



Rick Sargeant, 123RF

Seguimiento de estadísticas SQL con ExtSQL

EL MARCADOR

ExtSQL proporciona estadísticas de uso detallado para sistemas de bases de datos MySQL y PostgreSQL.

POR JOHN MURTARI

En vista de sus comunidades de desarrolladores, su sólido rendimiento y su fiable funcionamiento, se puede aseverar que MySQL y PostgreSQL se han convertido en las bases de datos de código abierto por excelencia. A pesar de la potencia y flexibilidad de estas variantes SQL, las opciones de obtención de datos históricos sobre el uso de este tipo de bases de datos son relativamente limitadas. Las herramientas de monitorización SQL suelen centrarse en una monitorización tosca a nivel de servidor. El proyecto ExtSQL (*Extended Usage Statistics for SQL*) supone un esfuerzo por poner a disposición del administrador de bases de datos un rango de datos estadísticos mucho mayor a través de comandos simples disponibles desde el intérprete de SQL. De hecho, ExtSQL proporciona características de administración avanzadas para MySQL y PostgreSQL que ni siquiera se encuentran disponibles en los “grandes” proveedores de SQL. El administrador de bases de datos tiene a su disposición ahora ExtSQL para monitorizar y tener conocimiento tanto del uso del servidor como de las bases de datos individuales, los usuarios o incluso los clientes conectados.

Se puede guardar registro de casi cualquier variable mostrada por el comando `SHOW STATUS`. Por ejemplo, para ver la actividad

de todos los usuarios por horas de las últimas veinticuatro horas, mostrando los números de consultas agrupadas por tipos, ejecutamos:

```
SHOW STATISTICS ↗
Com_select, Com_update, ↗
Com_insert FROM user WHERE ↗
Com_select > 100 HISTORY ↗
LIMIT 24
```

La empresa Software Workshop Inc. desarrolló ExtSQL. Actualmente el producto está disponible en forma de software de código abierto bajo licencia GPL. Nuestra intención es que las características implementadas a través de ExtSQL sean implementadas por otras bases de datos como Oracle o DB2 y pasen a formar parte del estándar SQL. Mientras tanto, ExtSQL está disponible en forma de reemplazo para los demonios de servidor usados por MySQL y PostgreSQL.

En este artículo describimos cómo instalar y configurar ExtSQL en una instalación de MySQL típica. El procedimiento para la instalación de ExtSQL en PostgreSQL es similar.

Crear un mysqld Nuevo

El sitio web de ExtSQL [1] ha compilado binarios para varias plataformas de hard-

ware y sistemas operativos disponibles para evaluación y licenciamiento. También están disponibles allí las instrucciones de instalación y uso de ExtSQL más actualizadas.

Primero, descargamos los parches ExtSQL correspondientes a la versión de nuestro servidor de MySQL y colocamos el archivo `.gz` en el directorio padre de nuestra distribución de MySQL. Ejecutamos `make distclean` desde ese directorio para eliminar las compilaciones de productos anteriores. Y luego, ejecutamos

```
tar -zxvf ↗
extsql-my-src-5.0.45-rhel14-↗
x86-2-1a-2.0b.tar.gz
```

para descomprimirlos. Como resultado, obtendremos unos archivos similares a estos:

Tabla 1: db – Una Base de Datos

Clase	Entidad
user	Un usuario
host	La máquina de la conexión
conuser	Combo usuario-máquina de la conexión
condb	Combo db-máquina de la conexión
server	El servidor

```
patch-5.0.2.0b
compat-5.0.1a
do_conf-5.2
README.source_extsql
```

Los dos primeros son parches. El archivo *patch-* contiene la mayor parte de los cambios llevados a cabo por ExtSQL. El segundo archivo, *compat-*, los arreglos específicos para la versión particular de MySQL.

El archivo *do_conf* contiene los comandos de configuración exactos usados para la compilación de los fuentes, por lo que combinaremos esta lista con las selecciones personalizadas que hayamos escogido. Como de costumbre, se debe consultar el archivo README para estar al tanto de la información más reciente.

