Capturing Data with Performance Monitor

A White Paper From

GOLDSTAR SOFTWARE
www.GoldstarSoftware.com

For more information, see our web site at
http://www.goldstarsoftware.com

Capturing Data with Performance Monitor

Last Updated: 01/27/2022

One of the hardest things to track down is random performance issues occurring within an Actian PSQL/Zen database environment. The problem with tracking down a solution to a performance issue is that "performance" is actually an amalgamation of several factors, including (but not limited to) the following aspects:

- CPU Speed, Cores, and Workload
- Memory Allocated, Free, and Cached
- Disk Throughput, Latency, and Workload
- Network Throughput, Latency, Workload, and Retransmissions
- Database Engine Requests, Workload
- And more….

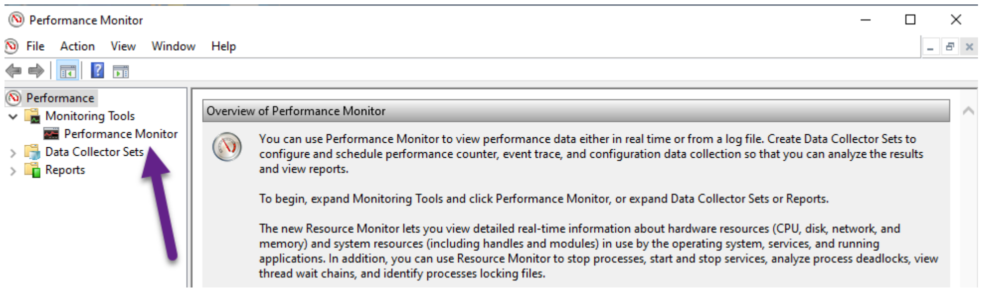
Virtualizing the hardware adds a number of additional factors at the VM level, including host CPU, host memory, host disk, SAN latency, resource oversubscription, and contention amongst various VM's on the same host.

Making the troubleshooting even more difficult is the fact that an issue in one area may mask an issue in another. For example, if a large SQL query comes in to run over a large data set and the database engine cache memory is too small, this can cause cache thrashing under the heavy load, which then requires a higher amount of disk reads to process the request. However, the additional disk activity means that the engine is not running at full speed (because it is waiting for the disk drive to return data), so CPU load is vastly reduced.

In order to pinpoint the limiting factor within the environment, you need to obtain a complete picture of all of these aspects at the same time and then understand the interplay between the various factors and how they affect each other. This is where the *Performance Monitor* comes into play.

Launching Performance Monitor

The Performance Monitor is an integral part of the Windows operating system, so it is already installed in every Windows environment by default. The quickest way to start it up is to simply run "PerfMon" from either a Command Prompt or the Windows Start Menu.



You can use Performance Monitor to view performance data either in real time or from a log file. Create Data Collector Sets to configure and schedule performance counter, event trace, and configuration data collection so that you can analyze the results and view reports.

To begin, expand Monitoring Tools and click Performance Monitor, or expand Data Collector Sets or Reports.

The new Resource Monitor lets you view detailed real-time information about hardware resources (CPU, disk, network, and memory) and system resources (including handles and modules) in use by the operating system, services, and running applications. In addition, you can use Resource Monitor to stop processes, start and stop services, analyze process deadlocks, view thread wait chains, and identify processes locking files.

