**Asignación de espacio en disco para base de datos**

El tamaño que ocupan en el disco duro las Bases de Datos de Firebird no es tan sencillo de conocer como a primera vista se podría uno imaginar.

Cuando creas una Base de Datos de Firebird ésta no está vacía sino que ya tiene dentro suyo muchas tablas que luego serán usadas durante las tareas que se realicen en esa Base de Datos. A esas tablas internas se las conoce como “metadatos“.

Entonces, lo primero que notarás después de crear una Base de Datos si miras su tamaño es que éste es de cientos de kilobytes. Eso es debido a los metadatos. Está ok, así tiene que ser.

Luego, tú o los usuarios empiezan a realizar operaciones de manipulación de datos (inserciones, modificaciones, borrados) y allí puedes notar algo curioso: el tamaño de la Base de Datos parece ser muy grande para la cantidad de filas que hay en las tablas.

¿Por qué?

Porque cuando el Firebird necesita que la Base de Datos tenga más espacio para las filas que estás insertando, modificando o borrando no aloja espacio solamente para esa fila en particular sino que aloja mucho más espacio.

¿Y por qué hace eso, por qué no aloja solamente el espacio en disco exactamente necesario?

Por tres motivos:

Porque la tarea de alojar más espacio lleva tiempo, no es instantánea. Si cada vez que se inserta una nueva fila tendría que estar alojando espacio en el disco duro para esa fila entonces esas inserciones demorarían demasiado tiempo en bases de datos donde hay muchas inserciones concurrentes y los usuarios se quejarían de la extrema lentitud

Porque como el Firebird aloja el espacio en páginas esos continuos alojamientos fragmentarían excesivamente al disco duro

Porque si el disco duro se queda sin espacio entonces hay una gran posibilidad de que la Base de Datos aún tenga suficiente espacio libre previamente alojado para que pueda terminar todas las operaciones que se estaban realizando en ella, sin corrupción.

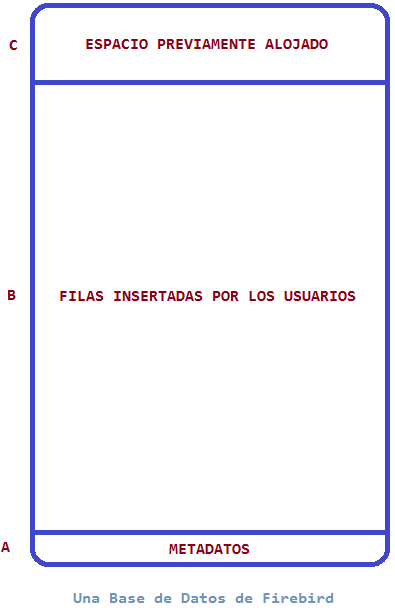
Entonces, alojar más espacio del necesario, para tener bastante espacio libre disponible tiene tres grandes ventajas: a) se consigue una gran rapidez en todas las operaciones de inserción, modificación, borrado, porque el espacio ya está disponible, b) no se fragmenta excesivamente el disco duro y c) si el disco duro se queda sin espacio libre es muy probable que la Base de Datos no se corrompa porque los datos que faltaban grabarse se grabarán en el espacio previamente alojado.

Claro que esto también tiene sus desventajas. Estas son: a) cada vez que hay que alojar más espacio ocurre una demora, y b) la Base de Datos puede estar ocupando en el disco duro mucho espacio y si la cantidad libre en el disco duro es escasa eso afectará a los otros programas.

Claro que la desventaja b) es muy improbable que ocurra en esta época en que los discos duros son gigantescos y muy baratos, además si se cuenta con un Administrador que verifique el espacio libre en el disco duro nunca debería ocurrir algo así.

Un gráfico que ilustra el concepto

En este gráfico podemos ver como se aloja el espacio en una Base de Datos de Firebird.



