Propuesta de Oruga Amarilla

Carlos Thompson

Oruga Amarilla, una marca Invermeq e Interlecto

Diciembre de 2013

Resumen

Plan de trabajo para lograr mejoras substanciales en la página *http://test.orugaamarilla.com* de Oruga Amarilla, la cual, en la actualidad, se cuelga frente a algunas consultas.

Este plan de trabajo incluye optimizar la base de datos, crear sistemas de control de procesos en la base de datos y sugerir algunos pasos para una solución más permanente.

Contenido

Resumen I

1. El problema actual 1
- 1.1. Observación . . . 1
- 1.2. Diagnóstico . . . 1
- 1.3. Posibles soluciones . . . 2

2. Optimización de la base de datos 3
- 2.1. Situación actual . . . 3
- 2.2. Sugerencia de optimización . . . 3

3. Optimización en el diseño de la aplicación Web 6

4. Sistemas de control de procesos 7

5. Sugerencias 8

Capítulo 1

El problema actual

Sección 1.1. Observación

Cuando se intentan cargar páginas que muestran los resultados de algunos sitios, principalmente el sitio San Pedro de Elite Flowers la página tarda un tiempo largo sin presentar ningún tipo de actividad tras el cual aparece un error 500 Internal Server Error.

Consultas subsiguientes, incluyendo aquellas que no pidan información de San Pedro, fallan igualmente.

Sección 1.2. Diagnóstico

Las consultas por últimas actividades de algunos de los sitios, y en particular de San Pedro, ejecutan lentamente en el servidor de bases de datos MySQL *mysql.orugaamarilla.com* .

Estas consultas tardan significativamente más que el tiempo configurado por el servidor web para presentar resultados. Como el programa no termina de ejecutar en el tiempo estipulado, el servidor emite un error 500 Internal Server Error y detiene la ejecución del script pero la consulta en el servidor de base de datos sigue ejecutando.

Las subsiguientes consultas, incluyendo aquellas que son generadas por el usuario cuando ejecuta F5 o cambia manualmente el URL de la aplicación frente a la frustración por no recibir información. Este comportamiento justificado del usuario agrava la situación pues encola nuevas consultas.

Este bloqueo en el servidor de bases de datos podría estar causando también pérdidas en la información que la base de datos recibe automáticamente de los sitios. Esto último no ha sido verificado, pero es una conclusión razonable.

Sección 1.3. Posibles soluciones

Entre las soluciones planteadas se incluyen:

1. Optimizar la base de datos con el objeto de reducir el tamaño de las tablas y, en consecuencia, agilizar las consultas individuales.

   (ya se han adelantado pasos en este sentido pero no se han realizado pruebas.)

2. Agregar sistemas de limpieza de consultas abandonadas. Esto es llevar un control de las consultas que se inician pero que no terminan porque el script fue suspendido bien por *timeout* del servidor, bien por acción del usuario (detener carga de la página, relanzar la consulta, cambiar la consulta, etc.)

   (Se ha probado con la inclusión de conexiones persistentes, el cual incluye automáticamente un módulo de limpieza, pero no se han observado resultados positivos, igualmente esto sólo serviría para consultas relanzadas o cambio de consulta y no para otro tipo de abandonos.)

3. Agregar rutinas en el diseño de la aplicación web que permita la carga de la página antes de terminar la consulta y que el resultado de la consulta se agregue después, independientemente del tiempo que tarde la consulta.

   Esto tiene un efecto más psicológico pues el usuario no *ve* una página bloqueada sino que es informado que la consulta está tardando y evita la aparición de errores 500 Internal Server Error, disminuyendo por ende la sobrecarga del servidor por parte de usuarios frustrados.

   (La técnica es conocida a través de AJAX, pero aun no se ha buscado cómo incluirla en la página de Oruga Amarilla.)

4. Utilizar un servidor dedicado.

   Varios de los problemas de desempeño son causados por el hecho de que los servidores actuales de Oruga Amarilla son un servidor web compartido y un servidor MySQL compartido.

   Además del desempeño, no se tiene acceso de súper-usuario al servidor lo que impide hacer algunas maniobras de corrección rápidas.

Capítulo 2

Optimización de la base de datos

Aquí se tratarán las estrategias de optimización de la base de datos de Oruga Amarilla.

Sección 2.1. Situación actual

Actualmente la aplicación de Oruga Amarilla utiliza dos tablas en la base de datos `orugadata` en el servidor *mysql.orugaamarilla.com*. Estas tablas son: `dl_status_i` y `dl_status_p`. Actualmente estas tablas poseen respectivamente más de 3,4 y más de 22 millones de registros, ocupando respectivamente más de 400 MB y 1,5 GB.

Toda consulta sobre estados debe hacer un cruce entre estas dos tablas. Desplegar información además requiere consultar otras tablas en las que se almacena la información *no cambiante*.

El tiempo de proceso que se requiere para hacer el cruce de estas tablas, debido al tamaño, es grande. Ya hay ciertas optimizaciones al filtrar los resultados pero esto requiere un ordenamiento el cual también tarda.

Sólo San Pedro compone () registros en `dl_status_i` y un número proporcional de datos en `dl_status_p`.

Sección 2.2. Sugerencia de optimización

En lugar de dos tablas que incluyan todas las estaciones en una consulta cruzada se sugiere crear cuatro o cinco tablas por cada estación, evitando también las consultas cruzadas.

Cada estación tendrá una tabla de datos activos, en los cuales se registren en una sola tabla todas las actualizaciones de los últimos 60 días. (Podrían ser dos tablas, una para estado de instrumentos y otra para alarmas.)

Adicionalmente habría tres tablas de resultados históricos: una tabla con todos los datos anteriores a 60 días y dos tablas con resultados promediados. De estas tablas se eliminarían los datos que se sabe que no son relevantes.

Ventajas.

- Las tablas activas (últimos 60 días y promedios) serán más pequeñas y requerirán menos recursos y menos tiempo en ser consultadas.
- Ninguna de estas tablas requerirá cruce de tablas para hacer las consultas pertinentes lo que reduce el procesamiento en el servidor con la consiguiente ganancia en tiempo.
- La mayor parte de los datos históricos que se requieren se requieren promediados. Las tablas de resultados promediados agilizan estas consultas en la mayor parte de los casos.
- Se conservan los datos históricos relevantes.
- No se guarda información poco relevante.

Desventajas.

- Aumenta la complejidad de las relaciones en la base de datos.
- Cada nuevo sitio implica la creación de nuevas tablas.
- Se duplica información.
- Cierta información se perdería.
- Migrar requerirá de consultas largas.
- Requiere cambios grandes en la aplicación web para responder al nuevo diseño.
- El rediseño planteado podría no ser el óptimo

Ninguna de estas desventajas es crítica. Las desventajas más críticas afectan sobre todo la fase de pruebas y migración las cuales, frente a la situación actual, no representan una desventaja adicional.

Conclusión. Este u otro rediseño de la base de datos **debe** hacerse.

Aun si el rediseño aquí propuesto no es el óptimo debe ser una mejora significativa frente a la situación actual.

Capítulo 3

Optimización en el diseño de la aplicación Web

ORUGA AMARILLA Capítulo 4

Sistemas de control de procesos

ORUGA AMARILLA Capítulo 5

Sugerencias