

Slashcode: Un servidor de aplicaciones libre

Alvaro del Castillo San Félix
Desarrollador y admin de software libre
Barrapunto.com

acs@barrapunto.com
<http://mi.barrapunto.com/acs>

Slashcode, más conocido como slash, es la herramienta que se creó originalmente para dar servicio en Slashdot.org, uno de los primeros weblog que apareció en la red.

El éxito del sitio rápidamente hizo evolucionar el software que lo sostenía y, al apoyar con fuerza Slashdot el software libre, las circunstancias obligaron a que slash fuera publicado en una primera versión 0.1, que cuenta la leyenda sólo su creador, Rob Malda, fue capaz de instalar.

A lo largo de esta presentación iremos cubriendo los distintos aspectos de slash, su evolución, sus múltiples puntos fuertes y algún problema que aún tiene, como su laboriosa instalación

Una de las necesidades fundamentales ha sido la de tener un rendimiento excepcional, para ser capaz de servir los millones de páginas mensuales que soporta Slashdot. Por ello, se eligió como entorno de desarrollo a Apache en conjunción con mod_perl algo que ha permitido en la actualidad servir más de un millón y medio de páginas diarias.

slash ha sido un software que ha evolucionado con fuerza y más desde que VA Linux se hizo cargo de él. En la actualidad hay un gran equipo de hackers de Perl y Apache que han logrado construir una sólida arquitectura extensible, algo que ha culminado en slash 2.0 conocido como "Bender".

En la actualidad, slash ha logrado convertirse en un servidor de aplicaciones muy potente, con gran rendimiento, fácilmente extensible y con licencia GPL. En esta presentación analizaremos la evolución de slashcode y presentaremos la nueva arquitectura de slashcode 2.0, así como el desarrollo de plugins.

Durante esta ponencia se va a presentar la arquitectura de slash 2.0 y se va a mostrar como se desarrolla un plugin o extensión dentro de la nueva arquitectura.

1. ¿Por qué hablar de slash?

slash es el software que hace ya funcionar a decenas de sitios en la red, entre ellos Slashdot y Barrapunto.

slash es un software que lleva ya varios años desarrollándose, probándose en entorno en producción y demostrando una escaldabilidad y robustez que le ponen al mismo nivel que muchas soluciones comerciales para montar comunidades virtuales en la red

slash utiliza la licencia GPL para cubrir su código, tiene una amplia comunidad de desarrolladores y usuarios a su alrededor y está respaldado por una de las compañías principales dentro del mundo del software libre: VA Linux.

2. Breve historia de slash: Slashdot y Barrapunto

Slashdot nació de la mano de Rob Malda como la consecución de una serie de ideas que venían oyéndose en la red: sitios interactivos, comunidades virtuales, noticias, weblog ... Rob Malda logró captar todas estas ideas y sintetizarlas en slashdot, un sitio que resumía noticias aparecidas sobre tecnología en otros sitios de la red, y que poco a poco fue cobrando mucha fuerza en la red.

La actualización diaria de noticias, la forma de presentar la información, la navegación histórica por todo lo publicado, las búsquedas en los contenidos, las URL únicas por artículo y la posibilidad de realimentar el sitio con noticias y comentarios por parte de los usuarios fueron claves en su éxito.

En 1999, y tras una visita en la que se conoció Gildot, se decidió lanzar el proyecto Barrapunto que, basado en el software de slashdot y tomando parte de sus ideas, se ha convertido en el centro de noticias de software libre y tecnología en español de la red.

Barrapunto ha ido evolucionando con el código de slash hasta el punto de llegar a colaborar con el desarrollo actual de slash. En la actualidad Barrapunto ha migrado a la última versión de slash, la 2.0, y ya prepara la migración a 2.2 que será la plataforma

estable para el desarrollo de aplicaciones basadas en slash, y que hace ya funcionar a Slashdot.

3. Arquitectura básica: Perl (mod_perl) y Apache

Cualquier desarrollo basado en software libre para el web tiene como primera opción en la actualidad el servidor de web Apache. El disponer del 60% de cuota de mercado es una clara muestra de la gran aceptación que ha tenido, con una Fundación detrás apoyada por empresas como IBM, y con un continuo desarrollo. En los próximos meses culminará con la versión 2.0 de Apache, que solventará algunos problemas de rendimiento debido a su arquitectura basada en procesos, y que seguro, marcará un antes y un después en el desarrollo de aplicaciones para Internet.