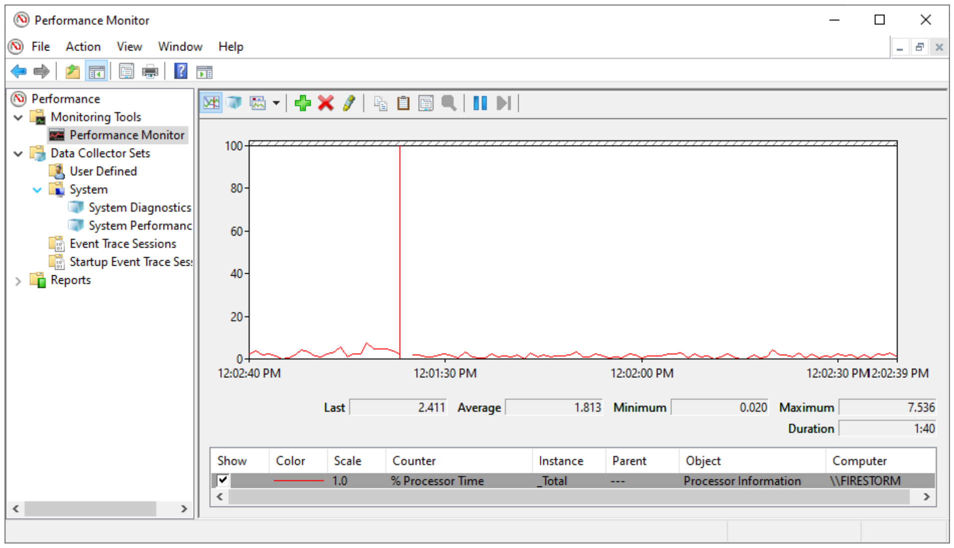
Performance Monitor Data Capture Modes

There are two ways to collect counter data from PerfMon:

1. Performance Monitor GUI: You can load up the GUI window, add one or more counters to the list, and view the data interactively right on the screen. This is best if you are looking at the system quickly to see where a problem may reside and just need data from the last minute or two.
2. Data Collector Sets: You can collect data in the background (which reduces some of the overhead) directly into a *data collector set* – or a binary file that contains the samples from the counter values over a longer time period. This takes a bit of extra work to set up and then to analyze later on, but it can capture data across a very long time period (even days), which can be useful for those "random" issues.

Capturing PerfMon Data Interactively from the GUI

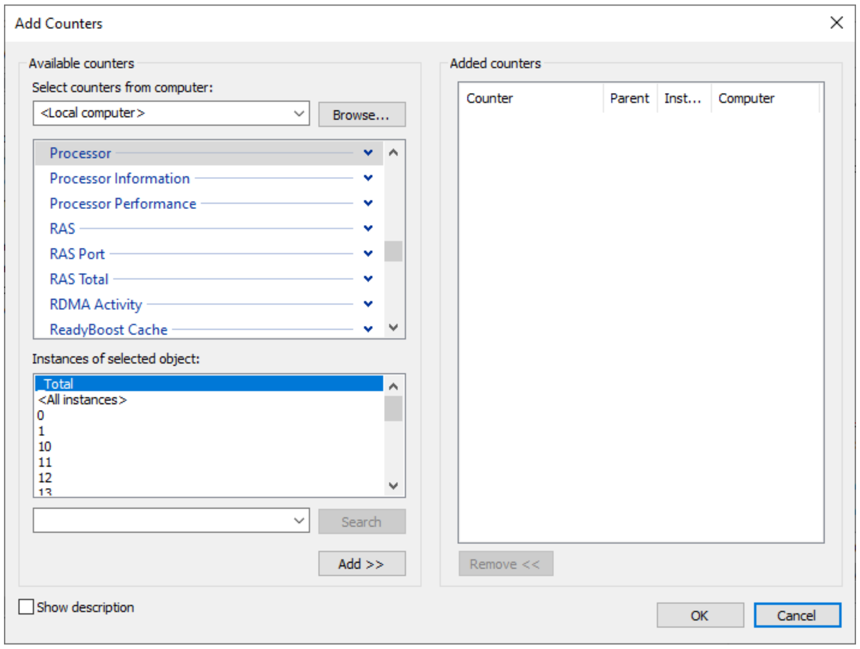
Once you start the Performance Monitor tool, you will see a starting screen with a graph and a single counter -- % Processor Time – that looks like this:



Last 2.411 Average 1.813 Minimum 0.020 Maximum 7.536 Duration 1:40

Show	Color	Scale	Counter	Instance	Parent	Object	Computer
✓	🟥	1.0	% Processor Time	_Total	---	Processor Information	\\FIRESTORM

