Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

De como sobrevivimos, más mal que bien, al efecto Slashdot (14612 lectures)

Per **Ricardo Galli Granada**, ***gallir*** (http://mnm.uib.es/gallir/)

Creado el 11/05/2001 01:08 modificado el 11/05/2001 01:08

> Si, un atentado sin aviso[1] previo :(nos dejó el Postgres hecho polvo y tuve que hacer virguerías para poder servir páginas, mas lentamente pero al menos sin errores. De ellos aprendí una cosa importante, nunca te fíes pensando "total tengo poco tráfico[2]" y menos aún si se te ocurre envíar un mensaje[3] a la lista linux-kernel.

[Actualización, 2:27 hrs]: Que bien parece a una peli ;-). Bueno, son las 2 y pico, hice unas modificaciones a los scripts para reducir los *select count()* y algunos otros redundantes. También agregué un control del valor retornado por el *pg_pconnect* y ya activé los scripts nuevos.

Ahora está todo normal, le puse a Guillem el contador más o menos con los accesos que hubo en estar horas, aunque seguramente hubiera sido mucho mayor si no hubiese habido semejante tormenta.

Aunque siguen habiendo muchos accesos (unos pocos por segundo), ya se comporta normal y la carga del sistema está bastante bien (entre 0.01 y 0.10).

Sipe, se me ocurrió enviar esta madrugada (por el día 10 de mayo) un mensaje a la lista de linux-kernel para que miren las pequeñas pruebas[4] hechas por Guillem. La sorpresa comenzó a los 10 minutos de haberlo enviado cuando empezamos a recibir bastante tráfico, pero muy soportable, no más de una visita cada pocos segundos.

Pero ocurrió lo peor, alguien envió el enlace a Timothy[5] de Slashdot[6] y nos sacaron en sus páginas. Según los *logs*, todo comenzó a las 8 de la noche, cuando yo estaba dando clases de Sistemas Operativos. A las 9:30 termino, voy a mi despacho a leer un poco las noticias de Bulma (y controlar el servidor, sobre todo porque estoy haciendo pruebas[7] del APC) y me encuentro con la desagradable sorpresa de los mensajes del PHP: *no puedo conectarme a Postgres, número de clientes excedidos...*

Vaya faena, para mantener funcionando tuve **reduje los clientes máximos** del Apache y **subí los del Postgres**. No puede poner más de **32 al PG porque es el límite que viene en el precompilado** (si señor, nunca instales binarios...;-). en RPM del sitio de Postgres.

Al final tuve que ponerme a editar los *scripts* de Bulma para **reducir al máximo los queries y eliminar los *updates*** que cuentan la cantidad de lecturas de cada artículo. Con esto mejoró muchísimo, pero aún así daba errores a veces, así que me di cuenta:

- No controlábamos que la conexión a la base de datos haya sido satisfactoria (en el pg_pconnect) por lo que el PHP daba montón de errores al intentar seguir ejecutando el script. Esto es un BUG como una casa... La solución fue ponerle un bucle que lo intentaba 10 veces con una espera de unas pocas décimas cada vez. Con ello retrasaba la respuesta pero al menos no daba tantos errores.
- En el *php.ini* estaba habilitada la salida de errores. Aunque está comentado que para servidores en producción se deshabilite la salida, lo teníamos habilitado porque lo usaba para el debug de desarrollo de los programas. Moraleja: intenta tenerlo siempre deshabilitado, nunca se sabe si aparecerás en Slashdot.
- El Postgres es muy lento para responder a consultas que involucran el ordenamiento de la salida. Y no hay forma de optimizarlo a niveles razonables.

Al final pude volver a casa a las 11:15 de la noche, y mientras Guillem disfrutaba de la fama, yo me comía el rapapolvo de mi mujer...

RESUMEN: tuvimos unas **12000** visitas y **100.000** *hits* en **2** horas. El Linux (2.4.4) no tuvo ningún problema, la carga (*loadavg*[8]) nunca superó **1.5** en las peores horas y **0.7** más o menos cuando ajusté los parámetros. Además, mientras pasaba esto, estaba enviando los mensajes de la lista de correo y yo estaba editando los *scripts* y leyendo mi correo. ¿Que hubiese pasado con un Windows y ASP sobre la misma máquina?