El siguiente paso consiste en aplicar los parches, pero comenzando por el archivo *compat-*:

```
patch -V t -p0 -iNu >
< compat-5.0.1a
patch -V t -p0 -iNu >
< patch-5.0.2.0b
```

Luego se debe ejecutar el comando *configure* con las mismas opciones que se usaron para compilar la instalación de MySQL previa a ExtSQL, sumando cualesquiera opciones que se vayan a añadir a la nueva versión parcheada.

Para ExtSQL 5.0.x, se debe añadir la siguiente opción al listado actual de *CXXFLAGS* utilizado en la definición del *configure* (como por ejemplo, *-DEXTSQL_50=1*):

```
'CXXFLAGS=-O2 -DEXTSQL_50=1
-pipe -m32 -march=i386
-mtune=pentium4'.
```

Seguidamente viene el *make* habitual con el que se compilará *mysqld*. El resultado será un nuevo binario de *mysqld* en el directorio *build*.

Instalación del Servidor ExtSQL

En vez de hacer *make install*, la forma de instalar la nueva versión es sustituyendo manualmente *mysqld*. Aquí recomendamos el siguiente procedimiento, suponiendo que el directorio base de la instalación de MySQL sea */usr/local/mysql*.

Comenzaremos realizando una copia de seguridad de todas las bases de datos, así como de *mysqld*:

```
cp -a mysqld mysqld.sav
```

ExtSQL guarda información de su estado en el archivo de errores de MySQL. Desde otra ventana, monitorizamos el servidor con el siguiente comando (ajustando la ruta correcta según sea el sistema):

```
tail -f /usr/local/mysql/var/
nuestro-hostname.err &
```

Y ya podemos instalar y activar el nuevo servidor mediante los comandos básicos disponibles en MySQL para parar e iniciar el servidor. Si la instalación usase otros wrappers, la forma de proceder con ellos sería la misma:

```
/usr/local/mysql/bin>
/mysqladmin -uroot >
-p"password" shutdown
060612 07:58:08 [Note] >
/usr/local/mysql/bin/mysqld: >
Shutdown complete
```

Nota: La ruta al archivo de registro que aparece con el comando *shutdown* debería ser la misma que la mostrada con la copia de seguridad de *mysqld*.

Copiamos el nuevo *mysqld* a su sitio con

```
cp -a /ruta/a/nuestra/
instalación/de/sql/
mysqld mysqld
```

e iniciamos el nuevo servidor:

Listado 1: Ejemplos de Uso

En los siguientes ejemplos se muestran comandos ExtSQL y su salida.

```
## Mostrar el número de selects e inserts de todos los usuarios:
sql> SHOW STATISTICS Com_select, Com_insert, Questions FROM user;
+-----+-----+-----+-----+
| user      | Com_select | Com_insert | Question |
+-----+-----+-----+-----+
| bandala   | 8302675   | 95973      | 23153940 |
| sandymao  | 1702812   | 6205       | 3829023  |
| ponnetli  | 24909     | 4784       | 95646    |
## Mostrar el número de selects, consultas totales y updates hechos
## desde todos los hosts cliente:
sql> SHOW STATISTICS Com_select, Questions, Com_update FROM host;
+-----+-----+-----+-----+
| host      | Com_select | Questions | Com_update |
+-----+-----+-----+-----+
| db2.adomain.com | 17715223 | 44224076 | 4143634981 |
| lathe.adomain.com | 2738061 | 9743215 | 3913397495 |
| telkomadsl.co.za | 195 | 5390 | 539604 |
## Mostrar número de selects, consultas totales y updates hechos por
## la máquina users@client con más de 10000 consultas:
sql> SHOW STATISTICS Com_select, Questions, Com_update FROM conuser
WHERE Questions > 10000;
+-----+-----+-----+-----+
| conuser   (user@host) | Com_select | Questions | Com_update |
+-----+-----+-----+-----+
| bandala@db2.adomain.com | 8306726 | 23163320 | 3439850933 |
| sandymao@db2.adomain.com | 1704040 | 3831803 | 3365501841 |
| ponnetli@taro.adomain.com | 24920 | 95662 | 156529077 |
## Mostrar el número de selects, consultas totales y updates en la base
## de datos allison de los tres últimos minutos de actividad:
sql> SHOW STATISTICS Com_select, Questions, Com_update FROM db
LIKE 'allison' HISTORY LIMIT 3;
+-----+-----+-----+-----+
| db | minutes | Com_select | Questions | Com_update |
+-----+-----+-----+-----+
| allison | 11/20 13:56 | 216 | 382 | 318343 |
| allison | 11/20 13:55 | 642 | 1618 | 1386347 |
| allison | 11/20 13:54 | 280 | 699 | 646855 |
```

```

/usr/local/mysql/bin
/mysql_safe
-user=$usuario_mysql
$otros_args &
071121 7:42:47 [Note]
ExtSQL build: ExtSQL version:
extsql-my-bin-5.0.45-
rhel4-x86-2-1a-2.0b
060612 7:58:38 [Note]
ExtSQL disabled by user,
no extsql_class_list defined

