Firebird White Paper

Firebird Performance or: How can I improve it?

Holger Klemt, April 2016

Firebird's transactional capabilities enable it to collectively confirm or undo related operations.

The designated commands for this, namely Commit and Rollback, are however based on an infrastructure within the database which every programmer should be aware of. Because a transaction can remain active for several days, the Firebird server creates a significant overhead inside the database file, if the developer does not rigorously pay attention to short-lived transactions. Many programmers working with Delphi dataset-based architectures unconsciously disregard these requirements, because it has negligible measurable impact in single-user operation - the mode most developers use when programming. The real issues are only noticed when the application is deployed at the end customer, who has considerably more work stations in continuous operation. When the programmer works on a copy of this database in his own development environment, no performance problem can be detected, because the Firebird server has a highly efficient garbage collection system, which is activated as soon as no old active transactions are blocking it. When performing a Backup/Restore as well as when making a simple copy of the database file (which for various reasons you should never perform during live operation), the garbage collector removes the database content responsible for the poor performance.

This results in the typical situation that many software vendors are confronted with: the customer is dissatisfied with the application performance. The software company can see these problems on the customer's system, but not detect any problems in their own development environment. While the customer insists that his database server is extremely expensive, extremely fast and the best server currently available on the market, and that the virtualization, which may perhaps be running on it, is certainly not responsible for the performance problems, the software company insists that it cannot be due to its own software, as a copy of the customer database can be demonstrated with significantly better performance, even on a laptop.

Who is right?

In short, based on our consultancy work: In the majority of cases neither of the two...

We usually begin by analyzing the symptoms on the customer database during normal operation. To determine an initial comparison value for the server speed, we use the IBExpert Firebird Benchmark. This appraisal value is referenced by 100% to a system, which was sold by ourselves as the first IFS database server back in 2010.

If the server attains 100%, then this is as fast as the 1500 € Server sold by us at the time.

The benchmark value can only be determined using IBExpert IDE full versions, the Personal Edition does not have this function.

The IBExpert Day Edition is particularly suitable for this test at a customer's. The benchmark result consists of two values. The Drive Index tests the I/O subsystem with an extremely small cache. RAID controllers and rotating hard drives are very often significantly slower than expected, similar to external storage systems, due to the physical fundamentals (block size, latency, etc.). It is very important that the benchmark is executed with a Firebird database, and not with a synthetic I/O benchmark, whose behavior rarely corresponds to that of a Firebird database.

The second value is the so-called CPU Index, which operates with a larger cache, so therefore substantially less I/O operations are performed. Here CPUs with a large number of cores often attain significantly poorer values, because the tact is lower and there is usually no turbo mode.

Our current IOS 6.0 Server series, which is available starting at 2500 € per server, achieves a Drive Index of at least 200% and a CPU Index of at least 150% benchmarking the Firebird 2.5 Superserver on a Windows operating system.

If your current server attains the frequently determined value of 20% Drive Index and 15% CPU Index, then all Firebird operations on this server are taking ten times longer than on our IFS Server. Experience has shown that these values are often simply considered unrealistic by the IT department responsible for hardware or the system integrator supplying the hardware, and they attempt to prove the hardware performance by simply copying a very large file within just a few seconds from one path to another. This may be suitable for a file server, but a Firebird server has quite different requirements. Instead of one very large file, it is better to copy 100,000 very small files parallel in 10 separate windows. This rapidly separates the wheat from the chaff, and the server being tested quickly turns out to be an expensive but poor investment! For a significantly lower budget you can purchase a much more economically-priced model, such as our IFS servers, from a specialist Firebird provider, with significantly enhanced Firebird performance.

Often the budget would suffice to buy two identical servers. Using hourly shadow backups and parallel shadow operation on a hot plug drive, these servers offer virtually 100% reliability, and for an additional fee, you can also deploy an IBExpert real-time master-slave replication.

Is faster hardware always the solution?

Unfortunately not, because very often it is long-running transactions or other faulty techniques which are responsible for the performance issues, and which cannot be compensated for by even the fastest hardware. Initially it is essential to locate the causes. Major differences in the database size following a backup/restore indicate an often underestimated symptom.

If the customer database is significantly larger during normal operation than the quantity of new user data indicates, and the database size following a backup/restore is significantly smaller, there are probably a great many record versions present in the database, which were generated by outdated transactions and which could not be cleaned up by the garbage collector. The aim is to first track down the cause. In the IBExpert full version you can find an overview of all current transactions in the *Service | Database Monitor* menu. Important: To view *all* transactions, you must be either the SYSDBA or the database owner, i.e. the user who created the database.

In the list of open transactions you can see the start time of all active transactions. The following rules apply for applications that we create for customers, such as for example our modular ERP, www.brp-software.com:

- Transactions that are older than one minute should be avoided
- Transactions that are older than ten minutes should be an absolute exception
- Transactions that are older than one hour are totally unacceptable

We can adhere to these rules thanks to an architecture that is not based on open datasets. Whenever the BRP software retrieves data from the database or writes back to the database, this takes place in a transaction which is subsequently immediately closed. Even with 50-200 concurrent users this results in a maximum of 2-4 simultaneously open transactions, and to a corresponding excellent server performance. In reality, we discover server transactions on many client databases, which were often launched when starting the program at the beginning of the day, and often from the day before or even older.

The number of record versions in the Database Statistics shows the effects of old open transactions. Each night the Firebird service is provisionally restarted, to give the garbage collector a chance to carry out its work, without having to wait for outdated transactions.

This does not remedy the cause, but ensures at least that in the morning the performance at the customer's is tolerable. Before going on to schedule a further Firebird restart again at noon, the causes should be identified and rectified. After all, if you have a fever you do not sit yourself down in a refrigerator, even though that is sure to cure your fever, and you may never suffer from a fever again.

We also always deploy a script, which transfers information from the database monitor into a static table, to log, for example, as many critical SQLs as possible, discernible by their high values on Indexed Reads, Non Indexed Reads, Page Reads or Page Writes. Whether this is due to missing or wrongly set indices, unusual SQL commands, or due to an unfortunate use of stored procedures, this should now be analyzed step by step. We then use the *Performance Analysis* and Trace API in IBExpert and Firebird, to obtain detailed information about the SQL commands and induced Triggers etc.

The solution: training or consulting by IBExpert database experts!

We are contacted by many clients requesting our help to optimize their software, due to its inability to work without open datasets. Over the years we have developed appropriate techniques, which can also be incorporated into existing software architectures without jeopardizing the whole system.

Visit our Firebird Developers Days, organized at regular intervals, to find out more about our performance, shadow and replication techniques. Please refer to our website for dates in your region, or enquire by email to sales@ibexpert.com. The Firebird Developer Days can also be booked worldwide as corporate training.