

1. Gestion5 SQL y FireBird 2.5

Indice

1. Gestion5 SQL y FireBird 2.5	1
Mejoras de la versión 2.5 de FireBird.....	2
Firebird 2.5 Release (octubre de 2010)	3
Otras características nuevas.....	3
▪ Mejoras Administrativas.....	3
▪ Otras adiciones y mejoras del lenguaje SQL	3
▪ Manejo de datos de Mejoras.....	3
▪ Adiciones API.....	4
▪ Soporte Internacional de Idiomas.....	4
Lista de las incompatibilidades conocidas entre Firebird 2.x y 2.5.....	4
Arquitecturas de servidor Firebird: Classic, SuperClassic y Superserver.....	6
2. Firebird 2.5 – ¿SuperServer, ClassicServer o SuperClassic?	8
▪ SuperServer	8
▪ ClassicServer	9
▪ SuperClassic.....	10
3. Conclusión	11
Cambios en los programas que realizan copias de seguridad.....	11
4. Borrador: CAMBIAR A FIREBIRD 2.5 los usuarios de GESTION5	12
Cambiar de Firebird 2.1 a Firebird 2.5.....	12
▪ ¿Cuándo necesito cambiar?	12
▪ ¿Puedo seguir en los puestos con fireBird 2.1 (o incluso el anterior Firebird 1.5)	12
▪ Ventajas “visibles” de FireBird 2.5	12
▪ Como elegir su mejor versión de Firebird	12
Conclusión:	12
Como proceder para actualizar.....	12
5. Ventajas para la programación en GESTION5 SQL	13
La Compatibilidad con FireBird 2.1 en GESTION5 SQL	13
6. Lista de Programas testeados y validados con FireBird 2.5	14

Mejoras de la versión 2.5 de FireBird

En resumen:

- El código SQLSTATE ha sido puesta a disposición como variable de contexto PSQL, para su uso en WHEN .. bloques de excepción, en la misma manera que GDSCODE y SQLCODE
- Ahora es posible escribir en las tablas temporales globales en una base de datos de sólo lectura
- Se mejoraron los Diagnósticos para las trazas de errores internos
- La utilidad **fbtracemgr** ahora hará un volcado periódico de salida
- Rendimiento de **gbak/restore** en las etapas de inserción de datos ha mejorado
- Las conversiones entre BLOB y otros tipos de datos ahora pueden efectuarse en las funciones de la API
- La API de servicios es ahora compatible con la restauración "sólo metadatos"
- Un interruptor de "instalación silenciosa" ha sido implementado para **realizar la instalación** en POSIX
- Los bytes no utilizados de valores VARCHAR en el buffer de avisos están puestos a cero
- La relación de compresión de grabación real se estima ahora en el optimizador
- El valor **MON\$STATEMENT_ID** ahora se mantiene constante entre las instantáneas de monitoreo
- Se definirá la opción **SO_KEEPLIVE** en el socket TCP cliente, como una medida para prevenir un tiempo de espera agresivo de sockets por los sistemas más modernos de Windows
- Lock Manager: ahora se pueden cancelar esperas que se convierten en interminables.
- Un puerto plataforma de v.2.5.1 para HPPA se ha hecho para Linux y Alfa

Firebird 2.5 Release (octubre de 2010)

A pesar de las mejoras de SQL no son el objetivo principal de esta versión, por primera vez, la gestión de usuarios se convierte en accesible a través de SQL **CREATE / ALTER / DROP USER** y se pueden aplicar en las sintaxis de **ALTER VIEW** y **CREATE OR ALTER VIEW**. Las mejoras en PSQL incluyen la introducción de transacciones autónomas y la capacidad de consultar otra base de datos mediante **EXECUTE STATEMENT**.