```

Las líneas anteriores nos confirman que el servidor ha sido compilado con el subsistema ExtSQL, pero está deshabilitado actualmente debido a que aún no se han solicitado estadísticas. Recomendamos ejecutar el servidor durante al menos un día para comprobar su buen funcionamiento. Su rendimiento no debería verse afectado.

Si ocurriese algún error, éste podría indicar un problema con la compilación. Para volver a un punto anterior hasta una configuración

buena conocida, bastaría con restaurar la versión de *mysqld* y reiniciar el servidor. La recopilación de datos estadísticos se hará editando */etc/my.cnf* para añadir la siguiente línea, que ha de estar en la sección *[mysqld]* del archivo de configuración:

```

extsql_class_list=
"user, max-100, time-120,
units-h, (Com_insert,
Com_select, Com_update,
Com_delete, Com_replace,
Qcache_hits, Questions,
Slow_queries)"

```

El comando anterior debe ser una sola línea física en el archivo. En caso de cambiar */etc/my.cnf*, hay que asegurarse de eliminar esta línea si se revierte a un *mysqld* estándar.

Al reiniciarlo, el servidor debe mostrar una salida como ésta en el registro de errores:

```

060612 8:26:12 [Note]
ExtSQL build: ExtSQL version:
extsql-my-bin-5.0.45-rhel4-
x86-2-5a-3.0b
060612 8:26:12 [Note]
ExtSQL ACTIVE tracking memory:
435600 bytes for 1 class

```

Cualquier error producido durante la inicialización o el funcionamiento, se escribirá en el registro, deshabilitándose a sí mismo el subsistema y dejando intactas las operaciones de MySQL.

Terminología

En ExtSQL, las entidades que interactúan con el servidor se llaman clases (*class*); ver Tabla 1. Dentro de una clase, las instanciaciones específicas se denominan instancias (*instance*). Instancias de usuario podrían ser *pedro* o *cristina*, por ejemplo. La clase *server* es única en tanto que se trata del propio servidor.

Por cada instancia se puede hacer un seguimiento de clasificaciones de datos individuales llamadas *vars*. Las *vars* incluyen casi todos los valores mostrados por el actual comando *SHOW STATUS*. Se usará el nombre de la variable tal y como la muestra *SHOW STATUS*; por ejemplo, *Com_select* rastreará consultas de tipo *select*.

