Checking Your Pervasive Server CPU Performance

A White Paper From

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

Checking Your Database Server CPU Performance

Last Updated: 05/22/2025

The Actian Zen (formerly Pervasive PSQL) database engine offers great performance over a wide variety of environments. The newer versions have been carefully tuned over the years to install and run much better out of the box than older versions. In most cases, you simply install the engine and you can enjoy screaming performance.

Recently, however, we have been seeing a scary trend in the database community. Sites are installing brand new servers, using the latest hardware, the latest CPU's, and the latest operating systems, and yet performance seems to be lagging behind older, often much less capable, systems. What's up with that?

What Is the Problem?

The problem in a nutshell is the new components themselves!

Many years back, the race to getting faster servers meant getting servers with faster CPU's and a higher clock speed. It was easy to tell that a 2.0GHz CPU would outperform a 1.0GHz CPU – and provide around 2x the performance.

In more recent years, though, the push has been towards more cores, running at **lower** clock speeds. (The lower clock speeds help regulate the power requirements, and thus the heat dissipation required.) The use of multi-core CPU's is great when a server is heavily loaded, but if you are trying to optimize a single-threaded environment (i.e. one long-running process), then you won't see much gain, and in fact, you may get even poorer performance than an older box.

As CPU designers (i.e. Intel and AMD) have now started pushing the envelope in terms of the number of cores that they can fit in each package, they are now turning their efforts towards the power profile of the chips – and using technology to lower the power requirements of the hardware to "go green". Again, this is a good thing, as it cuts the use of electrical power, and thus lowers the heat generated, enabling a further savings in cooling the data center as well.

So, what could possibly be wrong with all of these advances? Simple – many people test "performance" by running a single process (like a long-running report) and timing how long it takes. All of these "advances" in hardware can actually make this process take **longer** on newer, more advanced hardware than it ran on older boxes!

How Can I Test My Server?

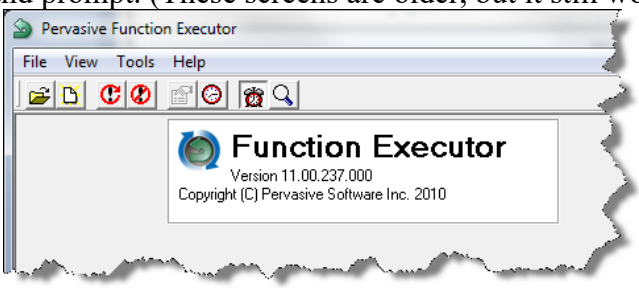
In a large, multi-user environment, the overall server performance is a complicated mix of CPU horsepower, memory usage, disk performance, network speed, and more. However, we've recently found that some problems (especially linear processes) can be boiled down to a single item – raw CPU power.

The concept behind this test is simple. We want to open up a Btrieve file from the server itself (in order to eliminate all networking issues), read ONE record from it (which will require just a few quick disk read operations), and then repeat the *exact same operation* over and over again (which eliminates all subsequent disk read operations and therefore the disk subsystem from the testing). This process forces the database accesses to be handled from memory only, eliminating all other impacts on the server, including the network and disk I/O.

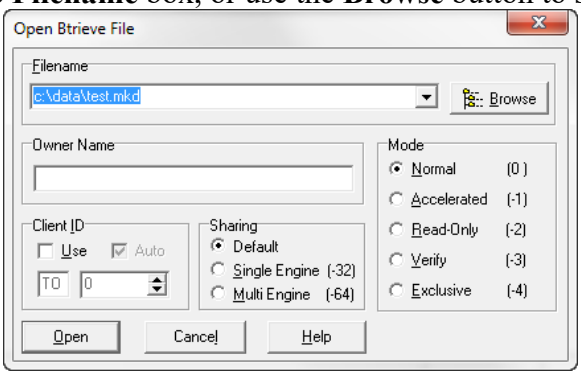
There are two ways to do this testing, using the built-in Function Executor utility and using the free BtrStress tool from Goldstar Software.

