

Firebird - SuperServer, ClassicServer or SuperClassic?

One of the most important decisions when deploying a Firebird server is the choice of server type. This choice is made during installation.

If it's hard enough to make an informed decision today, just wait until Firebird 2.5 is out. There will be 3 options: SuperServer, ClassicServer and SuperClassic.

The great differences among them are the page cache and the way the server handles the processes and threads that execute your statements.

Select Components

Which components should be installed?

Select the components you want to install; install. Click Next when you are ready to c

Full installation of Server and developmen

- Server components
- Classic Server binary
- Super Server binary

SuperServer

In SuperServer there is only one page cache and it is shared among all client connections.

Because it is shared, this cache is very efficient. When several clients access the same database areas every client benefit from a big and well fed cache.

For example, when client A issues:

```
SELECT NAME FROM CUSTOMERS WHERE ID = 1;
```

a few pages related to table CUSTOMERS and to the primary key index are loaded into the cache.

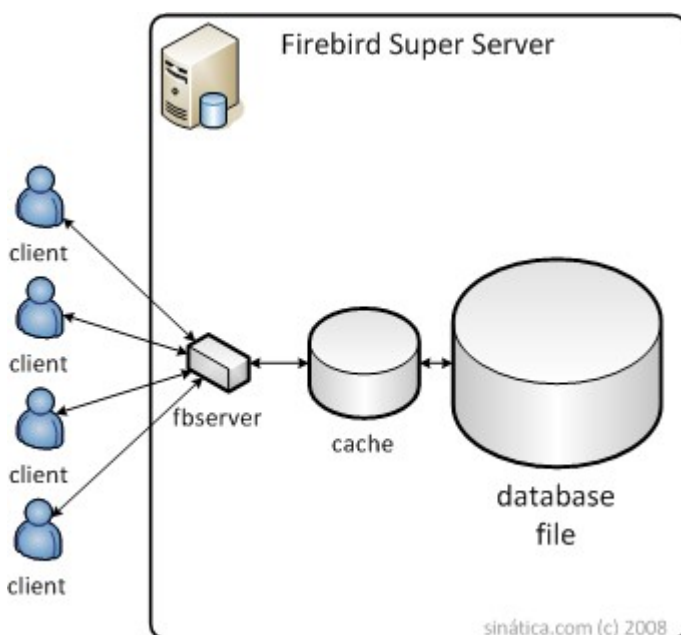
When client B issues:

```
SELECT NAME, ADDRESS, PHONE FROM CUSTOMERS WHERE ID = 2;
```

it benefits from the shared cache because the pages this statement needs are already in cache.

Also note that there is only a single process that all clients connect to.

Check the diagram:



SuperServer has scalability problems. If you install it on a multi-CPU computer (which is very likely nowadays) it will only use one of the CPUs¹. This is not a problem for small deployments

or environments where the server has activities other than the database.

For example, you have a dual-core server and want to use it as a file, web, print and database server. It is not a problem that Firebird uses only one CPU because the server has other activities to occupy the other CPU. And you get the benefit of a lightweight, small-footprint database.

But for big operations, where you want the database to use every cycle of server CPU, SuperServer can be frustrating.

¹ Except if you run more than one database. Starting from Firebird 2.5, Superserver will use more than one CPU that way. One for each database.

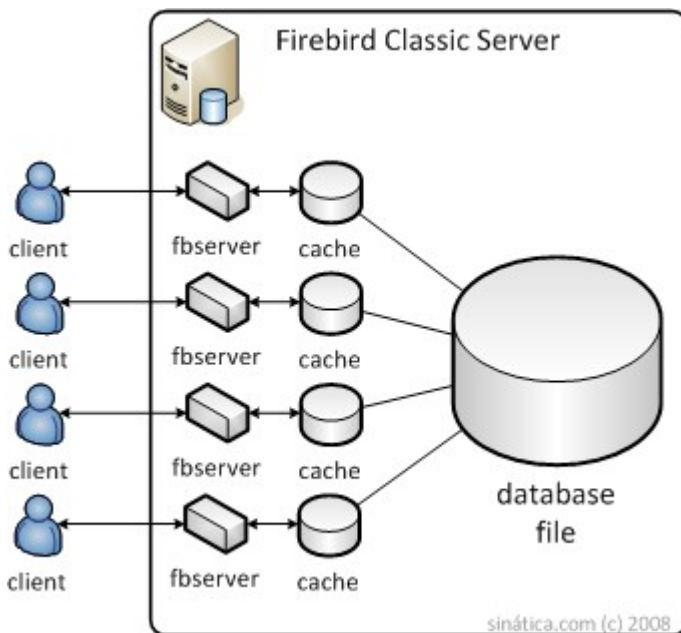
ClassicServer

In ClassicServer each client has its own page cache and is connected to a dedicated process.

