

## 25/05/09 - Escalado de Ruby on Rails

Después de liberar el primer proyecto escrito en Ruby on Rails, cuando lo pasamos a producción, nos dimos cuenta de que el sistema funcionaba realmente lento. En algunos casos, incluso, no respondía, con lo que buscamos información por internet y vimos:

### Mongrel y Thin no son multi-hilo

Fue algo que nos sorprendió mucho, una aplicación servidora secuencial, que atiende las peticiones una a una y, en caso de que una petición se tarde un par de segundos (que las teníamos :-S ), el sistema se queda trabado durante todo ese tiempo.

Las soluciones posibles eran dos:

- Phusion Passenger, también conocido como **mod\_rails**, es un módulo para Apache que mantiene tantos hilos como peticiones se vayan gestionando, siempre con unos límites máximos y mínimos, tal y como apache suele hacerlo.
- **Nginx + Thin**, esta solución fue un poco más *artesana*, por decirlo de alguna forma, ya que consistía en configurar este servidor web como proxy inverso, para hacer el balanceo de peticiones entre todos los hilos de thin que se quieran cargar en el sistema. Descartamos en este punto a mongrel por comentarios que había leído en otros blogs.

Bien, por correr lo más posible, lanzamos tantos hilos como nos fue posible en la máquina que dedicamos al proyecto y, nos complació ver que podía aguantar hasta 40 hilos, sin que sus CPUs lo notasen apenas. Con lo que conseguimos un pool de 40 puertos dedicados a atender peticiones HTTP.

En otra máquina se configuró un proxy inverso. En el escenario de pruebas usamos nginx, aunque el tenerlo en la misma máquina y hacer pruebas masivas, nos dio como resultado que una cantidad importante de peticiones se perdían o transformaban en respuestas de tipo 500 por parte de nginx. Optamos entonces por un balanceador que tiene la compañía vía hardware, muy rápido y, a día de hoy el sistema web tira tan rápido y casi mejor que el que tenemos montado en PHP

En el próximo artículo comentaré las configuraciones, arquitecturas y demás

seguidas, sobre todo a nivel de nginx y thin, que es lo que realmente interesa.

**Entradas relacionadas** sobre Erlang