

Introducción a LightTPD

Introducción

Con este documento **Anurix** pretende acercar a un administrador de sistemas UNIX al servidor web lighttpd como una alternativa fiable respecto a otros servidores http.

¿Que es LightTPD?

Lighttpd es un servidor http o https, seguro, rápido que respeta estándares y consume muy pocos recursos, todo ello con un código limpio y elegante.

Es ideal para ser usado en entornos donde la carga es máxima, se requieren respuestas rápidas y alta escalabilidad. **Anurix** lo utiliza como único servidor web para contenido dinámico, generalmente sobre estaciones x86 o sparc64 con sistemas NetBSD.

Algunas características básicas destacables que posee Lighttpd son:

- Virtual Hosting incluso por medio de MySQL.
- CGI, SCGI, FastCGI.
- Soporte de varios lenguajes de programación como PHP o Ruby a través de CGI y FastCGI.
- Admite certificados SSL y por lo tanto puede servir https.
- Autenticación con htpasswd, LDAP o MySQL.
- Tiene un módulo de reescritura y de redirección de URLs.
- Genera estadísticas mediante rrdtool.
- Permite Módulos Externos.
- Acepta Webdav.
- Acepta SSI (Server Side Includes).

Estas son características que cualquier servidor http medianamente decente debería poseer.

Algunas de las opciones que hacen más interesantes a lighttpd desde el punto de vista de **Anurix** es la posibilidad de elección del *Eventhandler*, que nos permite optimizar al máximo las respuestas de los sockets de nuestro servidor http. También creemos que una buena cualidad frente a otros servidores es la posibilidad del balanceo de carga en los servicios FCGI, con los que por ejemplo podemos distribuir las peticiones PHP entre diferentes máquinas.

Un FCGI o FastCGI es una alternativa al clásico CGI. La principal diferencia con CGI es que ejecuta un proceso persistente por cada programa que se ejecuta en vez de un proceso por cada conexión de un cliente al servidor. Lighttpd usa este interfaz para ejecutar aplicaciones escritas en PHP, Ruby o PERL.

Lighttpd se puede usar como servidor http único o combinado con otros. Se puede usar para liberar carga de otros servidores más lentos ya que, por ejemplo, durante el envío de ficheros grandes suele ser mucho más rápido que en el resto de servidores, aunque ésto lo explicaremos más adelante.

Comparado con otros Servidores.

En la web de lighttpd tenemos estadísticas comparando Lighttpd con otros servidores web (<http://www.lighttpd.net/benchmark/>)

En principio lighttpd se muestra hasta 2 o 3 veces más rápido que Apache y con un consumo de memoria muchísimo más bajo que el resto, sobretodo comparándolo con Apache. Ésto se debe en parte al uso de prefork por Apache, que lo hace mucho más lento, sobretodo en situaciones de carga extremas, llevándole a colgarse o a quedar inutilizado durante algún tiempo e incluso utilizar recursos del servidor de forma descomunal. Es por ello que, desde hace tiempo, desde Anurix aconsejamos siempre LightTTPD a nuestros clientes.

En la mayoría de comparativas podemos ver claramente que donde gana Lighttpd es cuando el número de conexiones concurrentes es realmente elevado o cuando se envían ficheros grandes. En estas situaciones no tiene rival, ni tan sólo thttpd, un servidor mucho menos complejo, le hace sombra.

Cambiar de Apache a Lighttpd no va a ser la solución definitiva para los problemas de velocidad sirviendo PHP combinado con SQL, ya que la mayoría de veces el cuello de botella suele estar en que el código PHP o SQL no son totalmente óptimos. Eso sí, Lighttpd ayudará sin duda mucho a que una aplicación funcione mucho más rápido sobre todo en cargas altas, por eso en **Anurix** lo recomendamos en este tipo de situaciones frente a Apache u otros servidores http.