Al fin y al cabo fue toda una experiencia.

--ricardo

Mas leña...

> > Como se nota que nunca has sido slashdotted con el puto postgres... ;-)
>
> Pero si el Postgres es una caña. (mientras no lo ataques desde
> Access).
>
> Es que ese error que saca es por alcanzar el máximo de cliente. Con
> subirlo ya esta. Y si al arrancar no lo defines el máximo se pone en 32.

Como se nota que no leen mis artículos y los comentarios de los mismos...;-)

Un tío, *bishi*, me estuvo pasando unos parámetros bastantes buenos.

No todo se soluciona aumentando servidores. Sino podríamos coger un PIII con 4GB de RAM y servir 4 millones de páginas por día sencillamente poniendo a 5000 el número de procesos de Postgres. Y todos sabemos que no es así....

El RPM que viene con el RPM de postgresql.org tiene un máximo "insuperable" de 32 servidores (ver explicación[9]). Pero este no era el problema principal, sino que el Postgres no podía servir más de 10-15 peticiones por segundo (como explico en los tests de aqui[10]).

O sea, pon los clientes que sea, pero no se superaban las 15 páginas por segundo, con un tiempo de retorno mínimo (sin concurrencia) de 0.08 segundos (para una página de artículo) pero que podía llegar fácilmente (como lo hizo) a más de 3 segundos cuando estaban los 30 servidores.

Mucha culpa tenías los SELECT complejos que hacíamos para el índice de la derecha, sobre todo por el "ORDER BY ... DESC" de la fecha de modificación, que fue lo primero que quité cuando estábamos /.ed.

Luego también saqué los dos a tres SELECT COUNT que hacíamos (verás que ya no aparece el total de artículo) y así pude reducir bastante el tiempo de retorno en el PHP.

Puse también 2048 buffers de memoria compartidda, pero no solucionaba nada y los bajé a 512 para tener más memoria para el Apache (llegamos a tener más de 400 procesos) y buffer/cache (que beneficia mucho al postgres).

Y para finalizar estuvimos todo el tiempo con mi versión modificada[7] del APC (disminuye la "contención" de semáforos) con un algoritmo de lectores/escritores con tres semáforos (que me costó un par de días pensarlo[11]) donde sólo los escritores (menos del 0.01%) mantienen un "lock" global para optimizar el PHP4.

En resumen, para poder servir al menos parcialmente tuve que modificar la cantidad de clientes del Apache (para reducir os bloqueos en la base de datos), modificar el php.ini para poner a tope las conexiones persistentes y dejarme un par de *backends* libres para hacer *vacuumdb* cada hora, modificar bulma.php3 y recordset.php3 para disminuir los sql complejos, cambiar los parámetros del buffer de Postgres teniendo en cuenta la memoria, porque me empezó hacer *swapping* y tenía que evitarlo a toda costa.

En resumen, bastante complejo y que sólo lo entenderías si lo hubieses vivido en directo y haber visto que si tocabas una parámetro, solucionabas una cosa y la cagabas por otra.

Ah... si la vida fuese _tan_ sencilla como hacer un

```
# postmaster -B 100000 -S 100000 -N 1000
```

hubiera podido dormir tranquilo anoche ;-)

Lista de enlaces de este artículo:

1. http://slashdot.org/article.pl?sid=01/05/10/1747213&mode=flat
2. http://bulma.net/stats/usage_200105.html
3. http://www.uwsg.indiana.edu/hypermail/linux/kernel/0105.1/0358.html
4. http://bulma.net/body.phtml?nIdNoticia=626
5. http://www.monkey.org/~timothy/
6. http://slashdot.org/
7. http://m3d.uib.es/~gallir/ext/APC/
8. http://bulma.net/body.phtml?nIdNoticia=550
9. http://bulma.net/body.phtml?nIdNoticia=632
10. http://bulma.net/body.phtml?nIdNoticia=625
11. http://m3d.uib.es/~gallir/ext/APC/apc/apc_rwlock.c

E-mail del autor: gallir _ARROBA_ uib.es