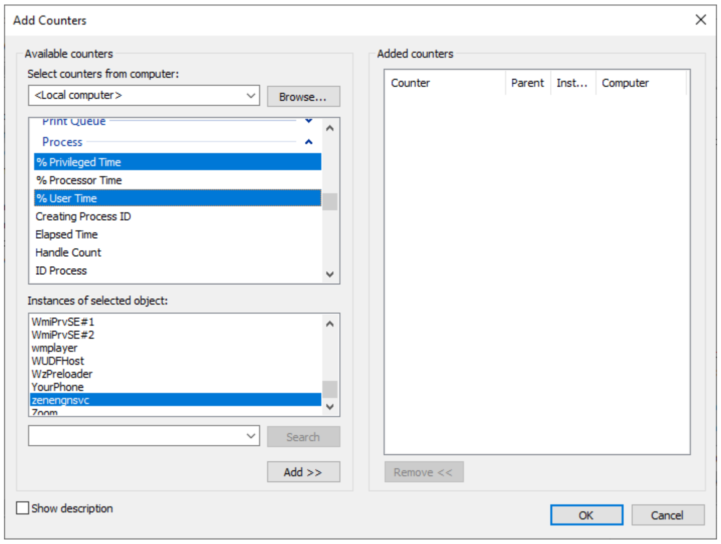
The first thing you need to do is to add new counters to this screen, which is done with the big, green "+" icon on the toolbar. Clicking this button will pop another dialog box with a list of the various counters that can be added:



There are three sections to the left side, used to select the computer, counters and instances. As you select the counters and click the Add button, they are added o the list on the right side so that you can use one trip through this dialog box to add many counters at once with ease.

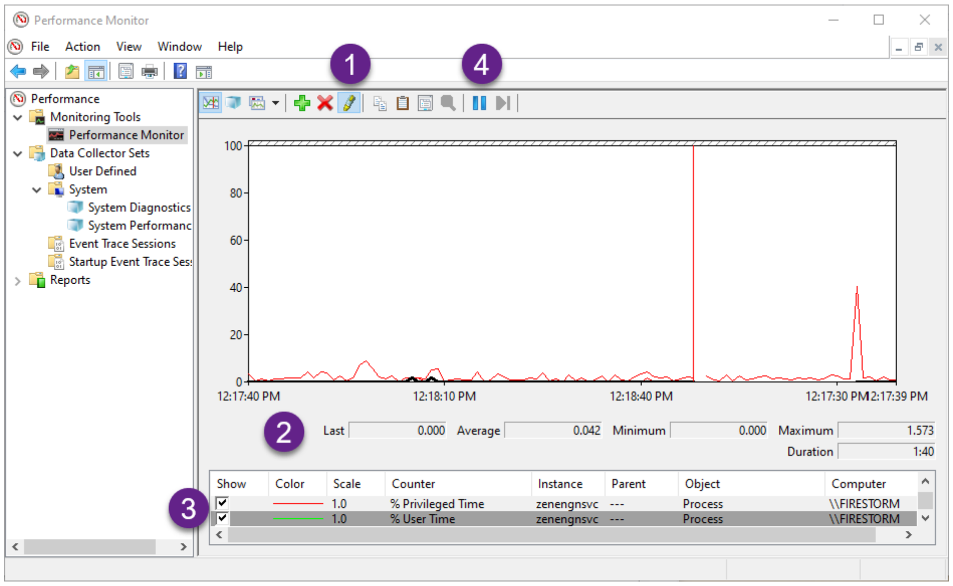
The first section on the left indicates the computer from which you want to pull the counter. Interesting, you can pull PerfMon counters remotely, which can be useful if you need to monitor multiple servers at the same time.

The second section provides a list of the available counter groups. Click on the down arrow to the right to expand the groupo into the individual counters. For example, the Process counter group can be expanded to show many different counters, such as % Privileged Time, % User Time, and more:



The third section is used when there are multiple instances available. For example, the **Process/% User Time** counter can be collected for any of the processes currently running on the server. In this case, we are only interested in monitoring the Actian Zen v14 database engine service, called “zenengnsvc”. (This was known as “NTDBSMGR” in older versions.)