Junto con Apache tenemos a mod_perl y con éste, a Perl el cual a pesar de la fama de lenguaje para tareas sencillas y orientadas al procesamiento de cadenas, Perl es un lenguaje muy maduro que tiene más de una década a sus espaldas de evolución. Muestra de su madurez es la gran comunidad de desarrolladores que tiene a su alrededor y de sitios desde los que se apoya el lenguaje, así como la apuesta de O'Reilly por este lenguaje. Es un lenguaje de desarrollo rápido que una vez se supera la primera curva de aprendizaje es un lenguaje muy productivo con características avanzadas en campos como la orientación a objetos.

Perl es, después de C, el lenguaje que mejor se integra con Apache gracias a mod_perl, un gran trabajo de integración de Apache y Perl en el que se ha cubierto dos frentes: mostrar la API de Apache a Perl a través de una interfaz orientada a objetos que permite el desarrollo de módulos de Apache desde Perl e incluir dentro de Apache el entorno de ejecución de Perl, con lo que no es necesario que con cada petición de programa Perl se cargue el entorno de ejecución, es decir, el intérprete de Perl. La API de Perl a Apache permite acceder a todas las fases en las que una petición de una URI es atendida en el servidor Apache, lo que permite un control total sobre como el servidor Apache presta servicios.

El gran rendimiento de mod_perl se logra manteniendo en memoria compilado todo el código Perl de las librerías que se va cargando. De esta forma, llamadas repetidas a funciones ya cargadas provocan llamadas a código precompilado en memoria por lo que su ejecución es muy rápida. De esta forma es posible el servir páginas dinámicas con unos rendimientos muy altos.

4. Problemas con slashcode pre 2.0

La versión 1.0 de slash tenía el problema de ser una evolución natural de las primeras versiones de slash, con alguna pequeña reorganización en el código y la introducción de nuevos servicios. Pero en esencia, se basaba en la misma arquitectura que nació de un proyecto en desarrollo continuo y regido por las necesidades del día a día de Slashdot. Los principales problemas que comenzaron a ser serios obstáculos en el desarrollo de slash fueron:

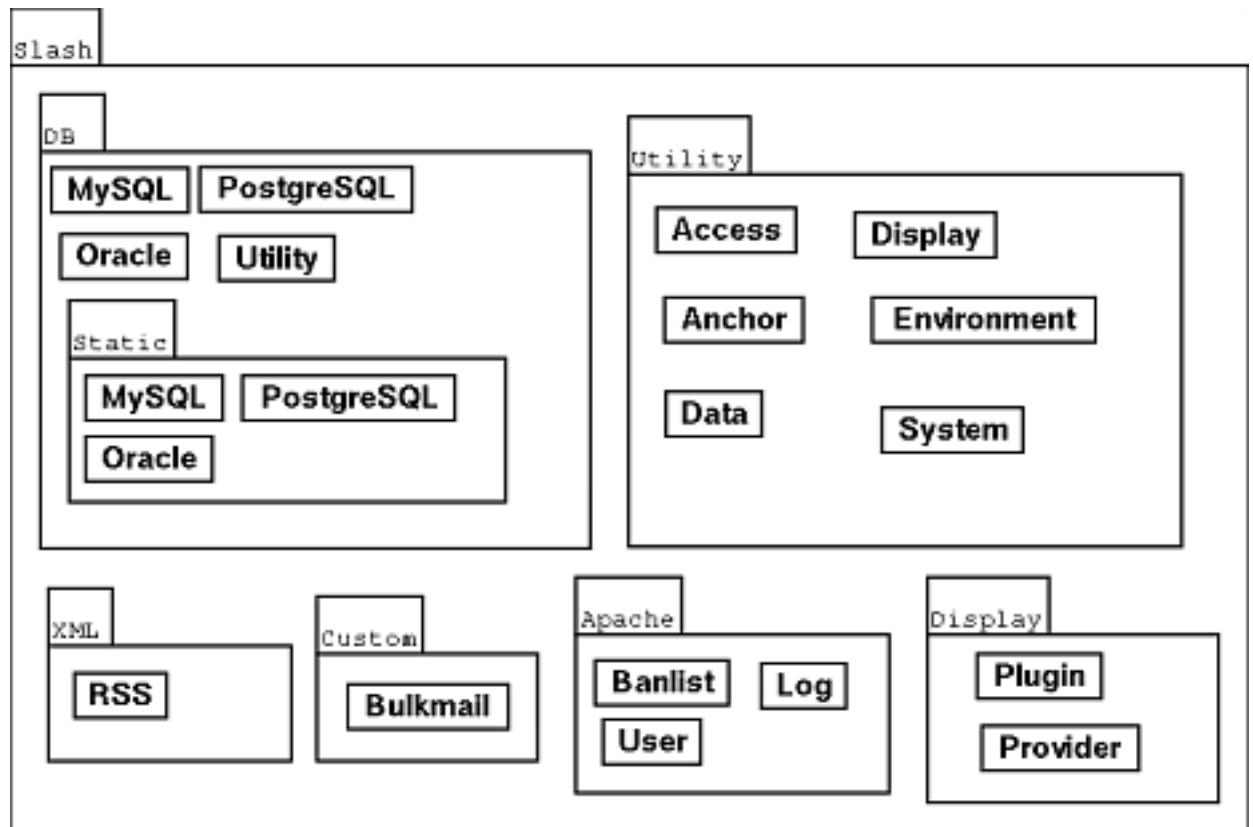
- El código HTML que se mostraba al usuario del sitio aparecía repartido por varios servlets Perl (cgis desarrollados en Perl) así como en alguna librería e incluso en la base de datos. La lógica de visualización y del programa estaban mezcladas.
- No existía una interfaz clara de acceso a base de datos, apareciendo peticiones SQL en el código, peticiones además dependientes de MySQL.
- Existía una gran librería en la que se metía toda la lógica del código, dificultando su evolución y mantenimiento.
- La idea de separar de forma clara las capas de visualización, acceso a datos y lógica de la aplicación se hizo cada vez más fuerte y, tras la publicación de la última versión 1.x de slash se tomó la determinación de llevar a cabo una gran transformación del sistema, utilizando un sistema de plantillas, creando una API de acceso a datos, y dividiendo la gran librerías Slash.pm en varias librerías que resumían distintas partes de la lógica del sistema. A esta nueva versión se la llamó Bender y es slash 2.0, de la que vamos a hablar en la ponencia.

