Firebird 2.5 – ¿SuperServer, ClassicServer o SuperClassic?

Tomado del blog de www.sinatica.com

Link: http://www.sinatica.com/blog/en/index.php/articles/firebird-superserver-classicserver-or-superclassic

Autor: Douglas Tosi

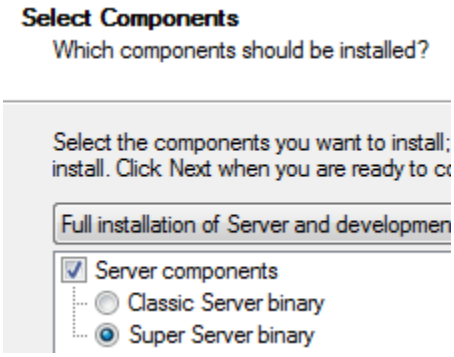
Traducido por: Javier Oros para www.clubdelphi.com a 09/Abril/2010

Revisión de la traducción: Casimiro Notevi (casimiro.clubdelphi@gmail.com)

Versión del documento: 1.1 a 12/Abril/2010

Nota del traductor: Este documento fue creado originalmente haciendo un análisis cuando estaba disponible Firebird 2.5 Alpha 1. Cuando se hizo esta traducción ya estaba disponible Firebird 2.5 Release Candidate 2.

Firebird 2.5 – ¿SuperServer, ClassicServer o SuperClassic?



Una de las decisiones más importantes al implementar un servidor Firebird es la elección del tipo de servidor. Esta elección se realiza durante la instalación.

Si a usted se le hace demasiado difícil tomar una decisión informada hoy, solo espere a que Firebird 2.5 esté fuera. Habrá tres opciones: SuperServer, ClasiccServer y SuperClassic.

Las grandes diferencias entre ellos son la página de caché y la forma en que el servidor maneja los procesos e hilos que ejecutan sus declaraciones.

SuperServer

En SuperServer hay solamente una página de caché y se comparte entre todas las conexiones de los clientes.

Debido a que es compartida, esta caché es muy eficiente. Cuando varios clientes acceden a las mismas áreas de la base de datos cada cliente se beneficia de una gran y bien alimentada caché.

Por ejemplo, cuando el cliente “A” cuestiona:

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

algunas páginas relacionadas a la tabla CUSTOMERS y a la clave primaria (primary key) se cargan en caché.

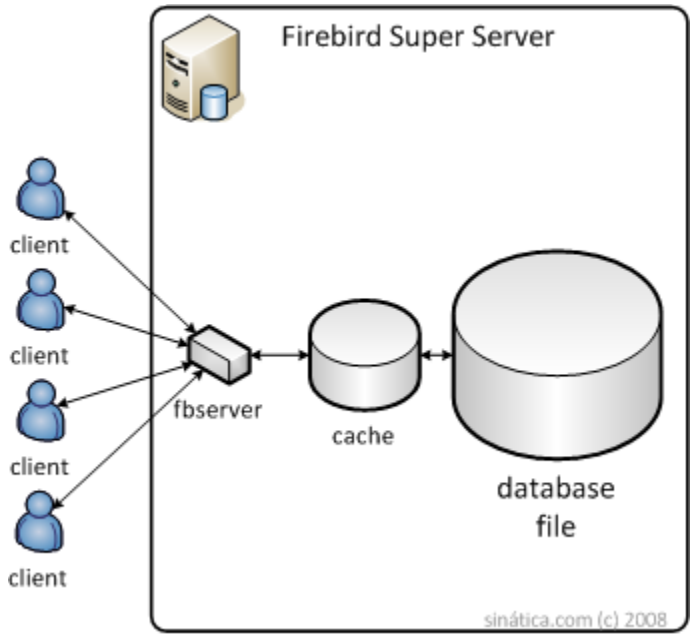
Cuando el cliente “B” cuestiona:

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

se beneficia de la caché compartida, porque las páginas que esta declaración necesita ya están en caché.

También tenga en cuenta que solo hay un único proceso que conecta a todos los clientes.