Otras características nuevas

Otras nuevas características y mejoras de esta versión incluyen:

▪ Mejoras Administrativas

- Sistema de seguimiento de auditoría y las sesiones de seguimiento del usuario a través de servicios de la API, que permitan monitorear y analizar todo lo que sucede en una base de datos en tiempo real
- Nuevo sistema de roles **RDB\$ADMIN** en la base de datos ODS 11.2 El usuario SYSDBA permite transferir sus privilegios a otro usuario en una base por base de datos
- Más información en los cuadros de seguimiento
- Cancelación asíncrona de conexiones
- Capacidad para los usuarios normales para controlar cualquiera de sus anexos, así como **CURRENT_CONNECTION**

▪ Otras adiciones y mejoras del lenguaje SQL

- Soporte para expresiones regulares mediante el predicado **SIMILAR TO**
- **ALTER COLUMN** para las columnas calculadas
- Transacciones autónomas dentro de un módulo PSQL (procedimiento almacenado, desencadenador o bloque PSQL ejecutable dinámicamente)
- Mejorar el acceso a los procedimientos almacenados de definiciones de vista
- Opcional **GRANTED BY** (o, en su defecto, AS) de **GRANT** y **REVOKE**, permitiendo que el otorgante sea un usuario que no sea el **CURRENT_USER** (por defecto).
- La sintaxis **REVOKE ALL** permite eliminar todos los privilegios de un usuario o rol a la vez
- Apoyo a predicados del tipo **WHERE SOME_COL = ? OR ? IS NULL**
- Eliminación de la condición de "palabra reservada" para todos excepto unas cuantas palabras clave que no están reservadas en el estándar SQL

▪ Manejo de datos de Mejoras

- Nueva función interna para convertir UUID CHAR (16) octetos cadenas a formato compatible con RFC4122 y viceversa

- Capacidad para pasar enteros de 32 bits y 64 bits en formato hexadecimal en binario cadena formatos literales literal y X-prefijo numérico
- **Adiciones API**
 - Las sentencias ahora devuelven un estándar SQLSTATE de código alfanumérico de 5 caracteres terminación según las especificaciones de SQL-2003
 - Nueva constante `DSQL_unprepare` disponible para su uso con `isc_dsql_free_statement` para des-preparar eficientemente los estados
 - **Soporte Internacional de Idiomas**
 - Cláusula `COLLATE` predeterminado para `CREATE DATABASE`
 - Posibilidad de cambiar el `COLLATE` predeterminado para un conjunto de caracteres utilizado
 - GBAK: Nuevas opciones de restauración `FIX_FSS_DATA` y `FIX_FSS_METADATA` para restaurar las bases de datos existentes con los datos `UNICODE_FSS` y / o metadatos correctamente sin recurrir a guiones y técnicas manuales
 - `COLLATION` distingue acentos para Unicode

Ver el [Documento completo](#)

Lista de las incompatibilidades conocidas entre Firebird 2.x y 2.5

Este documento describe todos los cambios que hacen v2.5 incompatible en modo alguno, en comparación con las versiones anteriores y por lo tanto puede afectar a las bases de datos y aplicaciones.

Por favor, lea las siguientes descripciones cuidadosamente antes de actualizar el software a la nueva versión de Firebird.

Instalación y configuración

- El proceso de **migración de base de datos** podría requerir algunas medidas especiales. Si la restauración de base de datos falla con el error "cadena con formato incorrecto" o "malformed string", preste atención a los archivos del directorio `/misc/upgrade/metadata` de la instalación y el uso de las nuevas opciones `-fix_fss_data` y `-fix_fss_metadata` de la línea de comandos de GBAK.
- Los Parámetros `OldParameterOrdering` y `CreateInternalWindow` de `firebird.conf` no reciben soporte y se han **eliminado**.
- Parámetros `LockSemCount` y `LockSignal` de `firebird.conf` no requieren un ajuste en la nueva implementación del administrador de bloqueo, por lo que se han **eliminado** también.
- Ahora **FireBird puede ser construido para ser reubicable en entornos POSIX**. Eso significa que el directorio de instalación se basa en Firebird ejecutables y bibliotecas, y las bibliotecas no están codificado cargado desde `/opt/firebird/lib`.

Para activar esta función, parámetro `-enable-binreloc` se debe pasar a `autogen.sh`. Se espera que esta función esté activada de forma predeterminada en la versión 3.0. Si ha utilizado para copiar las utilidades a diferentes lugares y se basan en las bibliotecas se están cargando desde `/opt/firebird`, que no funciona en este modo. El enfoque correcto sería la creación de enlaces simbólicos.