Click the Add button, and the counters will be added to the right side, then click OK to close the dialog box, and you’ll be back to the graph again.



Last 0.000 Average 0.042 Minimum 0.000 Maximum 1.573
Duration 1:40

Show	Color	Scale	Counter	Instance	Parent	Object	Computer
✓	🟥	1.0	% Privileged Time	zenengnsvc	---	Process	\\FIRESTORM
✓	🟩	1.0	% User Time	zenengnsvc	---	Process	\\FIRESTORM

Here are a few tips to using the graph screen:

1. Click the pencil icon to highlight the counter that is currently selected in the lower section. This makes it possible to easily distinguish a specific counter without relying on the color alone.

2. This line contains the raw counter values. These are often more important than the graph, which can suffer from scaling issues. In this example, we can see that the **% User Time** counter peaked at 1.573% over the last 100 seconds.

3. The list of counters on the bottom provides you with a few capabilities:

    a. You can click on a counter to highlight it (with the highlighter enabled).

    b. You can press <Del> to delete a counter that you no longer need.

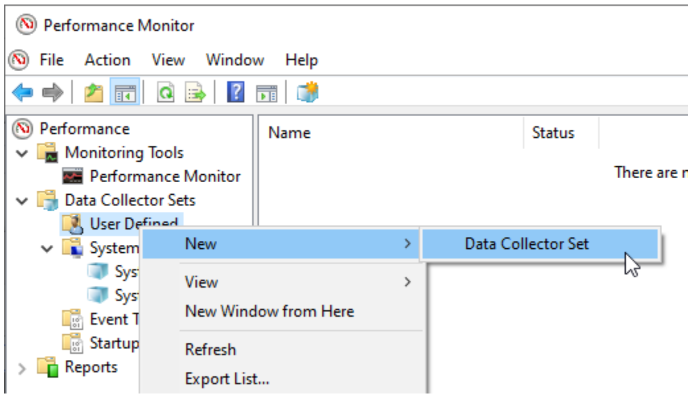
    c. You can uncheck the “Show” box to hide a counter temporarily.

    d. You can right-click on a counter (or multiple selected counters) and scale the graph automatically or adjust the scale manually.

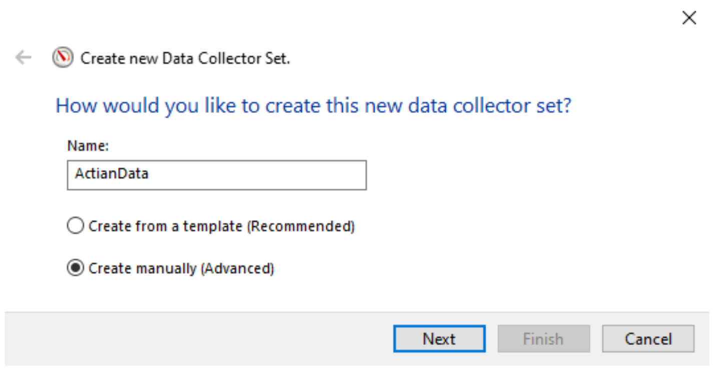
4. The Pause/Play buttons allows you to temporarily stop data collection for a short time – useful to look over an event that occurred before it gets wiped off the screen.

Creating a Data Collector Set

To capture detailed data over a longer time period, or to capture data for later analysis by additional experts, you will want to create a Data Collector Set instead. This is done by right-clicking the Data Collector Set section and select New / Data Collector Set:



You will see a series of dialog boxes (which have been shrunken to save space here):


Create new Data Collector Set.