Configuración básica lighttpd

A continuación se detalla una configuración básica de Lighttpd sobre NetBSD.

```
#Configuracion Lighttpd Basica lighttpd.conf

server.modules          = (
    "mod_rewrite",
    "mod_redirect",
    "mod_alias",
    "mod_access",
    # "mod_trigger_b4_dl", <-- Permite establecer
acciones al iniciar una descarga
    "mod_auth",
    # "mod_status", <-- Muestra el estado del server en
una web generada
    # "mod_setenv", <-- Permite setar variables de
entorno
    "mod_fastcgi",
    # "mod_proxy",<-- Modulo de proxy
    "mod_simple_vhost",
    # "mod_evhost", <-- Opciones extendidas de vhost
    "mod_userdir",
    "mod_cgi",
    "mod_compress",
```

```

#           "mod_ssi", <- Server side includes
#           "mod_usertrack",
#           "mod_expire",
#           "mod_secdownload", <-- Permite securizar ficheros
por medio del servidor y aplicacion
#           "mod_rrdtool", <-Stats con rrdtool
           "mod_accesslog" )

#Crea un nuevo socket que escucha por el puerto 443 con certifiacion
SSL
$SERVER["socket"] == "xyz.xyz.xyz.xyz:443" {
    ssl.engine      = "enable"
    ssl.pemfile     = "/usr/pkg/etc/lighttpd/server.pem"
    server.name     = "ejemplo.com"
    server.document-root = "/usr/pkg/share/httpd/htdocs/"
}

## Documento principal
server.document-root      = "/usr/pkg/share/httpd/htdocs/"

##Log de error
server.errorlog           = "/var/log/lighttpd.error.log"

#Log de acceso
accesslog.filename       = "/var/log/lighttpd/access.log"

#Archivos que pueden ser indice y en que orden
index-file.names         = ( "index.php", "index.html",
                             "index.htm", "default.htm" )

#Tipos de archivo mime
mimetype.assign          = (
    ".pdf"              => "application/pdf",
    ".sig"              => "application/pgp-signature",
    ".spl"              => "application/futuresplash",
    ".class"           => "application/octet-stream",
    ".ps"              => "application/postscript",
    ".torrent"         => "application/x-bittorrent",
    ".dvi"             => "application/x-dvi",
    ".gz"              => "application/x-gzip",
    ".pac"             => "application/x-ns-proxy-autoconfig",
    ".swf"             => "application/x-shockwave-flash",
    ".tar.gz"          => "application/x-tgz",
    ".tgz"            => "application/x-tgz",
    ".tar"             => "application/x-tar",
    ".zip"             => "application/zip",
    ".mp3"            => "audio/mpeg",
    ".m3u"            => "audio/x-mpegurl",
    ".wma"            => "audio/x-ms-wma",
    ".wax"            => "audio/x-ms-wax",
    ".ogg"            => "application/ogg",
    ".wav"            => "audio/x-wav",
    ".gif"            => "image/gif",
    ".jpg"            => "image/jpeg",
    ".jpeg"           => "image/jpeg",

```

```

".png"           => "image/png",
".xbm"           => "image/x-xbitmap",
".xpm"           => "image/x-xpixmap",
".xwd"           => "image/x-xwindowdump",
".css"           => "text/css",
".html"          => "text/html encoding=utf-8",
".htm"           => "text/html encoding=utf-8",
".js"            => "text/javascript",
".asc"           => "text/plain",
".c"             => "text/plain",
".cpp"           => "text/plain",
".log"           => "text/plain",
".conf"          => "text/plain",
".text"          => "text/plain",
".txt"           => "text/plain",
".dtd"           => "text/xml",
".xml"           => "text/xml",
".mpeg"          => "video/mpeg",
".mpg"           => "video/mpeg",
".mov"           => "video/quicktime",
".qt"            => "video/quicktime",
".avi"           => "video/x-msvideo",
".asf"           => "video/x-ms-asf",
".asx"           => "video/x-ms-asf",
".wmv"           => "video/x-ms-wmv",
".bz2"           => "application/x-bzip",
".tbz"           => "application/x-bzip-compressed-tar",
".tar.bz2"       => "application/x-bzip-compressed-tar"
)

## Deniega tipos de archivo
url.access-deny          = ( "~", ".inc" )

##
# Extensiones de fichero que no seran tomadas como un fichero
estatico.
#
# Normalmente las manejadas por FCGI o CGI
static-file.exclude-extensions = ( ".php", ".pl", ".fcgi", ".cgi" )

## Puerto por defecto
server.port              = 80

## Fichero Error 404
#server.error-handler-404 = "/error-handler.html"
## Crea fichero PID para Lighttpd
#server.pid-file         = "/var/run/lighttpd.pid"

#Directorios de Usuario
userdir.path = "public_html"
userdir.basepath = "/home/"
userdir.exclude-user = ("root")
##### virtual hosts

