

# Gestión distribuida de servidores con herramientas Open Source: Cfengine y Subversion

## PONENCIAS

### Distributed system management with Open Source tools: Cfengine and Subversion

◆ Miguel Á. García Lax, Fco. Javier García Ros, Angel Luís Mateo, Alfonso Marín Marín, Javier Tavira, Fco. Yepes Candel

#### Resumen

Este artículo describe la solución para la gestión de servidores que viene siendo usada en la Universidad de Murcia en tareas como puesta en producción de una máquina y mantenimiento de servicios mediante herramientas de software libre como Cfengine y Subversion. También recomendamos algunas pautas a tener en cuenta para la implantación de mecanismos similares.

**Palabras clave:** gestión distribuida, servidores, software libre, Cfengine, Subversion, administración de sistemas.

#### Summary

This article describes the solution for managing servers that is being used at the University of Murcia in tasks such as putting into production new hosts and maintenance services, using open source tools like Cfengine and Subversion. They also suggest some guidelines to be taken into account.

**Keywords:** managing distributed servers, open source, Cfengine, Subversion, systems administration.

## 1. Introducción

La incorporación de la virtualización en los CPD ha permitido disponer fácilmente de mecanismos de alta disponibilidad y redundancia a costa de incrementar notablemente el número de máquinas a administrar. Mantener la coherencia entre las máquinas de un cluster, aplicar acciones a todas y cada una de ellas, etc., es uno de los nuevos retos que conlleva este elevado número de servidores. Queremos exponer como las herramientas de código abierto como **Cfengine** [1], y **Subversion** [2], que están siendo utilizadas durante años en la Universidad de Murcia, pueden ayudar enormemente a realizar la gestión y administración de las máquinas [5,6], así como a centralizar el punto de administración y recolectar las tareas que hay que llevar a cabo en cada una de las máquinas.

## 2. Breve descripción de las herramientas

La solución hace uso intensivo de Cfengine como gestor de configuración y una herramienta de control de versiones para mantenimiento del histórico de cambios así como autor. A continuación damos una breve descripción de cada una de ellas.

**Subversion (SVN)** es uno de los sistemas de control de versiones más extendido. Es bien conocido dentro de la comunidad de código abierto y es utilizado en numerosos proyectos incluyendo la Apache Software Foundation, KDE, GNOME, FreeBSD, GCC, Python, Django, Ruby, Mono, SourceForge.net, etc. **Cfengine** es también una herramienta de código abierto, decana en la configuración y gestión de sistemas basada en políticas, que utiliza un **lenguaje declarativo** para la realización de todas las tareas. Una de las características más importantes es que las acciones deben ser pensadas y aplicadas para que

◆  
Mantener la coherencia entre las máquinas de un cluster es uno de los nuevos retos que conlleva este elevado número de servidores.

◆  
Subversion (SVN) es uno de los sistemas de control de versiones más extendido



sean convergentes en los sistemas implicados, de forma que pueda ejecutarse la aplicación en cualquier momento e independiente del número de veces que se haga.

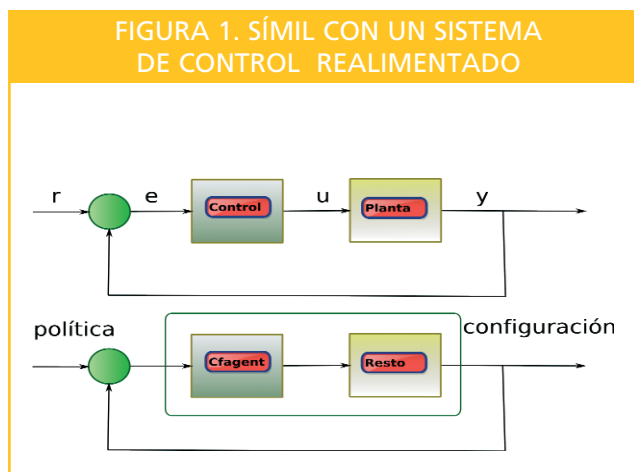
De la gran cantidad de tareas que pueden llevarse a cabo, podemos destacar:

- chequeo y establecimiento de permisos.
- creación y borrado de ficheros.
- mantenimiento, creación, modificación de ficheros de configuración.
- ejecución de scripts basados incluso en el estado del sistema.
- realización de sumas de comprobación (checksums) de ficheros al estilo de tripwire.
- comprobación y arranque en caso de que sea necesario de demonios.
- avisos en caso de anomalías y alertas.