El espacio en C) está libre, nada hay ahí pero ya está asignado a la Base de Datos y para el Sistema Operativo es espacio ocupado por la Base de Datos y por ese motivo no puede ser usado por otros archivos. Cuando los usuarios insertan, modifican, o borran filas, esas inserciones, modificaciones, o borrados ocurren en C) y cuando el espacio en C) ya es pequeño se aloja más espacio a la Base de Datos, para que siempre tenga espacio libre a su disposición. Pero siempre esos alojamientos sirven para alojar a cientos o a miles de filas, no son para alojar a una o a dos filas sino a cientos o a miles de ellas. O sea que cada vez que se aloja un nuevo C) hay suficiente espacio en él para cientos o miles de filas.

¿Y cuántos bytes se alojan cada vez que la Base de Datos requiere de más espacio?

Cuando el Servidor descubre que la Base de Datos ya casi no tiene espacio libre debe reservar más espacio para ella. En el gráfico anterior la parte B) va creciendo hacia arriba haciendo que la parte C) sea cada vez más pequeña. Es entonces que el Servidor le aloja más espacio, según esta fórmula:

Al tamaño actual de la Base de Datos lo divide por 16

Nunca aloja más espacio que el establecido en la entrada DatabaseGrowthIncrement del archivo FIREBIRD.CONF (que por defecto es 128 Mb) ni menos de 128 Kb

Ejemplos:

Si la Base de Datos tiene 16 Mb entonces aloja 1 Mb (o sea, el tamaño de la Base de Datos dividido 16)

Si la Base de Datos tiene 16 Gb entonces aloja 128 Mb. No aloja 1 Gb sino que aloja 128 Mb. ¿Por qué? porque ese es el valor de la entrada DatabaseGrowthIncrement (que indica cual es el incremento máximo), salvo que haya sido cambiada. Recuerda: Nunca aloja más que el incremento máximo.

Si la Base de Datos tiene 16 Gb y la entrada DatabaseGrowthIncrement tiene el valor 268435456 (que equivale a 256 Mb) entonces aloja 256 Mb.

¿Es conveniente aumentar o disminuir el valor de la entrada DatabaseGrowthIncrement?

En general, el valor por defecto de 128 Mb es bastante bueno para la gran mayoría de las situaciones y podríamos dejarlo así, pero en bases de datos que tienen muchísimo movimiento y crecen mucho y rápido puede ser conveniente aumentarlo, para que el Firebird no esté a cada rato alojando nuevo espacio. Disminuirlo no se justifica (salvo que el disco duro tenga poquísimo espacio libre) porque disminuir significa aumentar la fragmentación del disco duro algo que nunca es conveniente porque un disco duro fragmentado tarda más en ser leído.

El valor de la entrada DatabaseGrowthIncrement se expresa en bytes, no en megabytes. Por lo tanto en ella por defecto veremos el número 134217728 que equivale a 128 Mb.

**Asignación de espacio en disco para tablas**

¿Se puede cambiar el tamaño en bytes de las páginas?

Sí, se puede, para eso se debe usar el programa GBAK. Al restaurar un backup se puede especificar el tamaño que tendrán las páginas de la Base de Datos restaurada. La opción para ello es -page\_size [tamaño], por ejemplo: -page\_size 8192

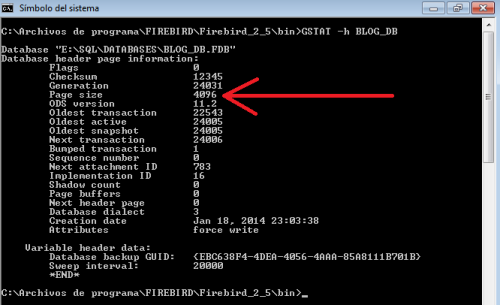
Recuerda que el tamaño de las páginas de la Base de Datos original no cambia, el que cambia es el tamaño de las páginas de la Base de Datos restaurada.

¿Y qué ocurre si especifico un tamaño que no sea ninguno de los anteriores?

Entonces el Firebird usará uno de los anteriores. Si el tamaño que especificaste es menor que 4096, usará 4096. Si es mayor que 4096, usará el tamaño predefinido que sea menor al que especificaste. Por ejemplo, si especificaste 16000, usará 8192 porque 8192 es menor que el tamaño que especificaste.