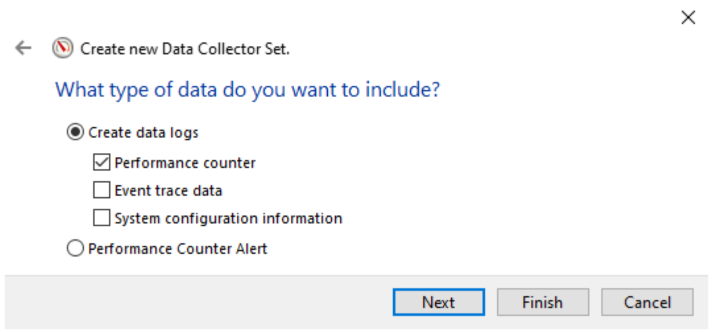
How would you like to create this new data collector set?

Name: ActianData

( ) Create from a template (Recommended)

(●) Create manually (Advanced)

The first screen simply asks for a set name (which can be anything you want), and you should use the "Create Manually" option to have full control over the counter collection. Click **Next** to go to the next screen.


Create new Data Collector Set.

What type of data do you want to include?

(●) Create data logs

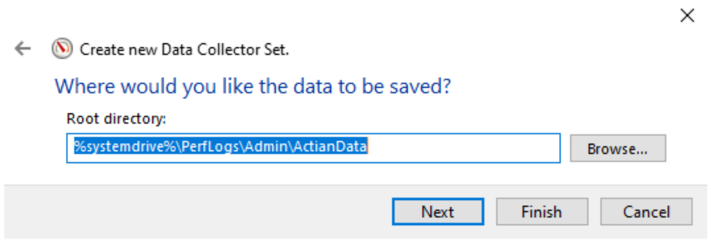
[✓] Performance counter

[ ] Event trace data

[ ] System configuration information

( ) Performance Counter Alert

On the second screen, select the Performance Counter Log option and click **Next**.

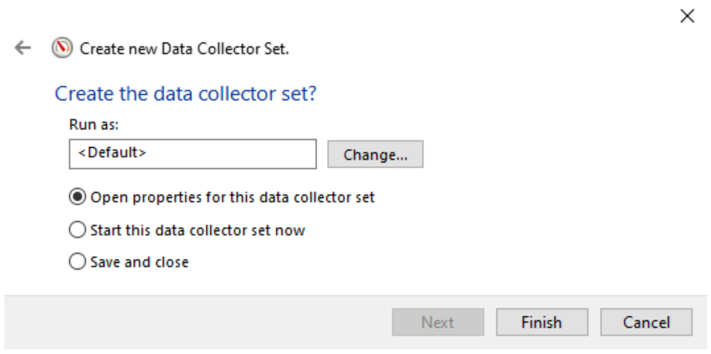

Create new Data Collector Set.

Where would you like the data to be saved?

Root directory:

%systemdrive%\PerfLogs\Admin\ActianData

Set the location for the log file (or accept the default) and click **Next**.


Create new Data Collector Set.

Create the data collector set?

Run as:

<Default>

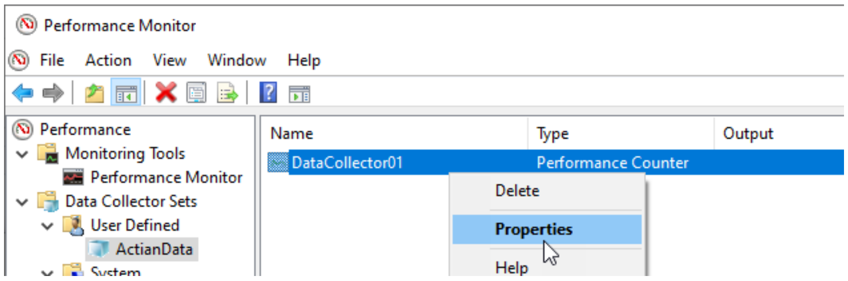
[✓ Open properties for this data collector set]