Cfengine utiliza un lenguaje declarativo para la realización de todas las tareas

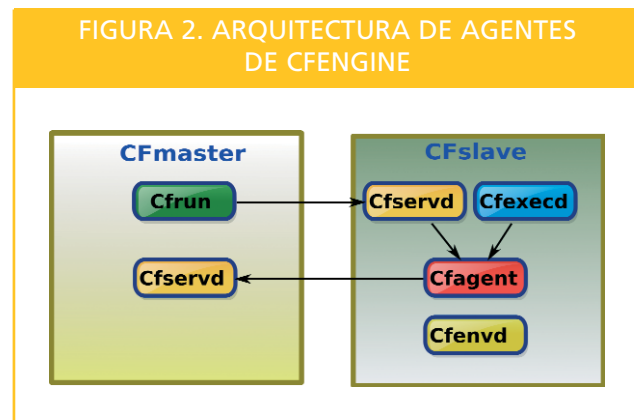
El lenguaje permite matizar las acciones en función de lo que denomina clases pudiendo condicionar la aplicación de una política a propiedades como: arquitectura, tipo de sistema operativo, carga de la máquina, dominio, clases definidas por el usuario... Esto nos permite el grado de granularidad que queramos, para aplicar acciones a determinados grupos de máquinas en función de las agrupaciones que necesitemos.

El modelo subyacente de funcionamiento de Cfengine está basado en un modelo de **inmunidad [4]**, cuya idea es definir el/los estado/s saludable/s de la máquina. Cuando un sistema cumple con dichos estados está ,diríamos, sano, mientras que una variación lo convertiría en un sistema enfermo, aplicando medidas correctivas para acercarlo al estado definido. Se podría establecer un paralelismo con los sistemas de control[3], en las que la política actúa como señal de referencia, produciendo cambios en la configuración del sistema como podemos ver en la Figura 1.



Cfengine está basado en un modelo de inmunidad, cuya idea es definir el/los estado/s saludable/s de la máquina

La herramienta Cfengine está diseñada como un sistema multiagente (ver Figura 2) en los que podemos destacar: agentes servidores (**cfserverd**), agentes de aplicación de las políticas (**cfagent**) y agentes recolectores de información (**cfenfd**) que van tomando información del estado de la máquina para aprender los parámetros de normalidad de la misma. En las variaciones de estos parámetros, medidos sobre el número de veces que nos hemos alejado de la desviación estándar, podemos incluso llegar a aplicar acciones. Esto es especialmente interesante en el descubrimiento y tratamiento de anomalías.



### 3. Sobre la elección de herramientas

Sobre el sistema de control de versiones, elegimos Subversion, porque era el sistema ya implantado en nuestra organización y cumple perfectamente con los requisitos. Otras alternativas como CVS serían igual de válidas. Cabe decir, que aunque en sí mismo podría constituir un mecanismo de gestión de configuración, perderíamos la enorme potencia e interesantes funcionalidades que las herramientas como Cfengine nos ofrecen.

Sobre la herramienta de gestión de configuraciones se eligió Cfengine por ser de las más maduras y ser muy ampliamente utilizada. En los últimos años han venido cogiendo fuerza alternativas como Puppets, que al ser posterior recogió mucho de los puntos fuertes de aquella y ha mejorado alguna de sus deficiencias. Nuestra recomendación es que si no se tiene nada implantado se analicen al menos ambas, teniendo en cuenta las nuevas mejoras que incorporará Cfengine en su versión 3.

### 4. Entorno y Descripción de la Solución

En una primera fase de la implantación de Cfengine en nuestra universidad, contábamos con un considerable parque de máquinas SUN bajo Solaris y empezábamos a montar los nuevos servicios sobre máquinas linux. Con el tiempo hemos evolucionado hacia máquinas Linux, virtualizadas mediante XEN, por tanto nuestra configuración de Cfengine ha permitido y permite la convivencia de distintos entornos.