SQL Sintaxis

- Se introducen una serie de nuevas palabras clave reservadas. La lista completa está disponible aquí: /doc/sql.extensions/README.keywords. Por favor, asegúrese de que sus declaraciones DSQL y procedimientos / disparadores no contienen esas palabras clave como identificadores. De lo contrario, tendrá que utilizar los citados (en dialecto 3 solamente) o cambiarles el nombre.

Resultados de la ejecución de SQL

- Ya no se permiten stings ni blob UNICODE_FSS malformados.
- Antes de Firebird 2.5 de la cláusula SET de la instrucción UPDATE asignaba las columnas en el orden definido por el usuario con los nuevos valores de la columna tenga acceso inmediato a los trabajos posteriores. Esto no cumple con el estándar SQL. A partir de Firebird 2.5, sólo los valores de las columnas viejas (OLD.xxx) son accesibles a todas las asignaciones en la cláusula SET. Puede volver al comportamiento heredado a través del parámetro **OldSetClauseSemantics** de firebird.conf, si es necesario. Tenga en cuenta que este parámetro se proporciona como una solución temporal para los problemas de compatibilidad y se dejará de utilizar en futuras versiones de Firebird.

Utilidades

- **fb_lock_print** ahora requiere un nombre de ruta de base de datos con el fin de imprimir la tabla de bloqueos. Use un interruptor de línea "-d <Nombre de ruta>" de comandos para especificar una base de datos para analizar.

API

- Contenidos inapropiados TPB (buffer de parámetros transacción) es ahora rechazada por rutinas de la API isc_start_transaction () y isc_start_multiple (). Por ejemplo, no está permitido especificar las opciones "no wait" y "wait timeout" (distinto de cero) juntas, ni tampoco es posible indicar el modo "no record version" con el modo "snapshot" de transacción de aislamiento, etc.

Seguridad

- Los miembros de los grupos de Windows administrativos no se asignan a SYSDBA más por defecto. El mapeo se controla en función de cada base de datos utilizando el siguiente comando SQL:

ALTER ROLE RDB\$ADMIN SET/DROP AUTO ADMIN MAPPING.

Ver el [documento original](#) para más detalles.

Arquitecturas de servidor Firebird: Classic, SuperClassic y Superserver

	Classic Server	SuperClassic	Superserver
<i>Procesos</i>	Un proceso de escucha (<code>xinetd</code> en Linux, <code>fb_inet_server</code> en Windows) escucha al puerto Firebird y genera un proceso servidor independiente (<code>fb_inet_server</code>) por cada conexión de cliente.	Un solo proceso (<code>fb_smp_server</code>) escucha al puerto Firebird, y sirve todas las conexiones, utilizando hilos para manejar las peticiones.	Un solo proceso (<code>fbserver</code>) escucha al puerto Firebird, y sirve todas las conexiones, utilizando hilos para manejar las peticiones.
<i>Cache</i>	Cada proceso del servidor (= cada conexión) tiene su propia caché.	Espacios de caché independiente para cada conexión.	Un espacio de caché compartido para todas las conexiones.
<i>Las conexiones locales (Linux)</i>	Las aplicaciones pueden utilizar la biblioteca <code>libfbembed</code> para conexiones locales. Este es un cliente y el servidor, todo en uno, permitiendo rápido, E/S directa a los archivos de base de datos. El proceso de usuario debe tener derechos de acceso a nivel de sistema de archivos a la base de datos para que esto funcione. Las conexiones locales a través de la capa de red también son posibles, en este caso, sólo el proceso de servidor necesita acceso al archivo de base de datos.		Todas las conexiones locales se realizan a través de la capa de red, utilizando <code>localhost</code> (a menudo implícitamente). Sólo el proceso de servidor necesita tener acceso al archivo de base de datos.
<i>Las conexiones locales (Windows)</i>	En Windows, todas las variantes locales admiten conexiones seguras y fiables utilizando el XNET protocolo, sólo con el proceso de servidor que requieren derechos de acceso al archivo de base de datos. Acceso directo en el espacio de proceso de usuario requiere un paquete separado, Windows Embedded servidor .		
<i>Acceso simultáneo de múltiples motores</i>	Clásico y SuperClassic utilizan un sistema de bloqueo compartido. Esto permite el acceso de múltiples servidores (por ejemplo, un servidor de regular y uno o más servidores embebidos) al mismo tiempo.		Superserver adquiere un bloqueo exclusivo en el archivo de base de datos. No hay otros procesos pueden tener acceso a la base de datos durante el mismo tiempo.
<i>Multiprocesador / multinúcleo</i>	SMP (multiproceso simétrico) con el apoyo de la caja. El <code>CpuAffinityMask</code> parámetro <code>firebird.conf</code> se ignora. A diferencia Superserver, Classic y SuperClassic también pueden asignar múltiples conexiones a la misma base de datos para diferentes procesadores.		De Windows por defecto Superserver a usar el primer procesador lógico solamente, porque antes de la 2.5 se realizó mal en sistemas SMP. Para hacer uso de todas sus procesadores, establezca el <code>CpuAffinityMask</code> parámetro <code>firebird.conf</code> : 3 por 2 CPUs / núcleos, 15 para 4 CPUs / núcleos de 255 para 8 CPUs / núcleos. Linux Superserver ignora <code>CpuAffinityMask</code> .
<i>Tutor (Linux)</i>	The Guardian no se utiliza: <code>xinetd</code> escucha las	SuperClassic y Superserver son servidores de un solo proceso, que pueden ser vistos por el Guardian.	