Start this data collector set now

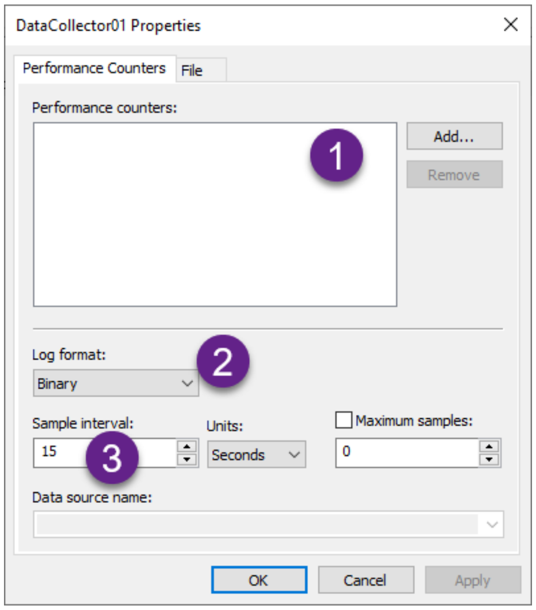
Save and close

Select the option to open the properties for the data collector set and click **Finish**. In the resulting (tabbed) dialog box, you can configuring the naming convention if you wish, you can set a schedule, or you can set a stop condition (such as afer a set number of seconds or file size). Click **OK** to close the dialog.

Next, we need to add the counters and configure the data collector. To do this, right-click the data collector set and select Properties:



This will give you a new dialog box:



Configure the data collector as needed:

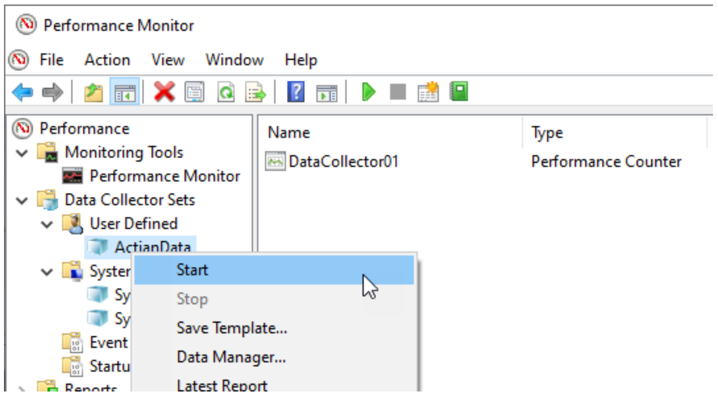
1. Use the **Add** button to add counters to this collector set. This will give you the same dialog box shown above.

2. Verify that the log format is Binary, which will be easiest to access with PerfMon. (Some people like to look at the raw data in Excel, where a Comma-Separated File may be useful.)

3. Set the sample interval here. ***Note that the default is 15 seconds, but then each counter represents the AVERAGE of the counter value over that time period. For most database issues, this is far too long of an interval. To collect useful data, you should change this from 15 to 1.***

Click OK to close this box and save changes.

To start the data collector set, right-click on it in the PerfMon dialog box and click **Start**.



You can also select the collector set click the green Play arrow on the toolbar. Use the Stop button or right-click and select **Stop** to stop it – and you now have a file that can be opened directly by PerfMon with all of your counter data.

Selecting Counters to Capture

Of course, you won't be able to capture **every** counter on a server at once, as that would add far too much workload on the system itself, making any performance problem far worse. Instead, we want to carefully select the counters to capture so that we can see a wide swath of the server at one time.

Because database issues involve so many possible factors, though, we want to capture a small cross-section of counters from every related system at the same time. The following is a list of recommended counters that we commonly use to look for database issues:

Process Group for the **ZenEngnSvc/NTDBSMGR** Instance:

- **% Privileged Time**: Percentage of one CPU core spent inside the OS code
- **% User Time**: Percentage of one CPU core spent inside the Actian code
- **IO Read Bytes/Sec**: Network and disk activity (Reading)
- **IO Write Bytes/Sec**: Network and disk activity (Writing)
- **Virtual Bytes**: Memory footprint (important only for 32-bit engines)