¿Y cómo puedo saber el tamaño de las páginas de mi Base de Datos?

Hay dos formas:

1. Usando el programa GSTAT con la opción -h, como vemos a continuación: 

El número que verás a continuación de “Page size” siempre será uno de los siguientes: 4096, 8192, 16384

1. Haciendo un SELECT a la tabla MON$DATABASE
2. SELECT
3. MON$PAGE\_SIZE
4. FROM
5. MON$DATABASE

NOTA: Versiones antiguas de Firebird también permitían 1024 y 2048, pero ahora esos tamaños ya son obsoletos.

¿Cuántas clases de páginas distintas hay?

Aunque todas las páginas tienen exactamente el mismo tamaño, se las utiliza para distintas cosas. Para saber en que se utiliza cada página tienen un número hexadecimal que las identifica al cual se le llama “tipo de página”.

0x01. Es la Header Page (página de cabecera) y solamente hay una. En ella se guardan datos tales como: el tamaño de las páginas, la versión del ODS, la transacción más antigua, la última transacción activa, la siguiente transacción, etc.

0x02. Es una Page Inventory Page (página de inventario). Puede haber varias. Su sigla es PIP. Siempre la primera PIP está a continuación de la Header Page. Se usa para saber cuales son las páginas que tiene actualmente la Base de Datos y si están libres para ser usadas (o sea, “disponibles”) o no.

0x03. Es una Transaction Inventory Page (página de inventario de las transacciones). Siempre hay al menos una. Aquí se guardan el número de las transacciones y su estado (activa, limbo, confirmada, desechada). El mayor número posible de transacción es 2.147.483.647, cuando una Base de Datos alcanzó a ese número de transacciones se debe hacer un ciclo backup/restore para que el número de la transacción regrese a 1.

0x04. Es una Pointer Page (página de punteros). Hay al menos una por cada tabla (de metadatos o del usuario) que tiene la Base de Datos. En la Pointer Page se guardan los números todas las páginas de datos que pertenecen a una sola tabla. Eso significa que en una página de datos solamente puede haber filas de una sola tabla, nunca se mezclan filas de una tabla con las de otra tabla en una página de datos. Las tablas grandes tienen muchas Pointer Page.

0x05. Es una Data Page (página de datos). Hay al menos una por cada tabla que tiene filas (tanto sean metadatos como del usuario). Todos los datos de esta página corresponden a una sola tabla.

0x06. Es una Index Root Page (página del índice raíz). Cada tabla de la Base de Datos tiene una Index Root Page, la cual describe los índices que tiene esa tabla. Aunque una tabla no tenga índices igual tiene una Index Root Page.

0x07. Es una Index B-Tree Page (página de índice B-Tree). Si una tabla no tiene índices, no tendrá una página de tipo 0x07. Todos los datos de una página Index B-Tree corresponden a un solo índice de una sola tabla.

0x08. Es una BLOB Data Page (página de datos para columnas de tipo BLOB). Solamente existen para tablas que tienen al menos una columna de tipo BLOB. En esta página se guarda el contenido de esas columnas. Todos los datos corresponden a una sola columna de una sola tabla.

0x09. Es una Generator Page (página de generadores). Hay al menos una por cada Base de Datos, aunque ningún generador (también llamado “secuencia”) haya sido definido.

0x0A. Es una Write Ahead Log Page (página de escribir por delante el log). Hay al menos una por cada Base de Datos, pero actualmente no se la está usando, es un desperdicio de espacio, y probablemente ya no exista en Firebird 3.0

**Asignación de espacio en disco para usuarios**

**Tipo de instalación de Firebird**

El gestor de bases de datos Firebird se puede instalar de dos modos, aunque la opción por defecto es habitualmente la más interesante, no está de más conocer el segundo formato, que en determinados casos (maquinas con varios procesadores, mucha memoria y para 20 o más usuarios) nos puede ayudar a mejorar el rendimiento.