Verifique el diagrama:



SuperServer tiene problemas de escalabilidad. Si usted lo instala en una computadora multi-procesador (lo cual es muy probable hoy en día) este solo usará uno de los procesadores[1]. Esto no es un problema para pequeñas implementaciones o entornos donde el servidor tiene más actividades que la base de datos.

Por ejemplo, usted tiene un servidor Dual-Core y desea utilizarlo como un Servidor de archivo, web, de impresión y de base de datos. No es problema que Firebird use solo un procesador, porque el servidor tiene otras actividades que ocuparon el otro procesador. Y usted obtiene el beneficio de una ligera y poco espaciosa base de datos.

Pero para grandes operaciones, donde usted quiere que la base de datos use cada ciclo del procesador del servidor, SuperServer puede ser frustrante.

[1] *Excepto si se ejecuta más de una base de datos. A partir de Firebird 2.5, SuperServer usará más de un procesador de esa manera. Uno por cada base de datos.*

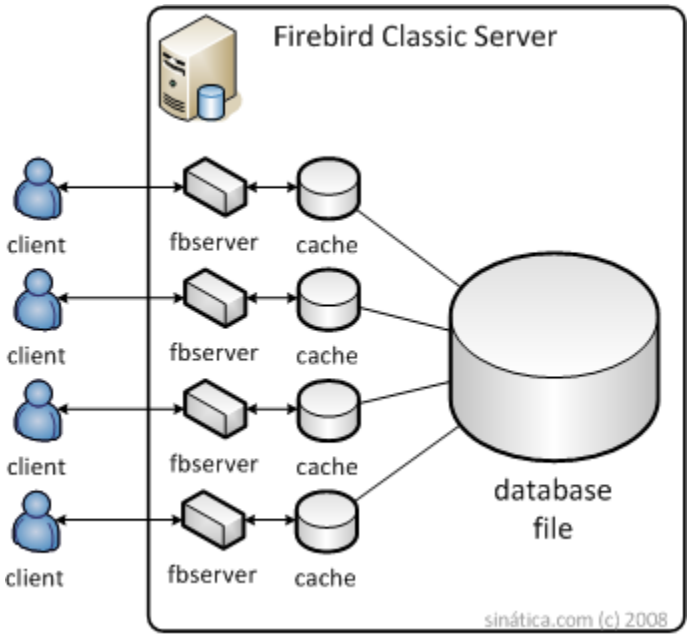
ClassicServer

En ClassicServer cada cliente tiene su propia página de caché y está conectado a un proceso dedicado.

Esta caché dedicada es mucho menos eficiente. Si dos clientes acceden a la misma área de la base de datos, esta área se copiara en la caché de cada cliente. Utilizando el ejemplo anterior, cuando el cliente “B” cuestione la declaración, no obtendrá el beneficio de una caché ya llena. En su lugar, Firebird tendría que acceder de nuevo al disco para responder a la petición.

Además, sincronizaciones caché son hechas en el disco. Esto aumenta considerablemente los costos de E/S en entornos de alta concurrencia.

Verifique el diagrama:



Una gran ventaja de este modelo es la flexibilidad ofrecida por los múltiples procesos. Si uno de ellos tiene problemas solo el cliente unido a él se desconectará. Todo lo demás sigue funcionando.

Otro gran beneficio es la escalabilidad. Creo que esto es responsable de la mayoría de las implementaciones Classic que andan por ahí. Incluso en los casos donde la caché dedicada es inferior a la caché compartida, la escalabilidad de Classic lo compensa. Sólo agregue hardware y bingo, su servidor de base de datos es más rápido.

Pero esta escalabilidad no es gratuita. Imagine que tiene 200 clientes simultáneos. Estos son 201 procesos. Uno para cada cliente y otro adicional para escuchar las nuevas conexiones. Su sistema operativo debe manejar todos los procesos y mantenerlos sincronizados. Estos procesos consumen una gran cantidad de recursos del kernel lo cual quiere decir que Classic puede ser relativamente lento.

Compruebe este ejemplo: Firebird 2.5 Alpha 1 Classic con 7 clientes unidos. Esto es 8 procesos, 18 hilos, 1050 manejadores.