Testing with Function Executor

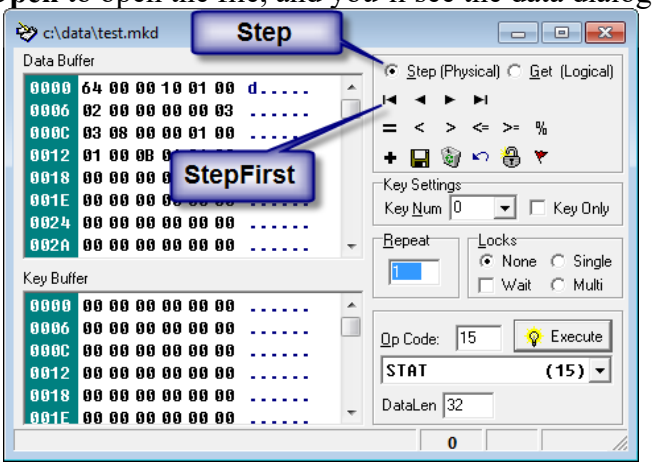
To run this test, first start up the **Function Executor** utility. You can do this from the ZenCC/PCC Tools menu, from the Start Menu, or by simply running **WBEXEC32** from any command prompt. (These screens are older, but it still works the same.)



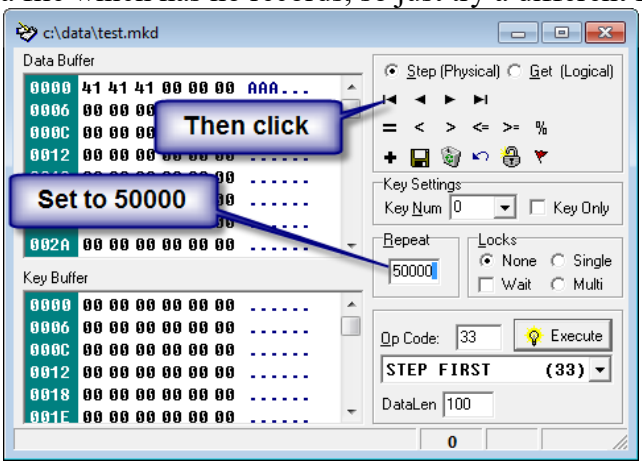
Next, select **File/Open** and get to the **Open Btrieve File** dialog box. Enter any data file name into the **Filename** box, or use the **Browse** button to select any Btrieve file.



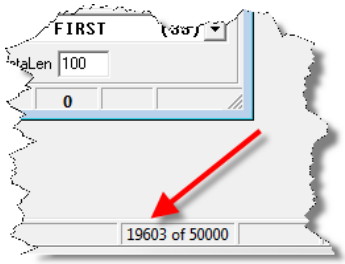
Click **Open** to open the file, and you'll see the data dialog box.



First, make sure that the **Step (Physical)** radio button is selected, then click on the **StepFirst** button. This reads data into the Data Buffer, and eliminates the need to do any further disk I/O activity for the duration of this test. (If you get a Btrieve Status 9 from this, then you picked a file which has no records, so just try a different file.)



Then, change the **Repeat** value from 1 to 50000, and click on the **StepFirst** button again. The tool will now read the same record, 50000 times. While it is working, you will see a counter at the bottom of the primary window that will increase until it hits 50,000, at which time the process is done.



Now, do it again, but this time get out your stopwatch and time the process. Record your results.

Interpreting Your Function Executor Test Results

In our tests, we have seen a wide variety of performance results from this crude test, even working with a simple stopwatch by hand. Every file may be different, and some files may have more complicated "first" records than others, which will be especially true for compressed files, or those with variable length components. However, the relative performance between two servers in your own environment, especially when tested with the same file, can be very telling, and this is why we recommend using the PERSON.MKD file from the DEMODATA database for testing.

At one site, we observed a variety of timings for 50000 operations ranging from 2.97s to 13.6s on various servers. At another site complaining of very slow performance, we saw timings as high as 18 seconds! At a third site (OK, our offices), we saw a machine clocking in at 1.9 seconds. Where does your server fit into this spectrum?

Testing with BtrStress

As this component is not provided by Actian, you must first download the **BtrStress** testing utility from this link:

