Tech Tip #5: Properly Sizing Your Database Engine Cache

Overview: Looking at the database size, cache size, and memory size and adjusting accordingly

If you followed the recommendations in the last Tech Tip, you now have a nice, fast server. However, is it configured properly? When you do a clean installation of PSQL, the engine configures itself to use 20% of total server memory for the database L1 cache, along with up to 60% of memory for an L2 cache. This is OK for most general-purpose configurations, but if you set up a dedicated database server, then you can be left wanting more performance. So, let's see how to optimize a server's memory usage.

Start with an evaluation of the current memory in the box. You can find this in the **Task Manager** or in the *Computer/Properties* screen, as shown in Figure 1.

Note the total memory indicated here is 24GB, with 12GB currently in use.

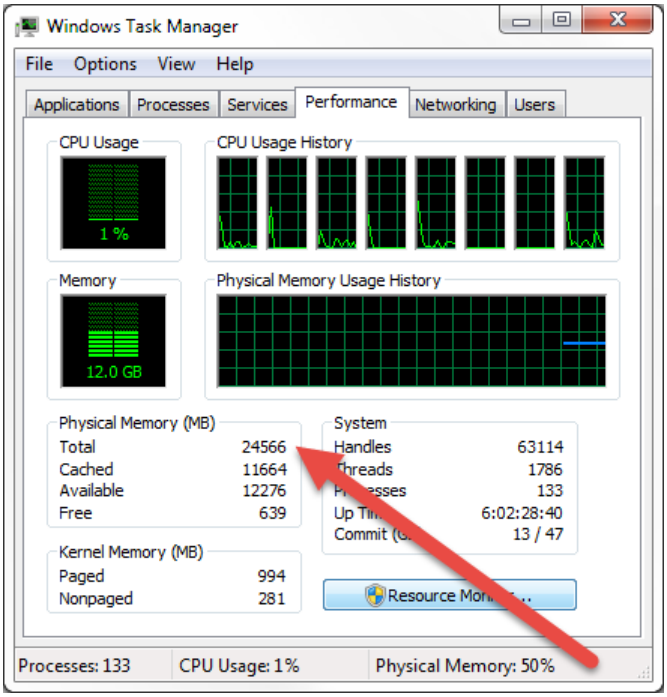


Figure 1: Task Manager Example

You can then determine the size of the L1 and L2 caches by starting the **Pervasive Control Center** and clicking on *Configure Local Engine*. From this dialog, select *Performance tuning* and check the settings for *Cache Allocation Size* and *Max Microkernel Memory Usage*, as indicated in Figure 2.

In this case, the L1 cache is set to 2500MB, and the L2 is disabled. However, the server has plenty of memory free – another 12GB. Knowing that we want to leave 4GB to 8GB free for the operating system to use, we could easily increase the L1 cache on this server by another 4GB – or almost triple the current value. As long as our database size is greater than 3GB or so, this change will likely result in faster performance for this server.

You can get another indication of cache usage in the **Performance Monitor**. First, start a PerfMon window and add the settings for *Level 1 Cache Usage* and *Level 1 Hit Ratio* from the *Pervasive PSQL Microkernel Cache* group, as shown in the example in Figure 3.