	Classic Server	SuperClassic	Superserver
	conexiones entrantes y crea instancias de servidor, según sea necesario.		
<i>Tutor (Windows)</i>	Quando se ejecuta como una <i>aplicación</i> de Windows (en lugar de un servicio), no se puede utilizar el Firebird Guardian. Tenga en cuenta que la ejecución de Firebird como una aplicación es la única opción en Windows 9x, ME. El instalador de Windows no ofrece la opción Tutor en absoluto para Classic / SuperClassic.		Puede ser utilizado con el tutor en Windows, si se ejecuta como una aplicación o como un servicio.
<i>Eventos</i>	Desde la versión 2.5, todos los modelos se pueden utilizar los eventos Firebird en todas las circunstancias. Si el servidor está detrás de un cortafuegos o si las conexiones se realizan a través de un túnel seguro, un puerto eventos específicos debe ser asignado a la RemoteAuxPort variable en firebird.conf, y el servidor de seguridad o un túnel configurado en consecuencia.		

Las diferencias mencionadas anteriormente dan como resultado las siguientes ventajas y desventajas de los distintos modelos:

- Si un proceso de servidor clásico tradicional se bloquea, el resto de conexiones no se ven afectados. Si un SuperClassic o Superserver bloquea el proceso, todas las conexiones se bloquean.
- The Guardian da un poco de Fiabilidad adicional porque se reinicia automáticamente un servidor caído. En Windows, esto da Superserver una pequeña ventaja, especialmente si se ejecuta Firebird como una aplicación (servicios estrellados también pueden reiniciarse el sistema operativo).
- Classic tradicional utiliza menos recursos si el número de conexiones es baja. **SuperClassic y Superserver son más eficientes si el número de conexiones simultáneas crece.**
- El modo de acceso local proporcionado por Classic y SuperClassic en Linux, aunque muy rápido, no ofrece seguridad completa. Puede desactivarlo (ver la [Seguridad](#) capítulo), pero también perderá el beneficio de velocidad.
- Al trabajar con una aplicación de Windows Embedded, puede ser útil si se puede abrir la base de datos a través del servidor regularmente, al mismo tiempo, por ejemplo, para la inspección "en vivo" o para hacer una copia de seguridad. Esto sólo es posible si su servidor regular es un clásico o SuperClassic.

Como puede ver, ninguno de los tres modelos es mejor en todos los aspectos. Si, después de estudiar la información anterior, usted todavía no está seguro de cuál es el mejor para usted, 64 bits SuperClassic puede ser una buena opción, a menos que necesite la solidez a nivel de conexión adicional del clásico tradicional, donde un servidor de estrellarse no acabar con las demás conexiones. En los sistemas de 32-bits, SuperClassic puede quedarse sin memoria en condiciones de alta carga, en este escenario, se prefieren Superserver y (especialmente) Server Classic.

