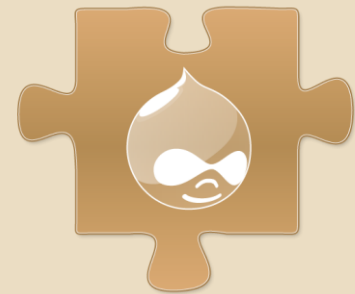


Drupal Summit Latino – Lima 2011

Drupal Performance para usuarios no autenticados.

Implementación y gestión

Karim Boudjema
Drupal Developer



WEB
en
STOCK
www.webenstock.com

La Comunidad Drupal

- ❑ Miles de cabezas son más potentes que una
- ❑ Muchas de la funcionalidades que buscamos ya existen! NO reinventes la rueda!
- ❑ Hay desarrolladores Drupal que son genios! Aprovecha de su experiencia!



Performance, rendimiento Vs Escalabilidad

- Escalabilidad
 - ▣ Capacidad de servir un incremento de usuarios y de datos
- Rendimiento
 - ▣ Tiempo de respuesta del servidor + tiempo de carga de la página
- Nuestro enfoque hoy
 - ▣ Mejorar el tiempo de respuesta de nuestra aplicación Drupal para usuarios no autenticados.



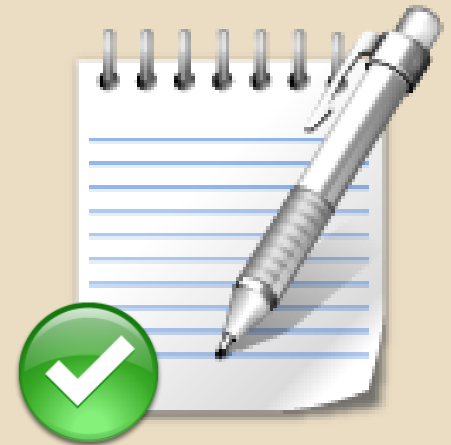
Lo que NO veremos hoy

- Front end performance
- Back end performance
 - ▣ Reverse proxy con Varnish
 - ▣ Apache mod_deflate
 - ▣ APC
 - ▣ Memcache
 - ▣ Cache router
 - ▣ Authcache
 - ▣ Query cache



Que vamos a ver entonces?

- ❑ Principio de caching
- ❑ Evaluar el rendimiento con AB y Devel
- ❑ Caching del core de Drupal
- ❑ Gestión del cache: Cache browser
- ❑ Gestión del cache: Content Refresh
- ❑ Rompemos con el caché: el módulo Boost



Principio de Caching

- Evitar repetir una misma operación guardando el resultado.
- Ej.: Calcular $2+3$,
 - ▣ escribir el resultado en un papel (cachear en la BD)
 - ▣ memorizar el resultado (cachear en memoria)



Principio de Caching - Definiciones

- En informática, una caché es un conjunto de datos duplicados de otros originales, con la propiedad de que los datos originales son costosos de acceder, normalmente en tiempo, respecto a la copia en la caché.
- Cuando se accede por primera vez a un dato, se hace una copia en el caché; los accesos siguientes se realizan a dicha copia, haciendo que el tiempo de acceso medio al dato sea menor.

Que cachear para usuarios no autenticados?

- **Páginas, páginas.... Y solo páginas...**
 - ▣ **Nos olvidamos de las views, los bloques ...**



Como evaluar nuestro rendimiento

- No podemos implementar políticas de rendimiento sin una evaluación.
- Evaluación con Apache Bench (AB)
 - ▣ Fácil, simple y todos lo tenemos instalado
- Evaluación con Devel
 - ▣ Más tedioso, no compatible con el cacheo agresivo de Drupal, útil si no tenemos acceso a la Shell del servidor



Apache Bench (AB)

- AB test para un usuario no autenticado
 - `ab -c 1 -n 100 http://example.dev/`
 - Donde
 - `-c` = concurrencia de los requerimientos
 - `-n` = total de requerimientos a la pagina
 - En este caso haremos 100 requerimientos a la página `http://example.dev/`
- Un solo indicador: 'Requests per second'
- Evaluaremos nuestras distintas políticas de caché con este indicador

Evaluación con Devel

- ❑ Descargar el módulo: <http://drupal.org/project/devel>
- ❑ Habilitar el componente: Performance Logging
- ❑ Configurar en `admin/settings/performance_logging`
 - ❑ Detailed logging: Enabled

