

Una alternativa a Apache...

# AOLSERVER



Pocos servidores web realmente estables y robustos hay en el mercado y pocas posibilidades de elección, pero posiblemente no nos habíamos percatado de la existencia del AOLserver, un servidor antiguo en desarrollo y muy actual en prestaciones y potencia. **POR PIETRO ZUCO**

**A**OLserver es un proyecto liderado por AOL (America Online) en Source Forge. Actualmente se utiliza como servidor Web de importantes portales con una gran carga de usuarios, desde la propia Web de AOL hasta la de Netscape o Creative Commons.

AOLserver es multihilo, soporta TCL, es altamente escalable y está particularmente optimizado para sitios web dinámicos de gran envergadura. Quizás algunos se puedan preguntar ¿Por qué un servidor Web tan potente y peculiar no ha llegado a alcanzar la fama de otros, como por ejemplo Apache? La respuesta es a la vez simple y sorprendente: falló el Marketing. Vamos a ver en este artículo

cómo instalarlo, configurarlo y poder sacarle partido desde el primer momento y veremos que no nos dejará nada que desear.

## ¿Soporta TCL?

Quizás esto pueda sonar un poco extraño para los que están acostumbrados a Apache o servidores similares, dado que dependen de un fichero de configuración propio con una sintaxis determinada, pero este no es el caso de AOLserver. TCL lo es todo en este servidor, desde su fichero de configuración hasta el lenguaje utilizado en el API de programación que incorpora. Tanto es así que algunos le definen no sólo como un servidor Web

sino también como un servidor de aplicaciones.

AOLserver puede mostrar un típico contenido estático, así como páginas ADP (AOLserver Dynamic Pages) para contenido dinámico, basadas en la incrustación de código TCL en HTML.

La ejecución del código en TCL la realiza directamente sin pasarla por ninguna interfaz CGI.

## Instalación

La principal forma de descargar la fuente e instalarlo es a través del repositorio CVS. De todas formas existen paquetes DEB o RPM que se mantienen relativamente actualizados. En este artículo abordare-

mos la instalación en una Debian Sarge.

La distribución principal contiene una serie de módulos mínimos, aunque podemos instalar más de 70 módulos, desde el soporte de bases de datos como PostgreSQL u Oracle hasta un módulo para Aspell de corrección ortográfica. El desarrollo de módulos es bastante activo y hay una comunidad



Figura 1: Uno de los cuadros de configuración en el momento de la instalación.

considerablemente amplia que aporta mejoras y documentación, por lo tanto no nos veremos desamparados a la hora de necesitar información y ayuda.

Instalaremos primero el core del servidor y luego un módulo para el soporte de PostgreSQL.

Nos hacemos root e instalamos el paquete del repositorio:

```
apt-get install aolserver4
```

En la fase de configuración automática del paquete se nos presentarán varias pantallas solicitando información que se incluirá en el fichero de configuración. La información mínima requerida es el número de puerto, el nombre del usuario con el que se va a lanzar el servidor y la IP de nuestra máquina. Todos estos parámetros se pueden pasar directamente al ejecutable, que en el caso de Debian es `usr/sbin/aolserver4-nsd`. De todas formas, no es necesario preocuparse de responder correctamente en esta fase, dado que toda la información que se solicita puede ser modificada en el fichero de configuración. Ver figura 1.

Finalizada la configuración inicial, instalaremos el módulo para poder conectar con PostgreSQL:

```
apt-get install aolserver4-nspostgres
```

Llegados a este punto ya podemos utilizar nuestro nuevo servidor Web. En la carpeta `/etc/init.d/` debemos de tener el script `aolserver4` que lanzará el servidor en el arranque. Este script además de contener los parámetros que introducimos en los cuadros de dialogo, ha sido ejecutado automáticamente en la fase de instalación, por lo tanto, si todo ha ido bien el servidor debe estar funcionando.

Para comprobarlo apuntemos con el navegador a `http://localhost` ver figura 2.

Hay que tener en cuenta que si estamos ejecutando un Apache escuchando por el puerto 80, la configuración por defecto del AOLserver nos dará problemas. Por lo tanto hay que detener el Apache antes de instalar el AOLserver o cambiar el número del puerto.