Cfengine hace uso de un sistema de políticas centralizadas. Esto suele traducirse en un único host que contiene las políticas y que llamaremos 'cfmaster'. En cada host se ejecuta un agente que es el encargado de ver periódicamente si hay nuevas políticas en el cfmaster, las descarga en local y las aplica. Al ser la ejecución de los agentes distribuida se evita el punto de fallo, de forma que si la máquina maestra del Cfengine no estuviera disponible, cada una de la máquinas gestionadas seguiría funcionando con su última política.

En el cfmaster se incluye además de las políticas, los ficheros específicos y plantillas que queramos distribuir, estando todos estos ficheros bajo el mismo sistema de control de versiones. De esta forma los operadores editan localmente, publican los cambios en el sistema de control de versiones identificados con su usuario y cuando se considera oportuna se valida la modificación y actualizándose en el cfmaster, de forma que hasta que no se aplique la actualización no se empezarán a distribuir los cambios, siendo posible la revisión por terceros. En la figura 3 se puede observar la un esquema de la solución.

En los últimos años han venido cogiendo fuerza alternativas como Puppets

Cfengine hace uso de un sistema de políticas centralizadas

FIGURA 3. ESQUEMA DE LA SOLUCIÓN

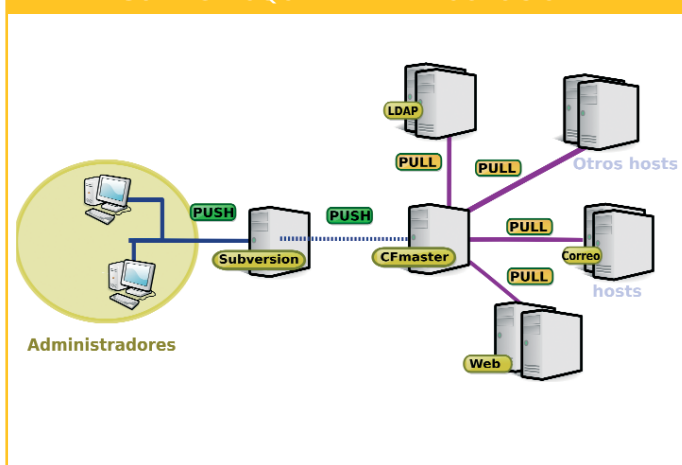
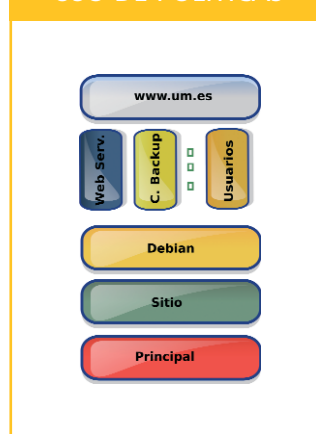


FIGURA 4. EJEMPLO DE USO DE POLÍTICAS





## 5. Tipo de tareas administrativas

Estas políticas suelen estar clasificadas de la siguiente forma:

- 1) Configuración/Instalación de servicios comunes a todas la máquinas: sincronización de reloj, configuración de estafeta local para envío de correos, servicios snmp, servicio de control de cambio de ficheros (tipo tripwire), revisión de permisos de ficheros importantes genéricos, control de suids, comprobación de que estos servicios básicos estén corriendo (snmp, sshd, ...) , etc.
- 2) Configuración/Instalación de servicios por sistemas operativos: configuración de repositorios específicos, clientes de copias de seguridad, distribución de claves ssh, tcp-wrappers.
- 3) Configuración/Instalación de servicios por categorías: programas y plantillas de servidores web (Apache), bases de datos (mysql), etc.
- 4) Configuración/Instalación específica de servicios: las máquinas del web corporativo, webmail, correo, etc.

Recomendamos extraer las partes comunes y crear políticas para ellas que compartan el mayor número de máquinas

Esta organización es a grandes rasgos la que recomendamos finalmente: extraer las partes comunes y crear políticas para ellas que compartan el mayor número de máquinas, de forma que una configuración final sea la aplicación de distintas políticas genéricas, de sistemas operativos, de servicios genéricos y por último de servicio final o principal. En la figura 4 podemos ver un ejemplo de configuración de un servicio como conjunto de políticas. De todas formas, de cara a una organización que quiera empezar, una configuración sencilla con pocas abstracciones es el mejor punto de partida.

