

Firebird

Is it better to use single or multiple database files?

Performance is the same when all files are on the same disk. But, you can split multi-file database on several disks, so it can speed up queries a bit if you have multiple users. However, you might already be spreading your main system, swap and temporary space to separate disks, which gives us three or four disks already. Adding more RAM in order to cache more database pages is usually much better investment than adding more disks.

Multiple files are really only needed when your database grows and reaches the filesystem limit for a single file. Otherwise it is probably not worth the effort.

What are those NPTL packages?

In Linux kernel 2.6, a new threading library has been introduced. It is known as Native POSIX Thread Library or NPTL. This library enables programs that use native POSIX threads to run very effectively. For example, it takes NPTL program about two seconds to start 100.000 threads, while non-NPTL kernel would need around 15 minutes for it.

NPTL builds only exist for SuperServer as it uses threads for each connection (Classic uses a separate process for each connection). You should download and install the appropriate Firebird package for your system: NPTL package on NPTL system and non-NPTL on non-NPTL. If you are unsure, you can check with:

```
getconf GNU_LIBPTHREAD_VERSION
```

You should get something like: NPTL 2.3.5.

Make sure you don't install the wrong version as it can cause malfunction of Firebird - slow connections, sporadic failure to attach the database, etc.

Why did 'gbak -r' overwrite my database?

That '-r' switch caused a lot of problems so far. The general advice is to never use -r, as R stands for 'Replace' not 'Restore':

```
gbak -b backup  
gbak -r replace  
gbak -c create
```

You should always use -c to create a new database from backup and then simply rename the database file. This will save you a lot of trouble as using temporary file ensures that nobody else is trying to access it until it is ready. When using -c switch, GBak will warn you if the database file already exists and it would abort the restore. Also, if you use -r and restore fails for some reason you won't have any functional database as the old one is deleted when restore starts.

This has been changed in Firebird 2, now -R means RECREATE database, and will not overwrite existing database file. If you desire the old behaviour of -R switch, you can use -REP or -R -O, where -O means OVERWRITE.

Firebird

What's the limit on the Firebird database size?

You'll probably hit the filesystem limit long before Firebird's.

Here are some facts:

Windows:

FAT12 has a 32MB file size limit
FAT16 has a 2GB file size limit (Windows NT 3 or newer)
FAT32 has a 2GB file size limit
FAT32 has a 4GB file size limit (Windows NT5 / 2000 and newer)
NTFS has a 4GB file size limit (Windows NT)
NTFS has a 16EiB file size limit (Windows NT5 / 2000 and newer)

Linux:

EXT2/EXT3 (with 4kB blocksize) has a 2TB file size limit
EXT2/EXT3 (with 8kB blocksize) has a 64TB file size limit
ReiserFS 3.6 has a 1EiB file size limit
XFS has a 8EiB file size limit
JFS (with 512b blocksize) has a 8EiB file size limit
JFS (with 4kB blocksize) has a 8EiB file size limit

Even if you hit one of those, Firebird supports multi-file databases.

Why doesn't Classic have CPUAffinity setting in firebird.conf?

Because CPUAffinity only applies to SuperServer. Classic can work on multiple CPUs without problems, while SuperServer needs that option to avoid context switching between processors with degrades performance dramatically.

Where is Firebird 2.0 on Ubuntu and Debian?

When only Firebird 1.0 and 1.5 were available, the Ubuntu and Debian packagers decided to build package for Firebird 1.5, but called it 'firebird2' to avoid confusion with just 'firebird' package. Firebird 1.5 was meant to be a migration to Firebird 2 anyway.

However, when Firebird 2.0 came out, it started to be a source of endless confusion. Here are the 2.0 Debian packages by Damyan Ivanov. Please note that those conflict with previous firebird2-* packages and you should remove those prior to installation.