## La configuración

Como otros servidores Web, AOLserver utiliza un fichero de configuración, que para el caso de Debian es `/etc/aolserver4/aolserver4.tcl`

Este fichero tiene algo en particular que lo diferencia de otros servidores Web y que a primera vista puede sonar un poco extraño: *Se trata de un script en TCL.*

Como decíamos antes, AOLserver tiene un API muy completa en TCL y esto nos da la posibilidad de incluir código dentro del fichero de configuración y forzar al AOLserver para que en cierta manera se *autoconfigure* antes de arrancar. Algunas de las opciones más comunes que se pueden pasar al ejecutable son:

- **-u** Ejecutarse como usuario `<usuario>` \* **-g** Ejecutarse con grupo `<grupo>` \* **-b** Dirección IP y puerto `<dirección:puerto>` \* **-s** nombre del servidor `<servidor>` en el fichero de configuración \* **-t** Fichero de configuración `<file>` (REQUERIDO)

Por defecto, la configuración que trae en el paquete para Debian es:

```
Usuario: www-data
Grupo: www-data
IP: 127.0.0.1
Puerto: 80
Fichero de configuración: /etc/aolserver4/aolserver4.tcl
```

Estos datos son los que vamos a usar a título de ejemplo y se encuentran en el script de arranque que se crea tras la instalación, que se ubica en: `/etc/init.d/aolserver4`

## El fichero de configuración

En el fichero de configuración de AOLserver `/etc/aolserver4/aolserver4.tcl` vemos que los parámetros que se pasan al ejecutable pueden ser configurados directamente en el propio fichero.

Las partes más importantes a destacar son:

```
set httpport 80
```

Para indicar el puerto en el que va a escuchar.

```
set hostname localhost
```

### Listado 1: prueba1.adp

```
01 <HTML>
02 <HTML>
03 <HEAD>
04 <TITLE>Prueba de ADP
   (1)</TITLE>
05 </HEAD>
06 <BODY>
07 <H1>Una prueba de ADP (1)</H1>
08 <p> ADP es como usar TCL embe-
   bido en HTML </p>
09
10
11 <FORM ACTION="prueba1.adp"
   METHOD="POST">
12     Introducir valor para
   contar:<br><INPUT TYPE="text"
   NAME="valor" VALUE="10">
13     <INPUT TYPE=submit
   VALUE="Contar">
14 </FORM>
15
16 <%
17     set r [ns_conn form
   $conn]
18     set valor [ns_set get
   $r valor]
19 %>
20 <b> Contamos hasta <%=
   $valor%>: </b><br>
21 <table border="1">
22 <tr>
23 <%
24     for {set i 1} {$i <=
   $valor} {incr i} {
25     ns_puts "<td>$i</td>"
26     }
27 %>
28 </tr>
29 </table>
30 <br>
31 </BODY>
32 </HTML>
```

```
set address 127.0.0.1
```

Estos parámetros deberán ser cambiados en función de la configuración que tengamos en nuestra máquina. El primero es el nombre de nuestro equipo y el segundo es nuestra IP.

```
set servername main
```

Nombre que le daremos al servidor. Estas variables que creamos las podemos usar en el resto de la configuración y este es un ejemplo que muestra la versatilidad del AOLserver.

### Listado 2: prueba2.adp

```
01 <HTML>
02 <HTML>
03 <HEAD>
04 <TITLE>Prueba de ADP
05 (2)</TITLE>
06 </HEAD>
07 <BODY>
08 <H1>Una prueba de ADP (2)</H1>
09 <p> Datos de configuración del
10 servidor </p>
11 <%
12     set config_file
13     [ns_info config]
14     set home_dir [ns_info
15     home]
16     set nombre_host
17     [ns_info hostname]
18     set paginas [ns_info
19     pageroot]
20     set plataforma
21     [ns_info platform]
22     set tiempo [ns_info
23     uptime]
24 >%
25 <b>Fichero de configuración
26 </b><%= $config_file%><br>
27 <b>Carpeta de instalación de
28 AOLserver </b><%=
29 $home_dir%><br>
30 <b>Nombre de la máquina
31 </b><%= $nombre_host%><br>
32 <b>Carpeta que contiene la Web
33 </b><%= $paginas%><br>
34 <b>Plataforma </b><%= $plata-
35 forma%><br>
36 <b>Tiempo en segundos de
37 encendido </b><%=
38 $tiempo%><br>
39 </BODY>
40 </HTML>
```