5. Sistema de publicación y servicios web

slash es un software que ha sufrido una gran evolución a lo largo del tiempo, algo que se nota en lo bien que está organizado el software, el código fuente, las tablas de SQL y todos los elementos que lo forman. Incluso a nivel de documentación, gran parte de las librerías están documentadas, así como las distintas utilidades que forman el sistema. El software dispone de múltiples herramientas de apoyo para facilitar las labores de administración haciendo que todo el entorno sea muy sencillo de mantener.

El núcleo de slash lo forman un conjunto de librerías comunes. En el siguiente gráfico podemos observar dicha organización tras la última reestructuración de las mismas en slash 2.2, versión conocida como Fry, y que se usa actualmente en Slashdot.

Figura 1. Organización de las librerías de slash



Como vemos en la figura, aparecen claramente separadas las librerías de acceso a base de datos (Slash::DB), de visualización de datos al usuario (Slash::Display) y de lógica de funcionamiento del sistema (Slash y Slash::Utility). Podemos también citar a Slash::Apache, una unión íntima entre Apache y Perl, que permite desde los módulos Perl controlar con gran detalle el funcionamiento de Apache.

Slash::DB define una interfaz de acceso a base de datos, que debe ser implementada para todas las bases de datos con las que se quiere que el sistema funcione. En la actualidad sólo MySQL tiene un soporte completo, aunque tanto Postgres como Oracle tienen ya versiones bastante avanzadas.

Slash::Display hace uso intensivo del Template Toolkit, un sistema de plantilla para Perl que ha buscado como objetivos ser extensibles y tener un gran rendimiento. Todo el HTML que se ve en el sistema slash está basado en estas plantillas, no apareciendo

nada de este código dentro de los servlets Perl o las librerías de Slash. Es sin duda el subsistema más potente de slash y donde se ha invertido un enorme trabajo.

Slash y Slash::Utility contienen toda la lógica de funcionamiento del sistema, y ofrecen un API de programación para todos los servlets que implementan el funcionamiento del sistema, así como para los plugins (extensiones) que se pueden desarrollar de forma rápida y sencilla para aumentar la funcionalidad de slashcode.

La arquitectura de slashcode 2.0 se divide en tres partes: librerías de soporte a la arquitectura (API slashcode), sistema de presentación de los datos basado en Template Toolkit y almacenamiento en base de datos utilizando un sistema de caché para optimizar su uso.

6. Extensión a través de plugins

Los plugins en slashcode son una extensión natural de las tres patas en las que se basa slashcode: lógica, presentación y datos. A lo largo de la presentación de la ponencia, se presentará el plugin Repository que implementa un sistema de gestión de plugins, temas y demás extensiones que permite slashcode.