Configuración de las Opciones

Hay más de 100 vars (clasificaciones de datos) disponibles. Configuramos el servidor ExtSQL describiendo las variables en las que estamos interesados para la recopilación de datos. Para ilustrarlo con un ejemplo más completo, consideremos esta cadena, que debe ser una línea en el archivo de configuración:

```

extsql_class_list=
user, max-50, time-60,
units-m, (Com_select,
Com_update), db, max-50,
time-10, units-h,
(Com_insert, Com_select,
Com_update, Qcache_hits),
host, max-5, time-3, units-d,
(Com_select, Com_update,
Slow_queries), server, max-5,
time-3, units-d,
(Questions, Qcache_hits)"

```

Del mismo modo que con *SHOW STATUS* (en el caso de datos históricos), el servidor rastreará la información de cuatro clases dife-

Listado 2: INFORMATION_SCHEMA

```

mysql> use INFORMATION_SCHEMA;
mysql> SHOW TABLES;
  ## Nota: las tablas adicionales configuradas por el DBA para
  ## ExtSQL llevan el prefijo 'EXTSTATS_'
| EXTSTATS_condb          |
| EXTSTATS_conuser       |
| EXTSTATS_db             |
| EXTSTATS_host          |
| EXTSTATS_server        |
| EXTSTATS_user          |
  ## Fin de las tablas añadidas
mysql> SELECT user, minutes, Com_select, Questions FROM EXTSTATS_user
      WHERE TIMEDIFF(now(), minutes) < '10:00';
+-----+-----+-----+-----+
| user   | minutes           | Com_select | Questions |
+-----+-----+-----+-----+
| thebook | 2008-07-19 09:28:00 | 2         | 4         |
| thebook | 2008-07-19 09:27:00 | 3         | 3         |
| matt   | 2008-07-19 09:53:00 | 4         | 4         |
| matt   | 2008-07-19 09:52:00 | 0         | 1         |
  ## Con esto se consigue que toda la sintaxis del analizador esté
  ## disponible, PERO... demuestra que los datos se guardan en un
  ## búfer circular en intervalos de tiempo (minutos en este ejemplo),
  ## con el tiempo '0000-00-00' conteniendo los totales desde el
  ## comienzo del servidor.
mysql> SELECT user, minutes, Com_select, Questions FROM EXTSTATS_user
      WHERE user='thebook' LIMIT 5;
+-----+-----+-----+-----+
| user   | minutes           | Com_select | Questions |
+-----+-----+-----+-----+
| thebook | 2008-07-19 09:27:00 | 3         | 3         |
| thebook | 2008-07-19 09:25:00 | 2         | 73        |
| thebook | 2008-07-18 16:44:00 | 0         | 14        |

```

rentes: *user*, *db*, *host*, y estadísticas acumulativas del servidor.

Con cada clase, *max*- antecede al número máximo de instancias de esa clase. El prefijo *time*- lo llevan las clases para mostrar el número de unidades de tiempo históricas a almacenar. Mientras, *units*- designa el intervalo deseado: *m*, para minutos; *h*, para horas; y *d*, para días. Después de los límites para cada clase se incluye un listado de *vars* separados por comas.

La línea que comienza por *db* es una directiva para el rastreo de hasta 50 bases de datos y la grabación de los datos históricos de las últimas 10 horas. Las *vars* fijadas en el anterior comando son inserts, selects, updates y coincidencias en la caché de las peticiones.

Sintaxis de SHOW STATISTICS

El objetivo, en cuanto a la sintaxis del comando, era el de seguir lo más fielmente posible el estándar SQL. La sintaxis descrita aquí debería resultar familiar, sobre todo si cambiamos *SHOW STATISTICS* por *SELECT*. En el Listado 1 se ilustran algunos ejemplos de uso de ExtSQL.

Algo más diferentes son *LIKE*, para buscar instancias específicas, y la nueva palabra clave *HISTORY*, para sacar históricos. Con *LIMIT* hemos de operar de manera ligeramente diferente si se especifica *HISTORY*. Sin *HISTORY* se limita el número de resultados mostrados. Con *HISTORY*, su uso es el de un limitador de tiempo.

```
SHOW STATISTICS *
( * | Var list) FROM Class
[LIKE 'Instance pattern']
[WHERE Var ( '<' | '>' | '=' )
num] [ORDER BY Var] [HISTORY]
[LIMIT num]
```

Por ejemplo, podemos pedirle:

```
SHOW STATISTICS *
FROM user LIKE '%joe%'
WHERE Com_select > 500
```

Sintaxis para Information Schema

Muchos sabrán que INFORMATION SCHEMA forma ya parte del estándar SQL, así como que su propósito es el de hacer de las bases de datos unas entidades más “auto descriptivas”. El proyecto ExtSQL dispone ya de la implementación de un ejemplo funcional para MySQL 5.0.x (ver Listado 2).