La evolución de los servicios pasa por la modificación en las políticas evitando en todo lo posible la actuación individual en ninguna máquina o grupo de máquinas sin que quede reflejado en la política.

## 6. Ventajas de la solución

Podríamos resumir las principales ventajas en:

- reducción de incidencias por máquina y mejora en la estabilidad de los servicios.
- reducción de la atención específica y de tareas repetitivas en cada una de las máquinas.
- tiempos menores de puesta en explotación de un servicio una vez definidas sus políticas.

Además de los beneficios arriba indicadas nos ha aportado adicionalmente:

- hacer de la administración una tarea documentada y explícita.
- fácil reutilización del trabajo en la migración de nuevos servicios y aumento del conocimiento global de todo el equipo de administración en estas tareas.
- punto único y centralizado de políticas con control de usuario.
- registro histórico de cambios en la configuración de los sistemas.

Como puntos débiles, hemos notado una falta de potencia de Cfengine en los mecanismos de iteración así como una falta de orientación a objetos del lenguaje, que permita definir políticas y heredar mediante clases su comportamiento, sobrescribiendo aquellos métodos que sean particulares, reflejando de forma más elegante la jerarquía de nuestros sistemas. Probablemente la nueva versión 3 de la herramienta, que supone una gran rediseño del lenguaje, mejore estos puntos, dejando intactos los enormes beneficios que la actual reporta.

De cara a una organización que quiera empezar, una configuración sencilla con pocas abstracciones es el mejor punto de partida

## 7. Conclusión

La utilización de ambas herramientas nos ha supuesto una gran beneficio en términos reducción de incidencias y de tiempo de la puesta en explotación de nuevos servicios, además de que la definición mediante un lenguaje de tipo declarativo de las tareas necesarias para la ejecución de un servicio, permite dejar constancia de forma escrita de lo realizado, sirviendo de base y apoyo, así como de reutilización en el diseño de los nuevos. Recomendamos una vez adquirida algo de práctica con el sistema, la generación de clases que abstraigan el comportamiento y sean reutilizables para otros servicios y configuraciones.

## Referencias

- [1] Cfengine.  
Consultado en (<http://www.Cfengine.org/>) 10-10-2008
- [2] Subversion.  
Consultado en (<http://Subversion.tigris.org/>) 10-10-2008
- [3] A control theory perspective on configuration management and Cfengine.  
Consultado en (<http://controlofsystems.org/febid2006/files/control-Cfengine.pdf>) 10-10-2008
- [4] M. Burgess, A Tiny Overview of Cfengine: Convergent Maintenance Agent, Proceedings of the 1st International Workshop on Multi-Agent and Robotic Systems, MARS/ICINCO 2005.  
Consultado en ([http://www.iu.hio.no/~mark/papers/tiny\\_intro.pdf](http://www.iu.hio.no/~mark/papers/tiny_intro.pdf)) 10-10-2008
- [5] Luke A. Kanies Distributed Cfengine, O'reilly Sysadmin 2004.  
Consultado en ([http://www.onlamp.com/pub/a/onlamp/2004/05/13/distributed\\_Cfengine.html](http://www.onlamp.com/pub/a/onlamp/2004/05/13/distributed_Cfengine.html)) 10-10-2008
- [6] Kirk Bauer. Automating Security with GNU Cfengine, Linux Journal 2004.  
Consultado en ([http://www.onlamp.com/pub/a/onlamp/2004/05/13/distributed\\_Cfengine.html](http://www.onlamp.com/pub/a/onlamp/2004/05/13/distributed_Cfengine.html)) 10-10-2008

◆  
Cfengine reduce el número de incidencias por máquina y mejora en la estabilidad de los servicios

◆  
Recomendamos la generación de clases que abstraigan el comportamiento y sean reutilizables para otros servicios y configuraciones

**Miguel Á. García Lax**  
glax@um.es

**Fco. Javier García Ros**  
jgarcia@um.es

**Angel Luís Mateo**  
amateo@um.es

**Alfonso Marín Marín**  
almarin@um.es

**Javier Tavira**  
jtavira@um.es

**Fco. Yepes Candel**  
pacoy@um.es

Ática, Universidad de Murcia