```

```

simple-vhost.server-root    = "/usr/pkg/share/httpd"
simple-vhost.default-host  = "localhost"
simple-vhost.document-root = "/htdocs"

include "vhosts.conf"

## change uid to <uid> (default: don't care)
server.username            = "www"

## change uid to <uid> (default: don't care)
server.groupname          = "www"

#Configuracion Basica para fcgi de PHP
fastcgi.server = ( ".php" =>
    ( "localhost" =>
        (
            "host" => "127.0.0.1",
            "port" => 1026,
            "bin-path" => "/usr/pkg/libexec/cgi-bin/php"
        )
    )
)

#### CGI module
cgi.assign                = ( ".pl"  => "/usr/pkg/bin/perl",
                              ".cgi" => "/usr/pkg/bin/perl" )

```

Configuracion Vhosts y htaccess.

```

### vhost.conf
Vhost simple
$HTTP["host"] == "Ejemplo.com" {
    server.document-root = "/usr/pkg/share/httpd/htdocs"
    server.errorlog = "/var/log/lighttpd/error.log"
    accesslog.filename = "/var/log/lighttpd/access_log"
    server.indexfiles = ( "index.php","index.html" )
    alias.url = ( "/phpmyadmin" => "/usr/pkg/share/phpmyadmin/" )
}
#Vhost con autentificacion
$HTTP["host"] == "autenticado.ejemplo.com" {
    server.document-root = "/home/vhost/autenticado/"
    server.errorlog = "/var/log/lighttpd/autenticado-error.log"
    accesslog.filename = "/var/log/lighttpd/autenticado-access.log"
    server.indexfiles = ( "index.html","index.php" )
    auth.backend = "htpasswd"
    auth.backend.htpasswd.userfile =
"/home/vhost/autenticado/lighttpd.user.htpasswd"
    auth.require = ( "/" =>
        (
            "method" => "basic",
            "realm" => "MAIL",
            "require" => "user=Usuario_auten"
        )
    )
}

```

```

    )
}
#Vhost con rewrite
$HTTP["host"] == "rewrite.ejemplo.com" {
    server.document-root = "/home/vhost/mediawiki-1.5.8"
    server.errorlog = "/var/log/lighttpd/wiki.com-error.log"
    accesslog.filename = "/var/log/lighttpd/wiki.com-access.log"
    server.indexfiles = ( "index.html", "index.php" )
    url.rewrite = ( "^(.*)/index.php\/(.*)" =>
"/$1/index.php?title=$2" )
#Añadiria lo de detras de la barra en la variable $2 esta es una
forma valida para hacer funcionar mediawiki.
}

```

(Cuidado si cortas y pegas esta configuracion es para NetBSD, las rutas cambiarán según el sistema operativo y la distribución que uses.)

Cooperación con Apache

En ocasiones, Lighttpd, debido a su capacidad de distribución de carga en la ejecución de FCGIs, es usado como servidor de apoyo en instalaciones con Apache y hacer de este un servidor más escalable y rápido en situaciones de carga. Para ello se usa mod_proxy y mod_rewrite de Apache. **Anurix** remienda este tipo de solución para ganar rendimiento y escalabilidad cuando tenemos todo nuestro sistema orientado a Apache, es decir, dependemos de algún tipo de Panel de Control que nos ata con Apache.

Algunos consejos para aumentar la eficiencia de Lighttpd

A continuación vamos a detallar algunos de los detalles de configuración , que en **Anurix** tenemos en cuenta para optimizar aún más si cabe el rendimiento de lighttpd.