(En la ventana de abajo verá que mi Vista está en portugués de Brasil ;)

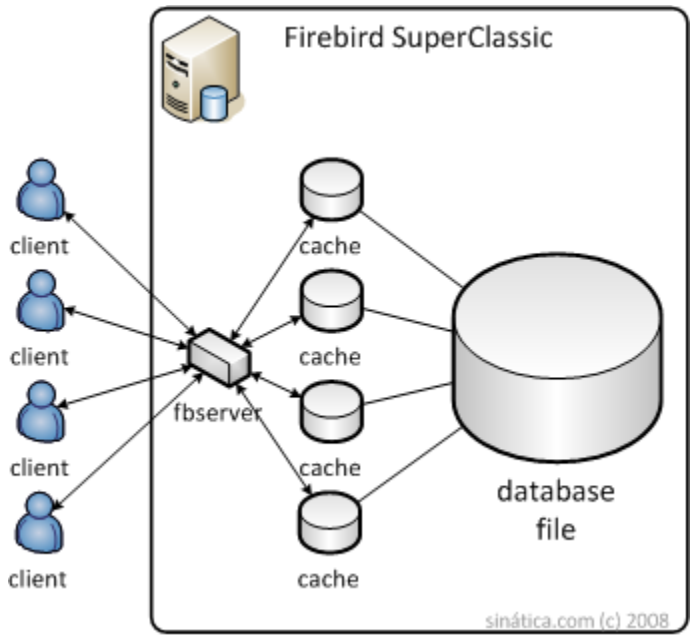
Nome da Imagem	Identifi...	Threads
fb_inet_server.exe	133	2
fb_inet_server.exe	133	2
fb_inet_server.exe	134	2
fb_inet_server.exe	116	4
fb_inet_server.exe	135	2
fb_inet_server.exe	133	2
fb_inet_server.exe	133	2
fb_inet_server.exe	133	2

SuperClassic

A partir Firebird 2.5

El equipo de desarrollo Firebird decidió construir Firebird 3.0 basado en Classic. Firebird 3.0 será completamente amigable con SMP. SuperClassic es el primer paso en esa dirección. Es una evolución y resuelve el problema más grande de Classic: todos aquellos procesos que lo hacen lento y hacen difícil el mantenimiento.

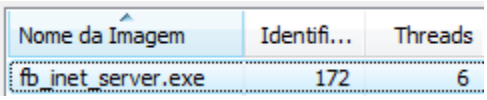
Bienvenido a SuperClassic: Un proceso único con una caché dedicada.



Viendo de esta manera y considerando el nombre, esto puede sonar como un híbrido entre Classic y Super pero no lo es. Lo que hicieron fue poner todos los procesos dentro de los hilos. Ahora cada cliente tiene un hilo dedicado dentro de un solo proceso.

La creación de cientos de hilos es mucho más barato que la creación de cientos de procesos y no hay pérdida de escalabilidad. La sincronización de la caché se realiza directamente en memoria con lo cual se reducen costos de E/S. Otros controles que solían ser inter-procesos ahora son inter-hilos y mucho más rápido.

Compruebe un ejemplo comparable: Firebird 2.5 Alpha 1 SuperClassic con 7 clientes unidos, 1 proceso, 6 hilos, 172 manejadores.



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

Conclusión

Esta recopilación de los casos más comunes es una sugerencia y sirve como guía, un punto de partida en su elección. Su implementación puede tener detalles no representados aquí.

SuperServer

- Bases de datos pequeñas o bases de datos con poco acceso
- Servidores pequeños
- Entornos donde la caché compartida es más deseable que la escalabilidad de SuperClassic.

ClassicServer

- Entornos donde la estabilidad es la prioridad principal
- Servidores multi-procesadores
- Grandes bases de datos con cientos de usuarios

SuperClassic

- Servidores multi-procesadores
- Grandes bases de datos con cientos de usuarios
- Entornos donde la caché dedicada es más deseable que la caché compartida de SuperServer.
- Entornos donde ClassicServer ya no escala bien