Hagamos una primera evaluación

- Con AB
- Con Devel



El caché del Core de Drupal

- El sistema de caché de Drupal almacena sus datos en las siguiente tablas de la base de datos:

Por defecto

1. **cache** – Configuración des modules, estructuras de las tablas, configuration des themes utilizados.
2. **cache_menu** – registra una copia de los menus y de las URLs asociadas a estos menus
3. **cache_filter** – registra una copia del contenido de nuestros nodos filtrados
4. **cache_form** – registra una copia de los formularios del sitio

Configurable

1. **cache_page** – registra una copia completa de del código html de las paginas pero solo para usuarios no autenticados.
2. **cache_block** – registra una copia de los bloques

Habilitar el cache del core

- Vamos en **admin/settings/performance**
- **Caché de páginas**
 - ▣ Modo de caché: Normal o agresivo
 - ▣ Mínimo de permanencia en caché: 3 horas
 - ▣ Compresión de páginas: Activado
 - Para ver si nuestro server ya realiza la compresión:
http://www.whatsmyip.org/http_compression/
- **Caché de bloques**
 - ▣ No lo necesitamos para usuarios no autenticados

Habilitar el cache del core

- ❑ **Optimizaciones del ancho de banda**
 - ▣ Optimizar archivos CSS: Activado (solo en producción)
 - ▣ Optimizar archivos JavaScript: Activado (solo en prod.)
- ❑ Guardar la configuración
- ❑ Ahora podemos ver como la tabla cache_page empieza a llenarse
- ❑ Ahora volver a testear con AB y Devel para usuarios anónimos.

Gestión del cache: Cache Browser

□ Cache Browser

- http://drupal.org/project/cache_browser

- Necesitamos el módulo Format Number

 - http://drupal.org/project/format_number

- Permite resetear el caché de cada tabla o de un registro de una tabla. Así no tenemos que resetear todo cada vez.

- Permite ver el contenido de cada registros dentro de las tablas. Así podemos ver como una página es cacheada.

Gestión del cache: Content Refresh

□ Content Refresh

- http://drupal.org/project/content_refresh
- Escenario: Los usuarios anónimos pueden dejar comentarios y el cache del core está habilitado.
- Cuando se postea un nuevo comentario, el caché de esta página es eliminado y permite al usuario ver inmediatamente su comentario.
- Cuando se postea un nuevo nodo, el caché de la home page es eliminado
- Ver la configuración en <admin/content/content-refresh>

Gestión del cache: Cache Actions

□ Cache Actions

- http://drupal.org/project/cache_actions
- Depende del módulo Rules
(<http://drupal.org/project/rules>)
- Resetear el caché de
 - cada bin (o tabla)
 - los CSS/JS
 - las views
 - Los panels
- Compatible con Varnish

Boost: Adios al caché...



- Boost crea un archivo .html para todas las páginas visitadas por los usuarios *anónimos*.

- El próximo usuario anónimo leerá este archivo .html en vez de ir al index.php y la BD.

- Es un excelente caché por archivos estáticos que se ejecuta desde el .htaccess gracias a mod_rewrite.

Instalar Boost

□ Necesitamos tres módulos:

- El módulo Boost (por supuesto)

<http://drupal.org/project/boost>

- El módulo Transliteration

<http://drupal.org/project/Transliteration>

- El módulo Globalredirect

<http://drupal.org/project/Globalredirect>

Configurar Boost

- Configurar el módulo Boost
([admin/settings/performance/boost](#))
 - ▣ Se puede guardar las opciones por defecto. Solo desactivamos la opción 'Only allow ASCII characters in path'.

- Modificar el archivo .htaccess
 - ▣ Copiar el código ([admin/settings/performance/boost-rules](#)) en el archivo .htaccess entre
 - # RewriteBase /
 - # Rewrite URLs of the form 'x' to the form 'index.php?q=x'

- Desactivar al cache del core de Drupal
- Modificar el archivo robot.txt

En resumen

- La estrategia de rendimiento para usuarios no autenticados (o anónimos) se basa en:
 - ▣ Habilitar el caché de página del core de Drupal
 - ▣ Gestionar este cache con
 - Cache Browser (administración)
 - Content refresh (expiración)
 - ▣ Olvidarnos el caché del core y utilizar el módulo Boost

