



Firebird White Paper

Firebird performance by numbers ...

Holger Klemt, March 2015

Many customers have come to us with their Firebird performance issues and in most cases we have been able to improve the general performance of the database operations inside Firebird, sometimes in rather strange ways, for example:

- Deleting an index can sometimes speed up your database operations
- Removing duplicate indices, which can often be found
- Avoiding bad data types, for example very long varchar/char columns
- Using global temporary tables where they make sense
- Superserver is often faster and works fine, when the priority scheduler is off
- ... and much more.

We check and change such things when we are performing on-site or remote performance support around the world.

If you could use our help, contact us for a quote, or if you want to learn the basics yourself, visit an IBExpert and Firebird Power Workshop: the next one is scheduled on the Mediterranean island of Malta at the end of May 2015; a must for professional developers working with the Firebird database system.

However there is one thing that we cannot change just using a keyboard: the hardware!

To give you an impression of the suitability of your specific hardware choice as a Firebird server, we use the IBExpert Benchmark tool, available in the IBExpert Developer Studio full version, the IBExpert Day License and the IBExpert Company Year license.

The benchmark uses Firebird 2.1, 2.5 or 3.0 and creates a demo database, filling a number of demo tables with data and finishes by executing a multi-threaded workload with 10 parallel threads, to see how the system works in a multi-user environment. The results are displayed in seconds and two indices are created based on the result of a databases server that we used to sell 4 years ago. This is referenced to be 100%.

The test is performed twice. In the first run, it uses a minimal cache buffer setting of 50 to create as much workload as possible on the I/O subsystem, which is your hard disk, RAID, SSD, SAN or whatever you use to put your database on. The resulting index is called the drive index, as it indicates the general quality of the I/O subsystem on your machine for a Firebird server.

In the second run it uses a cache buffer setting of 5,000 pages. This executes considerably less operations on the I/O subsystem. This index is called the CPU index, since it represents more or less the speed of the mainboard, memory and CPU. However if the I/O subsystem is slow, this index will also not attain high values, because with 5,000 pages in the cache, Firebird needs a lot of I/O operations.

Based on the number of CPU cores and the Firebird server implementation (Superserver, Classic and SuperClassic) you will see a big difference in the results. You should also be aware that running the complete benchmark results in about 14 GB of read and write operations on your I/O subsystem when you are using the Superserver and, based on the Classic or the SuperClassic server it performs around 43 GB of read and write operations. Perhaps you can imagine why the results for these two implementations are not my favorite!

Before we discuss any numbers, I'd like to invite you to perform the benchmark on the machine that you are using for your Firebird server to understand the differences. A database server requires a completely different kind of hardware compared to other systems. This is one of the reasons why we launched our IFS Server systems, which are completely optimized for use with Firebird at the best price, power and rate of reliability you can get.

If you do not have a full IBExpert version, please contact register@ibexpert.com with the subject "Benchmark test" and we will give you a free IBExpert Day License that can be used for the test.

Sample tests

Hardware/OS		Firebird Version:			
		2.5 Superserver	2.5 SuperClassic	2.5 Classic	3.0 beta 1 Superserver
Pentium 3220 Win81	Drive Index %	140%	92%	89%	134%
	CPU Index %	120%	28%	27%	147%
	initall	12,3	14,1	14,2	14,1
	tmp	1,3	1,4	1,4	1,5
	threads	13,8	25,9	27,2	13,0
	initall	3,4	3,4	3,4	3,5
	threads	3,6	26,1	27,4	2,2
Xeon e3 1225 Win81	Drive Index %	159%	99%	93%	139%
	CPU Index %	152%	29%	30%	198%
	initall	10,9	13,1	13,3	13,9
	tmp	1,0	1,2	1,1	1,1
	threads	12,2	24,3	24,5	12,7
	initall	2,8	2,9	2,9	3,0
	threads	3,0	25,2	25,7	1,3
Xeon e3 1225 Ubuntu 14	Drive Index %	175%	48%		
	CPU Index %	88%	14%		
	initall	11,9	22,3		
	tmp	1,1	1,2		
	threads	9,0	55,8		
	initall	2,9	3,1		
	threads	6,7	56,0		



What do these values tell us?

All tests were performed on completely identical hardware; in the first test a Pentium Dual Core 3GHz newest series was used. In the second and third test an Intel Xeon e3 Quad Core CPU with 3.2 Ghz was used in the same hardware. This includes 8 GB ECC RAM and a 200 GB Enterprise SSD, connected directly to the onboard SATA3 controller. With Linux, only the Xeon CPU and Firebird 2.5 has been tested. The Xeon version corresponds to our IFS Basic Server.

The following can be noted:

1. The much less powerful dual-core CPU is only a little slower with the 2.5 Superserver, and using the 2.5 Classic/SuperClassic test even equal.
2. The results for the Superserver are all much better.
3. The results for 2.5 Classic/SuperClassic hardly differ.
4. The results of the Drive Index for Firebird 3.0 Superserver are significantly better than any Firebird 2.5 versions.
5. The particularly important *Threads* value in the CPU Index benefited significantly from the Firebird 3.0 architecture and additional CPU cores.
6. Linux is the front runner with the best Firebird 2.5 Superserver Drive Index, however the CPU Index is inferior to Windows.

Discussions and further explanations

Can be found on our social media pages:

- <https://www.facebook.com/IBExpertise>
- <https://twitter.com/IBExpertise>

And take a look at our *Firebird for Professionals: IBExpert Benchmark* tutorial video:

- <https://www.youtube.com/user/IBExpertise>

Interested in our IFS Servers?

Download further information and our order form:

- http://ibexpert.net/ibe/uploads/Main/IBExpertKG_IFSServer_en_1501.pdf