Logical Disk Group for **each volume** instance. Note that we want to capture the C: drive, along with any data volumes where files (production or temporary) are stored. Depending on the issues and configuration, it may be better to track the Physical Disk group instead:

- **% Disk Time**: Percentage of time the disk is busy
- **Current Disk Queue Length**: Number of disk requests pending at a given time
- **Disk Read Bytes/Sec**: Bytes read from the disk in this interval
- **Disk Write Bytes/Sec**: Bytes written to the disk in this interval
- **Split IO/Sec**: Can indicate fragmentation or unexpected sector/cluster size

Network Adapter Group for **each active network adapter** instance:

- **Bytes Received/Sec**: Bytes read from the disk in this interval
- **Bytes Sent/Sec**: Bytes written to the disk in this interval
- **Output Queue Length**: Number of disk requests pending at a given time
- **Pacxkets Received/Sec**: Bytes read from the disk in this interval
- **Packets Sent/Sec**: Bytes written to the disk in this interval

Memory Group (only if memory issues are suspected):

- **Available MBytes**: Available memory in MB
- **Cache Bytes**: The size of the active OS File System Cache

- **Page Faults/Sec**: Indication of memory pressure to/from the page file
- **Pool Nonpaged Bytes:** Optional: Only if resource issues present
- **Pool Paged Bytes:** Optional: Only if resource issues present

Actian Zen Microkernel Btrieve Operations Group:

- **Btrieve Open Operations/Sec**: Number of FileOpen operations completed
- **Change Operations/Sec**: Database inserts, updates, deletes
- **Operations/Sec**: All Microkernel-level operations

Actian Zen Microkernel Cache Group (select only those for the cache that is being utilized):

- **L1 Cache Discards/Sec**: Pages purged from L1 cache
- **Level 1 Cache Dirty Percentage**: Dirty pages (needed only for small caches)
- **Level 1 Cache Hit Ratio**: Page hits/accesses for L1 cache
- **Level 1 Cache Hits/Sec**: Page accesses inside L1 cache
- **Level 1 Cache Misses/Sec**: Page requests not satisfied by the L1 cache
- **Level 1 Cache Usage Percent**: Percentage of the L1 cache currently in use
- **Level 2 Cache Hit Ratio**: Page hits/accesses for L2 cache
- **Level 2 Cache Hits/Sec**: Page accesses inside L2 cache
- **Level 2 Cache Misses/Sec**: Page requests not satisfied by the L2 cache
- **Level 2 Cache Usage Percent**: Percentage of the L2 cache currently in use

Actian Zen Microkernel I/O Group:

- **Pages Read/Sec**: Number of page reads completed
- **Pages Written/Sec**: Number of page writes completed

Actian Zen Microkernel Transactions Group:

- **System Transactions in Progress**: Count of active system transactions
- **Transaction Commits/Sec**: Number of database transactions committed

The above list usually gives a holistic overview of the entire database environment. Of course, additional counters may be beneficial in certain conditions (such as the Page Server counters when client-side caching is in use)

Analyzing the Collected Data

Trying to interpret the data is the hard part in all of this. As indicated at the start of this document, the interplay of each counter and what that counter signifies can differ depending on what else is going on in the environment. For example, a period of time where there is VERY heavy Microkernel activity could indicate a single SQL query rapidly pulling data from cache, or it could indicate a period of time with a bunch of users all making requests simultaneously. A period of time with low Microkernel activity could indicate that users were on a break, that the engine was waiting for disk I/O operations to complete, that the engine was completely suspected by the OS to flush memory to the page file, or that users were blocked waiting for a user to finisah a long-running transaction. It is only by combining all of this data, along with an understanding of the environment and how each component interacts, that you can start drawing conclusions. Of course, you then need to identify a way to test the theories and confirm or disprove each one. Once you've found the root cause of the bottleneck, you can then figure out what needs to be done to resolve it.

If you still have other questions, contact the technical wizards at Goldstar Software (www.goldstarsoftware.com/contact.asp) and let our professionals work with you – the optimizations you are able to make may just save you some time!