three pages (like two index pages and one data page) have been accessed. If, instead of a GET operation (which uses an index), we issue a "StepNext" command (which does not use an index), then we see that the cache accesses counter increases by only one. This happens because the index pages are not referenced, as only the data page is actually accessed by the engine:

Disk Accesses:	2
Cache Accesses:	7

As you can see, as the application runs and requests data for a given user, these counters continue to increase on every page access. When we examine a running system with numerous users, we can then see a small slice of the accesses by clicking on each user in turn and noting the results:

Disk Accesses:	146
Cache Accesses:	3030

Disk Accesses:	390
Cache Accesses:	368517

Disk Accesses:	505
Cache Accesses:	1322201

Using the Cache Statistics Data

So, how can I use this information? Simple. In a system running in "steady state", meaning that no unusual operations are taking place, the ratio between cache and disk should be a minimum of 100:1, and preferably 1000:1 or better. This represents a cache hit ratio of 99.9%. If we examine these three users, the first seems to be a very poor ratio of about 30:1. The second is looking good at 1000:1, and the third is about 2600:1 -- even better!

Of course, you cannot look at just one individual connection -- you have to take an approximate average of all user connections. Further, you should ignore any "outlying samples" which may be caused by automated processed constantly re-reading the same records, or by reporting systems which constantly troll through old data not normally in memory. In short, just rapidly run through the list and "eyeball" the results.

If you constantly see user connections doing "normal" activities with a low cache ratio, then this may be an important indication that you should reallocate memory on the server to provide more cache. Of course, if the server is running low on memory as well, then increasing physical RAM may be a very inexpensive solution to enjoy huge performance gains.

Helpful Shortcuts

If you have hundreds of users, collecting these statistics can be a painful affair for your mousing hand. Here are a few important shortcuts to make the task of collecting data much easier.

Use the Down Arrow

Instead of clicking on each user, use the mouse to click on the FIRST user in the list. Then, tap the down arrow key to go to the next user, and continue all the way down the list. This works great if your users typically establish connections and then stay in the system.

If your users create more transient connections, then you may experience problems with this, too. One issue is that if the current connection being selected in the list gets closed, you will be taken back up to the top of the list again. This can be minimized by decreasing the auto-refresh rate (or disabling it altogether). Another issue arises when the NEXT connection has disconnected when you click the down arrow, and the cursor simply refuses to move. You have to go back to the mouse and click down TWO connections to continue. This can be minimized by increasing the auto-refresh rate.

Use BMON

Pervasive Software provides a free Java tool with PSQLv10 called BMON to display the Monitor/DTI data. Using BMON requires that you have Java enabled, and it requires that you edit the configuration file monconfig.txt to provide connection info and a list of items you want to see. However, once you have it set up, it will quickly dump the entire block of user connections to a log file with a simple command like

this: "bmon -f monconfig.txt -runonce". You can then scroll through the log and find the numbers you want to see. Here's an example of one user connection from BMON:

```
==================================
***START RECORD 1/29/10 11:56 AM***
==================================
===============
ACTIVE USERS: 1
===============
        ====================
        ACTIVE USER INFO: #1
        ====================
        User Name:           bill
        Client ID:
        Connection Number:   NA
        Task Number:         6936
        Site:                Local
        Network Address:     Local
        Locks Used:          0
        Transaction State:   None
        Records Read:        56
        Records Inserted:    0
        Records Deleted:     0
        Records Updated:     0
        Disk Accesses:       0
        Cache Accesses:      126
```

Use PSConfig

Goldstar Software provides a tool called PSConfig that can generate the same result, but in a tabular format for easier reading. PSConfig uses the command line to specify its display options instead of a configuration file. When used with the /MUT (Monitor, Users, Totals) options, you get a display like this:

```
PSConfig Version 3.32: 01/28 (C)2010 Goldstar Software Inc.

Number of Users: 3

Site Platform Hndls Read  Ins Updt  Del Cached NetAddress      UserName
Locl Unknown      1  14K    0    0    0 1460:1 Local          bill
Locl Unknown      1  401    0    0    0 Cached Local          bill
Locl Unknown      1   56    0    0    0 Cached Local          bill

Total Btrieve Reads  :        15288
Total Btrieve Inserts:            0
Total Btrieve Updates:            0
Total Btrieve Deletes:            0
Total Cache Accesses :        50655
Total Disk Accesses  :           34
Cache:Disk Ratio     :       1489:1
```

Note that PSConfig scales the counters to keep the display data succinct, and it also provides your cacche:disk ratio for each user, in addition to providing totals for all users at the end of the report. Additional data, such as the raw disk and cache counters, network address and more, can be displayed by including the "E" option on the command line. You can download a trial copy of PSConfig from http://www.goldstarsoftware.com/tools.asp

Use the Distributed Tuning Interface

Of course, you don't have to use any existing solution! The Pervasive Monitor uses published API calls to display its data. If you're really ambitious, then can write your own program to access this data directly and collect your own statistics. See the online manuals for DTI or Distributed Tuning Interface for more information.

Author Information:

Bill Bach is the Founder and President of Goldstar Software Inc., a Pervasive reseller in the Chicago area that specializes in providing Pervasive products, services, and training to its customers in North America and abroad. Bill has written numerous tools and utilities to help system administrators and database developers work with their Pervasive database environments, and his training classes for Pervasive PSQL and DataExchange are the most comprehensive classes available. Get more information from http://www.goldstarsoftware.com.