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

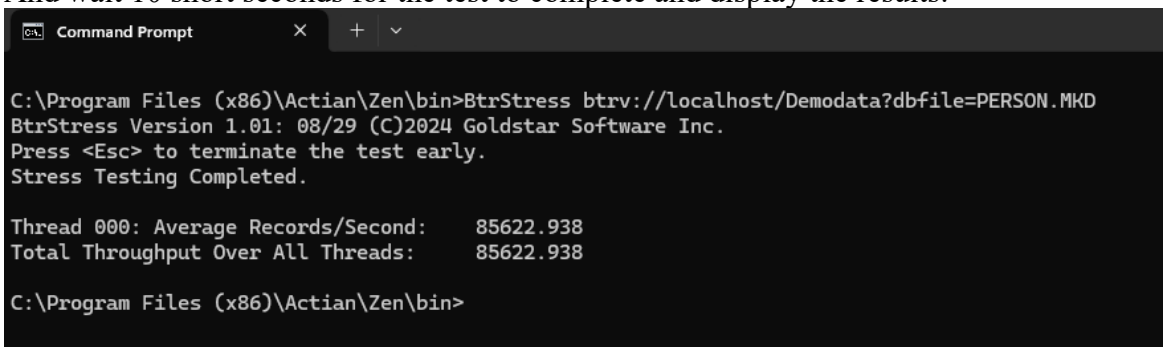
Open the ZIP file and extract the BtrStress.EXE utility into some folder on the server where you can find it easily. For best results, I recommend extracting it right into the database engine folder, which will be *C:\Program Files(x86)\Actian\Zen\bin* for current versions of Actian Zen. With it in that location (which is in the search path), it will be available from anywhere in a Command Prompt window.

Next, open up a Command Prompt and navigate to the folder where you placed the file (if it is not in your path).

Use this command to start the test:

BtrStress btrv://localhost/Demodata?dbfile=PERSON.MKD

And wait 10 short seconds for the test to complete and display the results:



If you have a multi-core server, then you can also check to see how well your system scales by adding the /T option along with a number of cores to check. Here's the same server with a /T4 option:

```
C:\Program Files (x86)\Actian\Zen\bin>BtrStress btrv://localhost/Demodata?dbfile=PERSON.MKD /T4
BtrStress Version 1.01: 08/29 (C)2024 Goldstar Software Inc.
Press <Esc> to terminate the test early.
Stress Testing Completed.

Thread 000: Average Records/Second:    60551.555
Thread 001: Average Records/Second:    59052.754
Thread 002: Average Records/Second:    60081.973
Thread 003: Average Records/Second:    60487.902
Total Throughput Over All Threads:    240174.188
```

Interpreting Your BtrStress Test Results

Again, we like to use the DEMODATA database file PERSON.MKD for the testing, which allows for comparable testing between like environments.

For a single thread, performance numbers in the 40-42,000 Records/Second range are typical for the 2.0 to 2.2GHz multi-core servers commonly used for virtualization today. A server clocking in at 3.5GHz should return metrics closer to 55-65,000. (The server I ran the test above on is 3.7GHz with turbo mode up to 4.2GHz.) If you see performance metrics that are far higher than these, then consider yourself a happy person! If you see numbers much lower than these, though, then you will likely experience performance problems in your database environment. I have even seen servers turning in numbers as low as 18000! If you need a brutal comparison, I have a Raspberry Pi that can get 26000 records per second….

When testing the multi-threaded values, things are a bit tougher. You can try various numbers of threads to find the "sweet spot" for your particular server. As noted in the above test with 4 concurrent threads, we see that each thread runs slightly more slowly (60K vs 85K) due to CPU and memory contention, but the total throughput when running all this activity at the same time is substantially higher (240K), so this CPU scales fairly well to at least 4 cores. You can re-run the test by changing the /T option to use more and more cores until the total throughput numbers start to decrease, and this will tell you how many concurrent database operations will effectively run on your server without slowing things down too much.

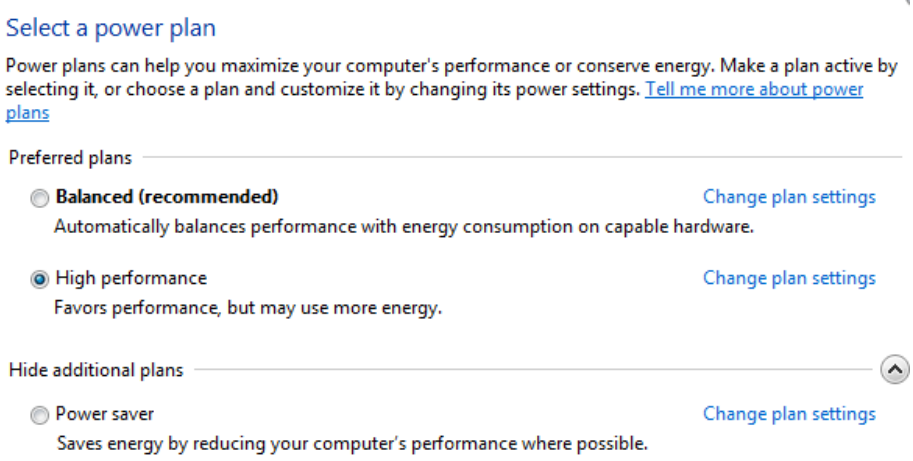
If you wish to compare NETWORK performance with these tests, then just run the same test on a workstation with the Zen/PSQL Client installed, and change "localhost" to the server name you wish to test to. You should see 8-10x *slower* performance over a typical network link due to the network latency overhead. If you see much lower throughput than this, then this could indicate a networking issue of some kind.