Cuando detectamos una nueva funcionalidad que no está cubierta en el sistema slash, podemos hacer uso de todas sus APIs para implementar en una extensión dicha funcionalidad. Estas extensiones suelen tener una parte de datos que hará uso de la base de datos disponible para su almacenamiento, creará sus propias plantillas con las que se mostrará la interfaz a través de la cual el usuario interactuará con la extensión.

Los plugins podrán hacer uso de todos los datos referentes a los usuarios, sus perfiles, la configuración del sistema, el sistema de gestión de contenidos o las librerías de tratamiento de XML. Todo está claramente accesible a través de APIs, como vamos a mostrar en el ejemplo que presentaremos en la ponencia.

7. Funcionalidad propia de Barrapunto

A lo largo de la evolución de Barrapunto con slashcode, se han ido desarrollando una serie de nuevas funcionalidades que inicialmente se incorporaron al código de slash en la versión 1.09 modificando partes de slash propiamente.

Una vez que se publicó slash 2.0 se dispuso del sistema de plugins, se intentó migrar toda esta nueva funcionalidad a plugins, proceso que se concluyó con éxito y con el que

se pudo proporcionar a slash modificaciones concretas como el sistema de ACLs que ha sido integrado dentro de la ram principal de slash.

Las funcionalidades principales de Barrapunto que son novedad con respecto a slash son:

- Secciones virtuales: una misma historia puede estar presentada en más de una sección a la vez, lo que permite tener diferentes vistas del sistema y compartir mucha información asociándola de distintas formas.
- MiBarrapuntos: Cada usuario registrado puede disponer de su propio weblog, en el que podrá publicar lo que a él le interese, bien como una historia nueva o bien captando historias de otras secciones de Barrapunto.
- Sección En Bruto: Todas las contribuciones que llegan a Barrapunto son publicadas en una sección especial, conocida como En Bruto, con lo que se elimina cualquier indicio de censura.
- Evoluciones: Una misma historia puede ser editada en diferentes fases y por diferentes personas, dando así contextos diferentes a la información y permitiendo una edición colaborativa.
- Auto Secciones: Existen secciones automáticas que muestran las historias más visitadas, más comentadas, que aparecen en los weblog propios de los usuarios ... son secciones que se construyen de forma automática y que proporcionan nuevas vistas del sitio.

8. Hacia la web semántica

Uno de los aspectos que más se está fortaleciendo dentro de slashcode es la posibilidad de acceder a la información por XML, es decir, utilizando DTD estándares como RSS que permiten obtener la información de una forma estructurada y con mucho más contenido semántico que las páginas HTML.

Hasta ahora el DTD elegido ha sido RSS, RDF Site Summary, el cual se está convirtiendo en un estándar de facto a la hora de exportar contenidos en la red. Este formato permite exportar la información organizada en canales y es muy adecuado para los sitios de noticias, donde se exporta el titular de una noticia, junto con la URL de la misma una descripción.

Dentro de slashcode se han creado librerías bajo Slash::XML que facilitan mucho la creación de información en formato RSS. Se ha incorporado en Julio el uso de OCS,

Open Content Syndication, que permite exportar varios canales de forma simultánea. Estos canales se pueden especificar en formato RSS, por ejemplo. De esta forma la sindicación de contenidos se lleva más lejos aún que con RSS.

9. Futuro de slash: Fry

El futuro inmediato de slash se llama Fry y será la versión 2.2 de slash. Entre sus novedades veremos muchas nuevas extensiones, como el sistema de mensajería, nuevas funcionalidades, como la integración de sistemas de corrección ortográfica (ispell) previos a la inserción de contenidos o un mayor control sobre los usuarios, permitiendo bloquear cuentas de usuario o impidiendo que pongan comentarios, con el objetivo de disponer de mejores herramientas para gestionar comunidades de personas cada vez mayores.

10. Conclusiones

Dentro de esta ponencia se ha querido presentar el sistema de publicación de contenidos slash para los desarrolladores. Ha sido una introducción eminentemente práctica, donde se ha observado como utilizar el sistema en distintos entornos y ampliarlo, así como los detalles de su arquitectura.

slash está llamado a convertirse en un completo software con el que construir todo tipo de servicios en la red, servicios con comunidades muy activas y en el que la gestión de los datos y la construcción basada en las colaboraciones de la comunidad serán sus pilares fundamentales.

11. Referencias

- Slashcode (<http://slashcode.com>)
- Slashdot (<http://slashdot.org>)
- Barrapunto (<http://barrapunto.com>)