Direct links:

<ftp://ftp.modsoftsys.net/public/pool/main/f/firebird2.0/>.

```
deb ftp://ftp.modsoftsys.net/public experimental main
deb-src ftp://ftp.modsoftsys.net/public experimental main
(source package name is firebird2.0)
```

To test super-server install firebird2.0-super, firebird2.0-common, libfbclient2 and firebird-utils. For classic-server get firebird2.0-classic, firebird2.0-common, libfbembed2 and firebird-utils.

Also, Firebird2.0 packages will be available in Ubuntu Gutsy:

http://packages.ubuntu.com/cgi-bin/search_packages.pl?keywords=firebird2.0&searchon=names&subword=1&version=gutsy&release=all

Firebird

Problem restoring large databases on Linux

When you restore large databases on Linux, you might see this kind of error:

```
I/O error for file "/tmp/fb_sort_XXXXXX"  
-Error while trying to read from file  
-No such file or directory
```

If usually happens when you have one big table and it takes a long time to build indexes. Firebird uses temporary file storage for this (/tmp by default) and some Linux distributions have cron jobs that clean up /tmp periodically. Turning off crond or changing the TempDirectories setting in firebird.conf should solve this issue.

What is index selectivity?

Index selectivity is a number that determines the effectiveness of index. Best possible selectivity is when all table records have a different value for the columns in index (this is typical for primary keys and unique constraints). Selectivity is quantified by index 'statistics' which is computed like this:

```
select 1.0/count(distinct(index_field)) from table;
```

Please note that this query would fail if the table has zero rows (due to division by zero). To prevent this, you can use NULLIF function:

```
select 1.0/nullif(count(distinct(index_field)), 0) from table;
```

As you can see, higher statistics mean that there aren't many different values in the table column. If you have a Y/N or true/false field, it would have statistics value of 0.5 which is worst possible selectivity. As statistics approach zero, index gets more effective.

Index statistics can be read from RDB\$INDICES system table. Statistics range from zero (the best selectivity) to 0.5 (the worst) with a special value of 1 (one) which means that all table rows have the same value for index column. Having index on such field is completely useless. You would never get zero selectivity, the lowest value is actually: 1/number of table records.

Index selectivity should be recalculated periodically because inserting, updating and deleting records changes it. Firebird calculates it only when index is created and when database is restored from backup. So make sure you recompute statistics after big changes in data. It can be done by running:

```
SET STATISTICS INDEX index_name;
```

Using EXECUTE BLOCK in Firebird 2 or higher you can recalculate statistics for all indexes using this simple SQL statement:

```
set term !! ;
```

```
EXECUTE BLOCK AS  
declare variable index_name VARCHAR(31);  
BEGIN  
for select RDB$INDEX_NAME from RDB$INDICES into :index_name do  
execute statement 'SET statistics INDEX ' || :index_name || ';' ;  
END!!
```

```
set term ; !!
```

Firebird

What file and directory permissions to set for database files on Linux?

It's wise to keep your databases in a directory with following permissions (770 in chmod terminology):

```
drwxrwx--- 2 firebird firebird 1136 2007-05-31 18:21 databases/
```

If it's a development machine and you wish to be able to manipulate files directly, add yourself to the 'firebird' group. Please note that you usually need to completely log off before the group membership changes are applied to your account on Linux.

As for the database files, use the following privilege (660 in chmod terminology):

```
-rw-rw---- 1 firebird firebird 6123520 2007-05-14 14:49 database.fdb
```

This ensures that only the firebird user (typically the one that runs the Firebird server) and members of firebird group (admins on the machine) will be able to access databases directly.

Note that you must give write access for database files, even if users will only run SELECTs. It because each transaction number needs to be recorded in database file.

How to recover a corrupt Firebird database?

Here's a short step-by-step walkthrough:

- * stop the Firebird server (and disable incoming connections)
- * make a copy of database file (or two copies) and work on it
- * use GFIX with -v option to validate the database file
- * use GFIX with -v and -f to do full validation

If problem is not too serious, you can try to backup the broken db and restore under a new name:

- * use GFIX -mend to prepare corrupt database for backup
- * use GBAK -b -g to backup the database. -g disables garbage collection (FAQ [#41](#))
- * use GBAK -c to restore backup to a new database.