La instalación por defecto es la denominada "SuperServer", el segundo formato se denomina "Clasic Server".

Se diferencian en el modo de organizarse para dar servicio a múltiples conexiones, y esta organización influye en el consumo de memoria RAM.

"SuperServer" reserva un número de MB por cada base de datos a las que se acceden. En este formato hay un único proceso, fbserver, lo que hace que el consumo de memoria sea fijo, independientemente del número de usuarios conectados. Por otra parte al ser un único proceso no se aprovechan máquinas multiprocesador: éste proceso ocupa uno de ellos y se olvida del resto de procesadores.

La versión 3 de Firebird, que debería salir a la calle en algún momento del 2008 - fecha aproximada- eliminará esta limitación en el número de procesadores y probablemente estas dos formas de instalar.

"Classic Server" crea un proceso y reserva un número de MB por cada conexión -usuario- y de este modo se aprovechan todos los procesadores que tengamos, pero la memoria RAM necesaria aumenta conforme añadimos más usuarios al sistema.

Solo se recomienda instalar de este modo si: Tenemos un servidor multiprocesador y suficiente memoria (2 Gigas por ejemplo) y el modo SuperServer no nos da un buen rendimiento... pero hay que tener en cuenta que la RAM necesaria se incrementará al añadir usuarios.

Si instalamos esta versión, debemos tenerlo en cuenta al elegir el tamaño del cache que asignamos a cada conexión (discutido a continuación), ya que en este caso no serán MB por base de datos, serán MB por usuario. De hecho, el valor por defecto de este parámetro es 2048 páginas (8 MB) en el SuperServer pero de solo 75 (300 KB) en el Classic server.

**Consejos útiles**

Veamos algunas cosas que pueden cambiar en la configuración "de fábrica" y qué efecto tendrán en el uso diario.

1) Tamaño de CACHE: Simplificando es cuanta memoria RAM reservamos para cada base de datos que abre Firebird, y cuanta más, mejor, claro. Esta memoria se mide en "páginas", y una página puede variar de tamaño, siendo el tamaño por defecto en los datos de HCLab de 4 KB. Por defecto el modo "Superserver" de Firebird -la instalación por defecto- usa DefaultDbCachePages = 2048, es decir, unas 8 MB, así que realmente aquí tenemos mucho que ganar poniendo, por ejemplo, 10 veces más (80 MB) o incluso bastante más. Conviene jugar con ciertos valores y ver como queda el S.O. (Windows o Linux) de memoria libre, ya que gastar toda la memoria libre, aunque sea para una causa noble, nunca nos trae cuenta. Como el uso normal de HCLab es sobre una única base de datos, si decidimos usar 1 GB para Firebird, usaríamos unas cien veces más que lo que viene por defecto, unas 200 mil páginas, asi que DefaultDbCachePages = 204800 podría ser un valor interesante a probar. En el caso de ClassicServer, un buen número podría estar en torno a los 512 (512 x 4 = 2 MB por usuario simultaneo).

2) Ficheros temporales: Se usan al ordenar datos y cosas así, es decir, muy a menudo. Estos ficheros no deben estar en el mismo disco donde está la propia base de datos, ya que en ese caso, el manipular ambos ficheros a la vez penalizará la velocidad considerablemente. Busque en el fichero de configuración la cláusula TempDirectories y haga que apunte al disco del sistema operativo, o si lo tiene, a un disco extra que use solo para estos ficheros temporales.

3) Disco RAM: Si tiene RAM de sobra, digamos que mas de 2 GB en sistemas de 32 bits o más de 4 en sistemas de 64 bits, la RAM extra puede usarse para dar más velocidad a los temporales -sí, de nuevo son la clave- usando un "disco RAM". Estos discos, son programas que se instalan y hacen creer al sistema que la memoria es un disco nuevo, pero muy rápido. Si instalamos uno de estos programas y usamos esa unidad como directorio temporal de Firebird, el cambio puede ser espectacular.

**Particionamiento de tablas**

No hay información en la documentación