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.

Habrán tres opciones: **SuperServer**, **ClassicServer** 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" consulta:

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.

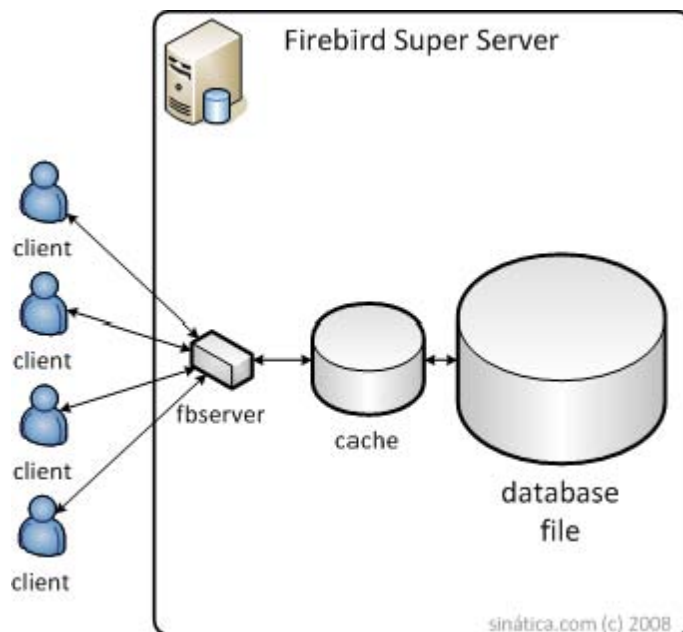
Cuando el cliente "B" consulta:

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!. 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.

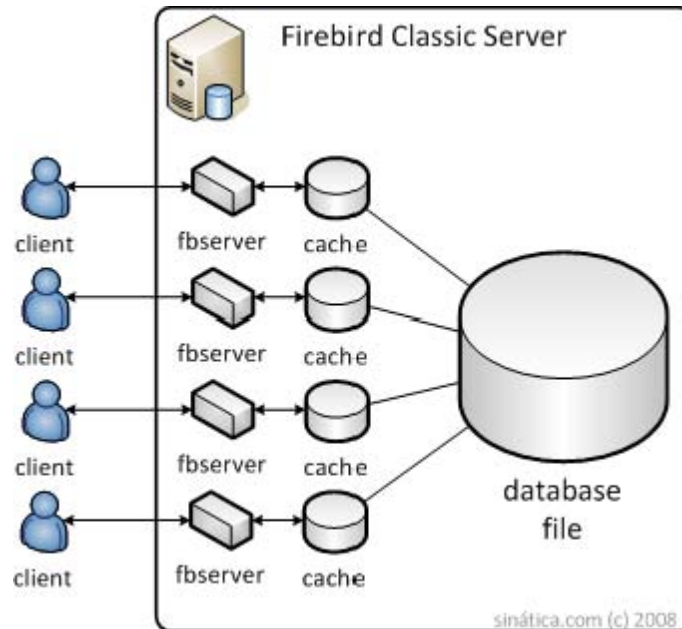
1 Excepto si se ejecuta más de una base de datos. A partir de Firebird 2.5, SuperServer usará más de un procesador de esa manera. Uno por cada base de datos.

▪ 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, las 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.

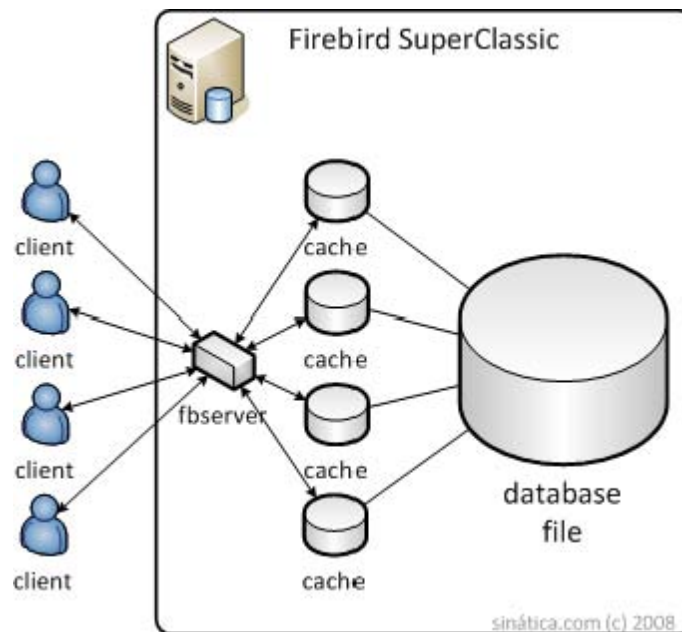
Compruebe este ejemplo: Firebird 2.5 Alpha 1 Classic con 7 clientes unidos. Esto es 8 procesos, 18 hilos, 1050 manejadores.