Por ejemplo, si tenemos una base de datos llamada “prueba”, podemos dar como nombre al servidor “prueba”, usando luego esta variable para establecer el nombre de la base de datos, más adelante en el mismo fichero:

```
set db_name
$server
```

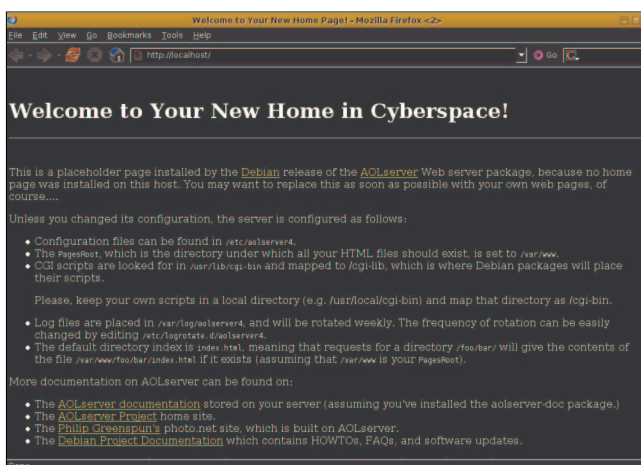


Figura 2: Web por defecto del AOLserver.

Aquí aprovechamos la potencia de TCL para poder usar variables y funciones dentro del fichero de configuración.

Otro ejemplo sería determinar el nombre de nuestra máquina sin necesidad de pasarlo explícitamente al fichero de configuración. Esto se consigue cambiando la línea anterior:

```
set hostname localhost
```

por esta:

```
set hostname [ns_info hostname]
```

Aquí llamamos a una función del API de AOLserver *ns\_info hostname* que nos devuelve el nombre de nuestra máquina.

El hecho de que casi todas las llamadas al API del AOLserver y de que los módulos empiecen por “ns”, se debe a que el nombre original de AOLserver fue *Navy Server*, dado que fue desarrollado por una pequeña empresa llamada NaviSoft que fue comprada por AOL.

Por último, la carpeta que contendrá los ficheros HTML o ADP se configura con la variable *pageroot*:

```
set pageroot /var/www
```

### El lenguaje ADP

ADP (AOLserver Dynamic Pages) es un lenguaje que puede ser embebido en HTML y es básicamente un TCL reducido que hace uso de las funciones del API del AOLserver.

Como ejemplo, el listado 1 muestra un fichero con extensión *adp* que básicamente es un HTML con partes de código en ADP. Implementa un simple

formulario que nos pide un valor para contar. Luego genera una tabla en la que cada columna contiene uno de los valores contados. Este código muestra cómo se puede usar la potencia de TCL en la creación de funciones que pueden ser llamadas como si de un CGI se tratara. El listado 2 muestra cómo obtener información relativa al servidor donde está corriendo el AOLserver. Con estos simples ejemplos nos podemos hacer una idea de las posibilidades que nos brinda ADP, similares a otros lenguajes como PHP o scripts en CGI. Es importante notar que el código ADP se debe encerrar entre los símbolos `< % % >`

### Conclusión

AOLserver puede ser una alternativa perfectamente viable respecto a Apache u otros servidores Web. Es especialmente adecuado en entornos con una gran carga de usuarios y de transacciones contra base de datos.

Su flexibilidad en la configuración lo hace especialmente adecuado en entornos extremos donde otros servidores Web no podrían adaptarse. Animémonos a explorar sus posibilidades, seguramente no nos defraudará. ■

### RECURSOS

- [1] Página principal de AOLserver <http://www.aolserver.com>
- [2] Documentación extensa (wiki) [http://panoptic.com/wiki/aolserver/AOLserver\\_Wiki](http://panoptic.com/wiki/aolserver/AOLserver_Wiki)
- [3] OpenACS comunidad virtual desarrollada sobre AOLserver <http://www.openacs.org>