Cuando tenemos un gran número de conexiones concurrentes deberíamos bajar el número de segundos que mantenemos vivas las conexiones con el cliente antes de que estas sean cerradas, de esta manera podemos bajar el número de descriptores de ficheros abiertos en el sistema, ya que en un ambiente de carga alta estos valores son importantes y deberían ser revisados y controlados respecto a una configuración por defecto , como siempre este tipos de valores requieren un estudio y un análisis previo.

```

server.max-keep-alive-requests = 4
server.max-keep-alive-idle = 4

```

Como anteriormente se ha comentado, la elección del *Eventhandler* adecuado es básico para un buen rendimiento del servidor, este en lighttpd se configura con la directiva server.event-handler = "Valor". El valor puede ser : select, poll, linux-rtsig, linux-sysepoll, solaris-devpoll o frebsd-kqueue. Ahora daremos una breve descripción de los mismos.

select() y *poll()*

Son por defecto las usadas en lighttpd y pueden funcionar en cualquier sistema operativo en los que esta soportado lighttpd , ya que son los *Eventhandlers* estándar en UNIX.

En algunos sistemas, como por ejemplo en FreeBSD, la syscall `select()` es mapeada mediante `lib_r` en la syscall `poll()`, por eso usa `select()` no siempre es la opción más óptima y más rápida, aunque existen alternativas a esta solución.

La configuración de Lighttpd ofrece la posibilidad de elegir como anteriormente hemos mencionado la opción más óptima.

rtsig() y epoll()

Son los *EventHandlers* que deberían ser usados cuando ejecutamos Lighttpd sobre linux, en concreto `rtsig` con kernels 2.4 y `epoll` en kernel 2.6 , esto ayudará a una ganancia sustancial en la respuesta con gran número de conexiones.

/dev/poll

Es el *EventHandler* que debería usarse en Solaris.

freebsd-kqueue

Es el óptimo para usar en FreeBSD y NetBSD. `Kqueue` es una evolución de las syscalls `poll()` y `select()`, ya que estas no escalan bien cuando el número de descriptors aumenta considerablemente, enlenteciendo de forma altamente notable la respuesta del servidor. `Kqueue` también puede manejar otro tipo de señales que `select` y `poll` no pueden, en definitiva `kqueue` es una capa de abstracción para el manejo de eventos dentro del sistema operativo, que mejora el rendimiento general del sistema. El soporte para este *eventhandler* en Lighttpd es por ahora experimental aunque es usable.

La opción de cambio de `Networkhandler` también mejorará el rendimiento, sobretodo en el envío de ficheros grandes. Se configura con la directiva `server.network-backend` y también va en función del sistema operativo que usemos.

Linux 2.4 `sendfile`

Linux 2.6 `sendfile64`

Solaris `sendfilev`

Freebsd `sendfile`

Número máximo de de conexiones

El número de descriptors de fichero máximo normalmente en la mayoría de sistemas operativos tipo UNIX está limitado por defecto a 1024 , si tenemos un tráfico grande en nuestro servidor esto puede ser modificado por Lighttpd para que no tengamos falta de descriptors de fichero y no podamos albergar más conexiones. En muchas ocasiones, los actuales administrador de **Anurix** se encontraron en otros trabajos con que en periodos vacacionales eran requeridos para reiniciar servicios web, debido generalmente a la alta demanda de descriptors. Aquí lo solucionamos mediante esta directiva:

```
server.max-fds = 2048
```

Esto sólo funciona si hemos arrancado Lighttpd como root, aunque también podemos configurar el número máximo de descriptores desde el sysctl en la mayoría de sistemas UNIX.

Stat() cache

Podemos mejorar los tiempos de respuesta y reducir los cambios de contexto forzando a usar FAM (File Alterator Monitor) a Lighttpd

Conclusión

Como hemos visto, LightTPD se muestra como una alternativa real, fiable, escalable, segura y veloz frente a otros servidores ya establecidos como Apache o IIS de Microsoft. **Anurix** ha comprobado con varios clientes cómo el rendimiento de sus portales web o ERPs dinámicos trabajaban de forma más eficiente al utilizar tecnología LightTPD. Si está pensando en cambiar la infraestructura software de sus servidores web, no dude en consultarnos, pues el cambio a LightTPD más la auditoría de código pertinente le permitirá obtener mayor beneficio de su hardware.