If you succeed, you have fixed the problem and have a functional database. If not, you can try to create an empty database with the same structure and pump the data to it (see FAQ [#20](#)).

If all fails, you can try IBSurgeon tool, which is able to fix most problems and extract data. Also, IBSurgeon's website has a detailed explanation of causes of database corruption and ways to fix it:

http://ib-aid.com/option.com_content/task.view/id.58/Itemid.62/

Firebird

How to prevent firebird.log file from filling up the disk partition?

Here are some tips:

a) create a scheduled task or cron job that will truncate or rotate the log file. By rotation, we mean renaming the files in such way that you always have a number of previous logs available. Example:

```
delete firebird.log.5
rename firebird.log.4 firebird.log.5
rename firebird.log.3 firebird.log.4
rename firebird.log.2 firebird.log.3
rename firebird.log.1 firebird.log.2
rename firebird.log firebird.log.1
```

This way you'll always have last five logs available, and those too old get deleted. You can also use zip, rar, bzip2 or some other packer to compress the old log files. Since they are plain text, they compress very well.

b) redirect logging to void. For example, on Linux, you can do it by creating a symlink to /dev/null instead of the regular log file:

```
# cd /opt/firebird
# rm -f firebird.log
# ln -s /dev/null firebird.log
```

Please note that you really shouldn't be doing this, as you will lose all valuable diagnostic information. It's better to try to find what are the errors, what's causing them and fix the problem at the source.

What is the benefit of Vulcan or Firebird 3?

The main benefit will be a multithreaded engine that can use multiple CPUs effectively. Current versions of Firebird have problems:

SuperServer does not work on multiple CPUs at all. You have to set CPUAffinity in firebird.conf to lock it to a single processor. This is fine when load is not big and your machine acts as something else as well (web server for example), so other applications can use the other CPU. However, it does not do SMP work.

Classic works fine with SMP, but doesn't scale very well. The main problem are locks which are delivered between processes using signals. If you have a lot of connected users (300 for example) on Quad CPU machine, it would mean that all 300 processes need to get lock info from those 4 processes using the database. This communication clutter creates a lot of traffic and load. A proper multithreaded engine that can work on multiple CPUs would solve this, as the communication would be done within a single process without hogging the operating system facilities.

Firebird

What are BLOB subtypes?

Blobs can store infinite amounts of data. They can be used to store text, images, videos, audio, or any other kind of data. Retrieving and writing data to blobs is done with separate functions, so it is usable to have some basic string (CHAR, VARCHAR) operations available for textual blob. Therefore the sub types were added (as blob is a type). The subtypes are:

- 0 - binary data (image, video, audio, whatever)
- 1 - text (basic character functions work)
- 2 - BLR (used for definitions of Firebird procedures, triggers, etc.)

User applications should only use subtypes 0 and 1.

Is it dangerous to run sweep on live database with active users?

No. Sweep is just another process accessing the database. All other users can keep working without any problems. Sweep only cleans up the records that nobody is using, so there is no chance to lose some data or see some invalid data.

The only issue with sweep is that it consumes a lot of server's resources, and if you have a large database it might hog the server until it completes. So, even there are no data integrity concerns, if performance is important, it's better to run sweep on off-hours, or periods of low user activity.

How to set the Firebird date format or why doesn't dd/mm/yyyy work?

In general, you shouldn't ever be using the strings user entered directly in your queries. Convert the string into some kind of Date object and pass it to Firebird as query parameter.

If you still wish to use strings, here are the formats that Firebird supports:

DD.MM.YYYY
MM/DD/YYYY
YYYY-MM-DD

In all of them, year can be written with 2 or 4 digits. The same format is also used in timestamps.