This dedicated cache is much less efficient. If two clients access the same database area, this area will be copied to each client's cache. Using the example above, when client B issued the statement, it would not get the benefit of an already filled cache. Instead, Firebird would have to access the disk again to answer the request.

Besides, cache synchronizations is done on disk. This considerably increases the I/O cost on high-concurrency environments.

Check the diagram:



One great benefit of this model is the resilience offered by the multiple processes. If one of them has problems only the client attached to it will be disconnected. Everything else keeps running.

The other great benefit is scalability. I believe this is responsible for most Classic deployments out there. Even in cases where the dedicated cache is inferior to the shared cache, Classic's scalability make up for it. Just add hardware and bingo, your database server is faster.

But this scalability does not come free. Imagine you have 200 simultaneous clients. It's 201 processes. One for each client and another to listen to new connections. Your operating system must manage all this processes and keep them in sync. They consume a lot of kernel resources which means Classic can be relatively slow.

Check this example: Firebird 2.5 Alpha 1 Classic with 7 attached clients. It's 8 processes, 18 threads, 1050 handles.

(from the screen below you'll see that my Vista is Brazilian Portuguese ;))

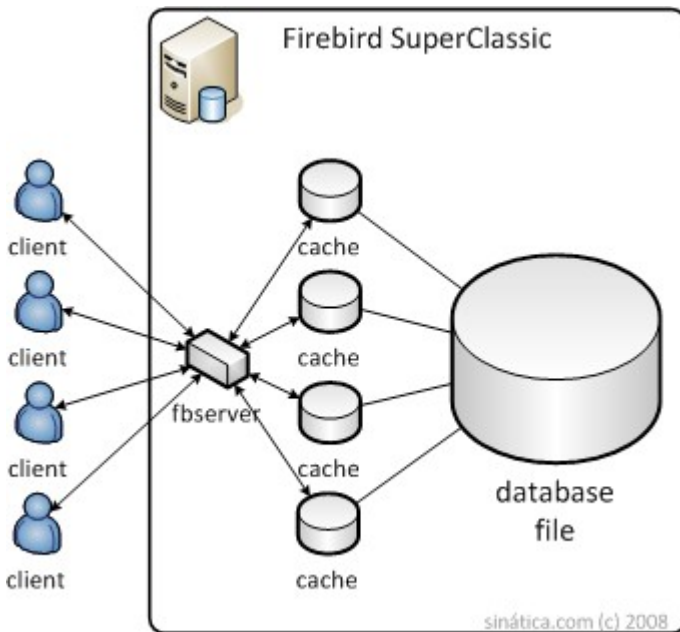
Nome da Imagem	Identifi...	Threads
fb_inet_server.exe	133	2
fb_inet_server.exe	133	2
fb_inet_server.exe	134	2
fb_inet_server.exe	116	4
fb_inet_server.exe	135	2
fb_inet_server.exe	133	2
fb_inet_server.exe	133	2
fb_inet_server.exe	133	2

SuperClassic

starting from Firebird 2.5

The Firebird development team decided to build Firebird 3.0 based on Classic. Firebird 3.0 will be completely SMP friendly. SuperClassic is the first step in that direction. It is an evolution and solves Classic's biggest problem: all those processes make it slow and make maintenance harder.

Welcome to SuperClassic: A single process with a dedicated cache.



Looking this way and considering the name, it may sound like a hybrid between Classic and Super but it's not. What they did was put all those processes inside threads. Now each client has a dedicated thread inside a single process.

Creating hundreds of threads is much cheaper than creating hundreds of processes and there is no loss of scalability. Cache synchronization is done directly in memory which reduces I/O costs. Other controls that used to be inter-process are now inter-thread and much faster.

Check a comparable example: Firebird 2.5 Alpha 1 SuperClassic with 7 attached clients, 1 process, 6 threads, 172 handles.

Nome da Imagem	Identifi...	Threads
fb_inet_server.exe	172	6

Conclusion

This compilation of common cases is a suggestion and serves as a guide, a starting point in your choice. Your deployment may have details not accounted here.

SuperServer

- Small databases or databases with little access
- Small servers
- Environments where the shared cache is more desirable than SuperClassic's scalability

ClassicServer

- Environments where stability is the top priority
- Multi-processor servers
- Big databases with hundreds of users

SuperClassic

- Multi-processor servers
 - Big databases with hundreds of users
 - Environments where the dedicated cache is more desirable than SuperServers shared cache
 - Environments where ClassicServer no longer scales well
-

<http://www.sinatica.com/>