Again, this test should be used as a metric only, and NOT be directly compared to real-world scenarios. In a live system, you would never access the same database record over and over again like this, but rather you will be reading and writing data from many files and many users at the same time. However, the metrics do provide a meaningful comparison between similar systems, and a way to find out if a server is performing as it should or not.

Adjust your CPU Performance Within the OS

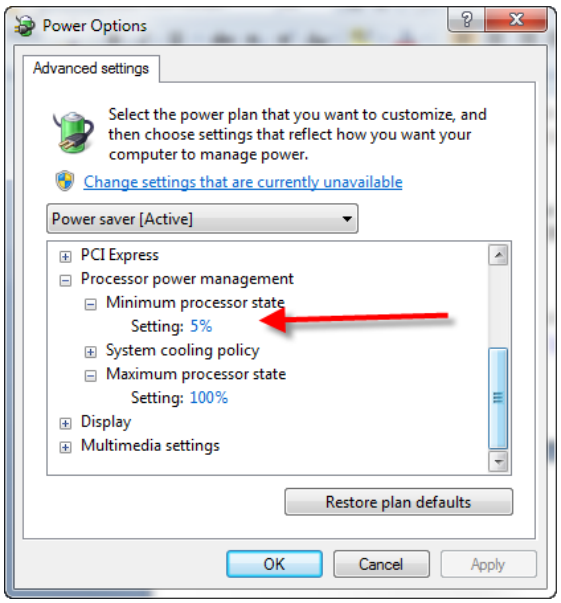
Remember all of the new features provided by AMD and Intel in the area of power management? The newer CPU's can not only put cores to sleep, but they can lower the clock speed when a system is not busy to cut power usage. Strangely, even though the manufacturers indicate that the system can instantly return to full speed, we are finding that this is NOT the case.

The first thing to try is to see if your power can be controlled by the operating system. Windows has a **Power Options** control panel, and this is a great place to start.



For example, on one test machine, changing the settings from **Power Saver** to **High Performance** provided almost a 2x performance gain!

You can also drill down further into the **Change plan settings** dialog, then **select Change Advanced Power Settings**, and fine-tune these settings even more.



Try playing with a variety of the settings in this dialog, but start with the **Minimum Processor State**.

Remember, though, that as you increase performance, you'll likely increase power consumption, increase heat generation, and increase fan speed (and thus system noise) on systems with variable-speed fans.

What If I Cannot Change the OS Settings?

If you do not have these options, which will likely be the case if you have an older CPU, then you may not be able to leverage the OS tuning capabilities for the system performance. Instead, you may want to reboot the server, dig into the BIOS, and look around for related settings in there.

On an HP Server, we found a setting in the BIOS called **Power Profile** that had to be set to Maximum Performance to get the best possible speed. More detailed information can be found on the HP white paper here:
http://h20000.www2.hp.com/bc/docs/support/SupportManual/c00502616/c00502616.pdf

On an Intel white-box server, we could only find one option related to performance, called **EIST**. This is short for Enhanced Intel SpeedStep Technology. Disabling this option provided a suitable jump in performance.

On Dell servers, there is a BIOS screen called **System Profile Settings**, and a value in there called **System Profile**, where you can disable the Dell Active Power Controller (DAPC) by changing the setting to **Performance Per Watt (OS)**.

If in doubt, check with your motherboard manufacturer for more information before changing or disabling things at random.

Can I Get the Best of Both Worlds?

Do you often need your system to run as fast as possible, but you also care about the environment? Windows users can use a Sidebar gadget that offers a quick and easy way to manage your power profile on the fly – just click on the icon and change your profile. You can also run a PowerShell script that changes the power profile. Of course, Microsoft keeps changing how this is done with each new release, so instead of including solutions here, I will simply recommend that you do a web search for "change power profile with PowerShell" and leverage the results.

Is This Only an Issue with Pervasive?

Certainly not! Here's some other web sites which comment on the same issue with MS SQL:
http://sqlserverperformance.wordpress.com/2010/02/22/power-plans-and-windows-server-2008-r2/
http://www.brentozar.com/archive/2010/10/sql-server-on-powersaving-cpus-not-so-fast/

If you still have questions, contact Goldstar Software and let us work with you to help!