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 multi-procesadores (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 (en términos de consumo de recursos) 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

3. 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.

Cambios en los programas que realizan copias de seguridad

1) Por compatibilidad, en las asignaciones de los parámetros de conexión a B.D. que se realizan por código (Delphi) en los programas, se debe incluir el nuevo parámetro **lc_ctype=ISO8859_1** junto con los parámetros de Usuario y clave.

Ejemplo Delphi:

```
BaseDatosGESTION5: TIBDatabase;

BaseDatosGESTION5.Params.Clear;
BaseDatosGESTION5.Params.Add('USER_NAME='+Usuario);
BaseDatosGESTION5.Params.Add('PASSWORD='+Password);
BaseDatosGESTION5.Params.Add('lc_ctype=ISO8859_1'); //NUEVO PARAMETRO DE CONEXION
...
BaseDeDatosAdministracion.Params.Clear;
BaseDeDatosAdministracion.Params.Add('USER_NAME='+Usuario);
BaseDeDatosAdministracion.Params.Add('PASSWORD='+Password);
BaseDeDatosAdministracion.Params.Add('lc_ctype=ISO8859_1');//NUEVO PARAMETRO DE CONEXION
```

2) Cambios en el componente **TibManager** (C:\Desarrollo\Fuentes\Delphi6\ibman\ibman.pas) que se usa en **AdministracionG5SQL.exe** y en **AdministradorConta5SQL.exe** para realizar los procesos de **Restore** de las Bases de Datos.

El cambio en el componente consiste en detectar si es FireBird 2.5.x y añadir al proceso de **Restore** de las Bases de Datos los parámetros:

```
-FIX_FSS_M ISO8859_1
-FIX_FSS_D ISO8859_1
```

4. Borrador: CAMBIAR A FIREBIRD 2.5 los usuarios de GESTION5.

Cambiar de Firebird 2.1 a Firebird 2.5

▪ ¿Cuándo necesito cambiar?

Se recomienda cambiar cuando hay Servidores con multiprocesadores y/o multinucleos

▪ ¿Puedo seguir en los puestos con fireBird 2.1 (o incluso el anterior Firebird 1.5)

Con FireBird 2.1 si se puede, con FireBird 1.5 no.

▪ Ventajas “visibles” de FireBird 2.5

Mejor tiempo de respuesta en entornos multiprocesador / multinucleo
Mejor rendimiento de *gbak/restore*

▪ Como elegir su mejor versión de Firebird.

Quizá la ventaja **más visible** sea el rendimiento en CPU's con procesadores de varios núcleos. Si tiene Procesadores Core 2, Core 4, Core i3, Core i5, XEON con varios núcleos. Permite que “otras aplicaciones” usen algunos de esos núcleos y dejen a Firebird en exclusiva el uso de uno o varios núcleos para servir datos a los usuarios de Gestion5 y Conta5.

Conclusión:

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.

Classic

- 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.

Si tiene dudas y por las características de nuestros clientes ...

- Para 5 o menos usuarios : **SuperServer**
- Para Servidores de 5,10,.. 30 usuarios : **SuperClassic**

Como proceder para actualizar.

Hacer copias de seguridad en Gestion5 SQL y en Conta5 SQL. Desinstale Firebird 2,,1 e instale 2.5. vaya al administrador de Gestion5 SQL y refresque los usuarios. Repita esto mismo en Conta5 SQL Administrador

--- fin ---