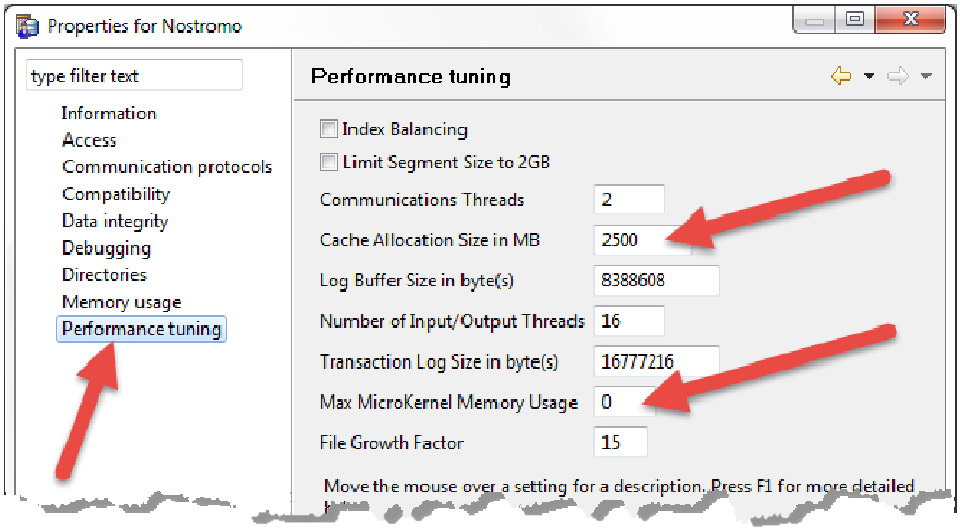


Figure 2: PCC Configuration Screen

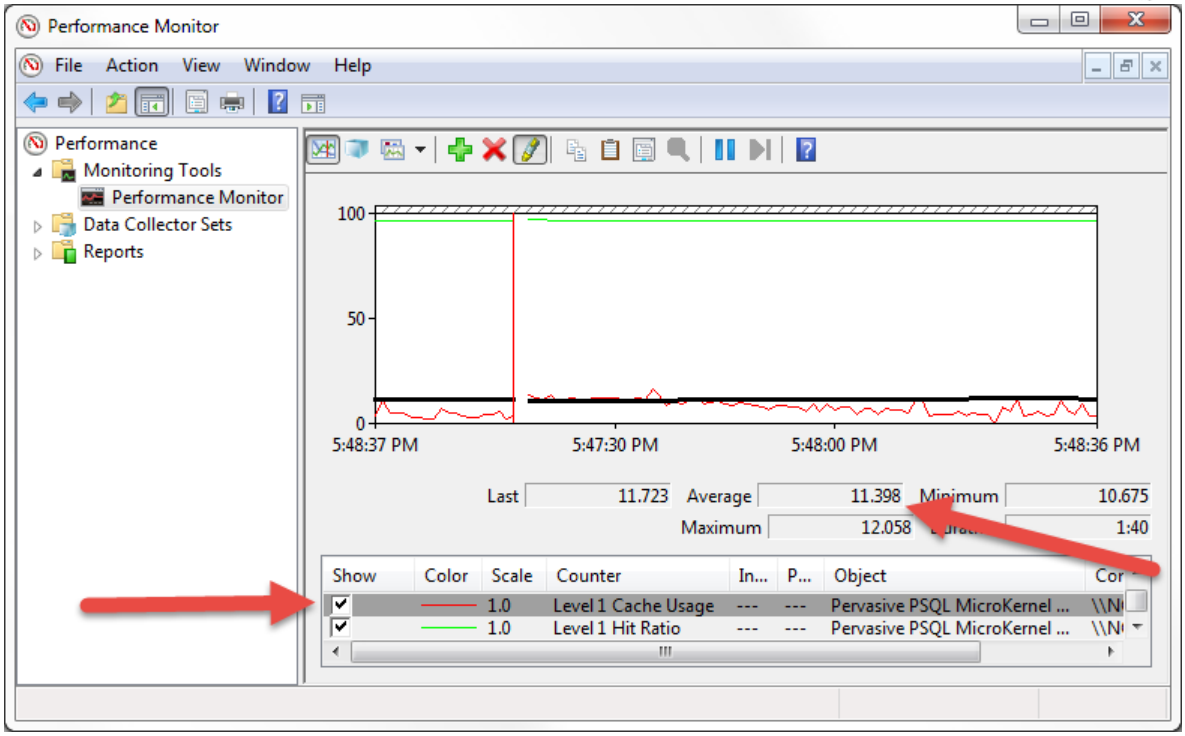
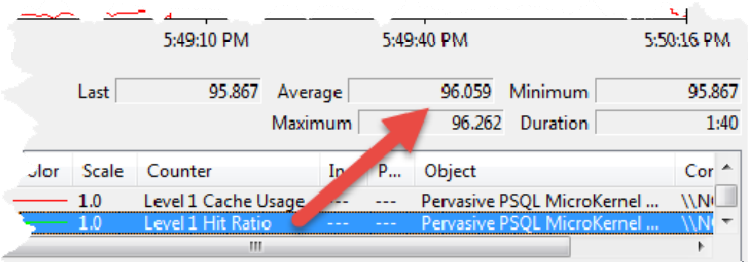


Figure3: Performance Monitor Screen

In this case, the cache usage is only 11.4% -- so we are actually wasting a large portion of the cache to begin with, and increasing the cache on this server makes no sense. If, instead, we see a usage rate of 95% or more, then increasing cache may be helpful. This is where the second value comes in:



Notice how the L1 hit ratio is 96%? Again, higher is better here. If the L1 cache is full and data is still coming in from the disk, the L1 hit ratio will start to drop – this is another indication that you need more cache memory. Try increasing the *Cache Allocation Size* and see if that helps performance.

There are, of course, many factors that feed into overall system performance. Goldstar Software has a **Server Performance Review** service that can help you review your server, look for any bottlenecks that may be slowing you down, and suggest ways to improve overall performance. You can find more information about this process here:

http://www.goldstarsoftware.com/perfopt.asp