Tabla 2: Otros Comandos de Administración

Instrucción	Acción
STATISTICS (ON OFF)	Poner la función de recolección de datos a <i>on</i> o <i>off</i> . Los datos históricos no se pierden.
STATISTICS RESET [conf_file]	Borrar los datos anteriores. Comenzar a recopilar usando las definiciones de clase existentes, o usarlas en <i>conf_file</i> .
STATISTICS WRITE reload_file	Guardar en un archivo una captura de los datos actuales. Útil para guardar los datos de un evento anómalo para su posterior análisis.
STATISTICS READ reload_file	Cargar datos estadísticos desde el archivo especificado.

En SQL estándar, sería:

```
SHOW STATISTICS *
FROM user WHERE user
LIKE '%joe%' AND
Com_select > 500
```

LIKE es una cláusula aparte en la sintaxis que coincide con instancias de tipo *Class*. Por ahora, la sintaxis está limitada a la cláusula *WHERE*, y *ORDER BY* sólo está disponible si *INFORMATION SCHEMA* está soportado por la versión base del servidor ExtSQL.

Rendimiento del Servidor

El rendimiento siempre ha sido uno de los objetivos principales de ExtSQL. Al haber cada vez más memoria disponible en los servidores, lo más lógico es guardar los datos en la RAM y dar al administrador la posibilidad de controlar absolutamente la cantidad de memoria que puede utilizar. De este modo, el rendimiento se ha mejorado en aproximadamente un 5%, un porcentaje nada desdeñable en la mayoría de los servidores.

Para comprender mejor las estadísticas sobre rendimiento, analicemos cómo almacena y reporta el tiempo ExtSQL. Con la definición de clase *user*, *max-50*, *time-10*, *units-h*, ExtSQL creará un búfer con capacidad para 11 períodos de tiempo. El búfer 0 alberga siempre números acumulativos desde el inicio del servidor, y es el número que se muestra al ejecutar el comando *SHOW STATISTICS * FROM user*.

La actividad histórica se captura en períodos de 10 horas de modo circular y sólo cuando se produce actividad. Por ejemplo, si arrancamos el servidor y 20 horas después introducimos

```
SHOW STATISTICS *
FROM user HISTORY
```

se guarda la actividad del usuario en cada una de las 10 primeras horas de fun-

cionamiento del servidor, y luego sólo durante las últimas 5 horas (es decir, un lapso de 5 horas sin actividad alguna). La actividad ocurrida entre las horas 5 y 10 se reporta hora por hora, y luego se reportan las últimas cinco. Cada línea lleva la marca de tiempo con su hora. Haciendo uso del comando *SHOW STATISTICS * FROM user*, deberíamos ver una única línea con los totales acumulados para las 20 horas.

Seguridad

Puede que no convenga compartir la información proporcionada por ExtSQL con todos los usuarios. El archivo de configuración nos permite especificar a qué usuarios se les permitirá acceder a ella:

```
extsql_users=
“juan,pedro,felipe,crisrina”
```

Por supuesto, a *root* siempre se le permite el acceso.

Controlar ExtSQL

Hay otros comandos (Tabla 2) que permiten al administrador de bases de datos suspender temporalmente la recopilación de datos, su almacenamiento, o incluso cambiar qué datos se han de recopilar, sin tener que reiniciar el servidor.

La opción de configuración llamada *extsql_reload_file* nos permite escribir a archivo los datos recopilados, que se guardarán y restaurarán automáticamente en los sucesivos reinicios del servidor.

Tendencias para el Futuro

El desarrollo con ExtSQL debería ser sólido. Planeamos proporcionar un punto centralizado para el debate comunitario, los reportes, soluciones y proposiciones de mejoras de errores.

RECURSOS

[1] ExtSQL: <http://www.ExtSQL.com/>