



Bisoños Usuarios de GNU/Linux de Mallorca y Alrededores | Bergantells Usuaris de GNU/Linux de Mallorca i Afegitons

MPD: un esquema diferente para un reproductor de música (18956 lectures)

Per Joan Ginard, [lumen](http://www.bulma.net) (<http://www.bulma.net>)

Creado el 07/08/2006 14:47 modificado el 07/08/2006 14:47

El proyecto MPD plantea e implementa un reproductor de música con el paradigma cliente servidor. Se explica en este artículo su idea básica y las posibilidades que ofrece.

Muchos de nosotros, al pensar en una aplicación de reproducción de música, pensamos en una aplicación normalmente sobre X, del que esperamos que pueda leer la música de nuestro equipo y reproducirla. Esperamos también que nos ofrezca algunas intrucciones de control sobre la reproducción (next, previous, fast forward...) y operaciones sobre listas de reproducción. Así se comportan básicamente Xmms, Beep-media-player, Kboogle, Xine etc.

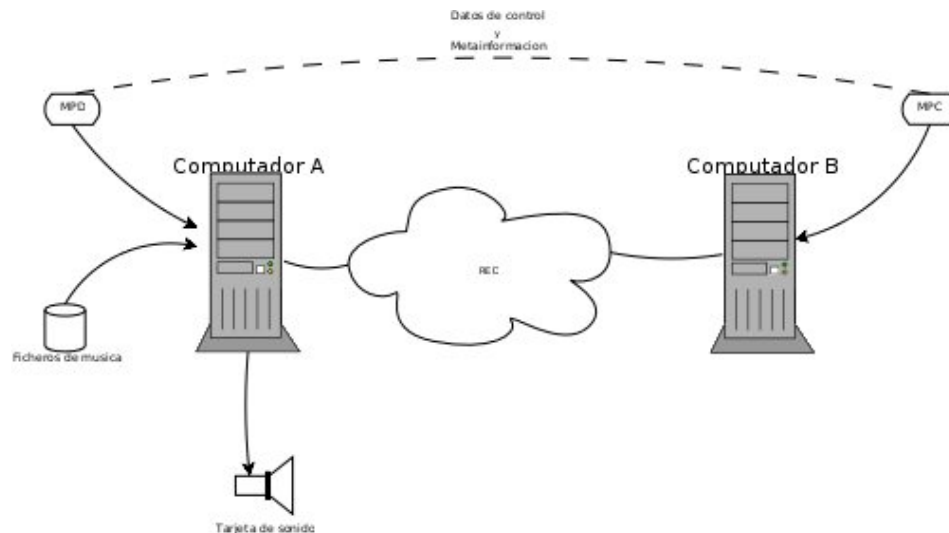
Algunos reproductores, como [Amarok](#) o [Juk](#), van un poco más allá y se esfuerzan en conseguir la gestión de la propia música que tenemos guardada en el equipo, permitiendo ordenarla, realizar búsquedas o transferirla a otros equipos.

Ese planteamiento tiene varios problemas

- ◆ Los equipos sin un sistema de ventanas no pueden utilizar el programa. Es decir, si no tienes arrancado un escritorio o windowmanager (gnome, kde, xfce ...) jamás podrás usar el Amarok, Juk o xmms.
- ◆ Derivada de la anterior, cada vez que el equipo necesita reiniciar el sistema de ventanas, la aplicación de música debe terminar también y por tanto debe suspenderse también la reproducción. Fijaos en el problema: a pesar de no haber ningún problema con alsa, la tarjeta de sonido o la aplicación de reproducción, la reproducción debe suspenderse debido a cualquier problema en el sistema gráfico. Esta dependencia es un poco aberrante y también algo molesta si precisamente queremos escuchar música mientras estamos experimentando algo con las X.
- ◆ Se juntan en la misma aplicación dos tareas inicialmente distintas: la de reproducir ficheros y la de controlar los ficheros que se deben reproducir. Eso causa que cada aplicación que necesite reproducir ficheros, utilice su propio sistema de reproducción de ficheros. Por ejemplo, Xmms, BMP, Xine o Vlc utilizan su propio sistema de descompresión de música, mientras que Amarok usa directamente otro reproductor como Xine o los demonios Artds o GStreamer. Qué ocurre si nos encanta la interface simplona de BMP pero no su reproductor. Ahí está el problema, la interface para el control y el reproductor en sí van obligatoriamente de la mano, cuando en principio eso no es necesario.
- ◆ Ese enfoque nos obliga a que el sistema de sonido, se encuentre en el mismo equipo que los propios ficheros de música y además que también podamos ejecutar ahí la aplicación de reproducción.

MPD y su arquitectura

Cliente Servidor ===== La idea de MPD es utilizar el [modelo cliente servidor](#). Separar por un lado el reproductor de audio junto con los ficheros de música y por otro la aplicación que controla qué reproducir. Esquemáticamente:



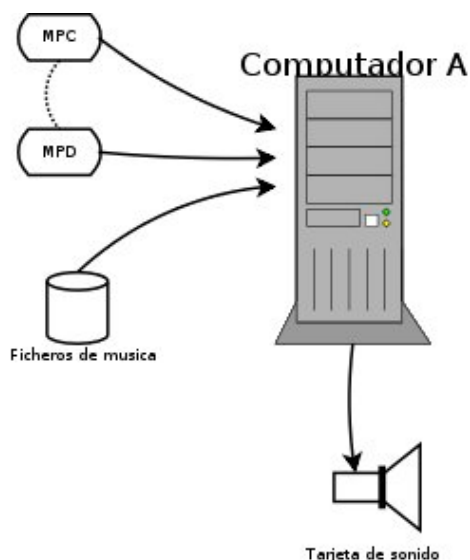
MPD (Music Player Daemon) se encarga de descomprimir los ficheros de música locales y reproducirlos por la tarjeta de sonido. MPC (Music Player Client) se comunica con MPD mediante [sockets](#) (por la red). Entre ellos intercambian la información sobre qué música está disponible para ser reproducida y también las instrucciones de control típicas (play, stop, prev, next...).

MPD se encarga de: * Descomprimir música y reproducirla * Descubrir qué música está disponible. * Informar de los que está haciendo a los MPC que lo soliciten. * Realizar las intrucciones que reciba de los MPC.

MPC no tiene un objetivo determinado. Precisamente esa es uno de sus potenciales. Lo primero que debemos pensar es que MPC no tiene porqué haber sólo uno. Puede haber varios MPC realizando diferentes funciones contra el mismo MPD. Normalmente los MPC hacen:

- ◆ Presentar de forma gráfica la música disponible para ser reproducida.
- ◆ Gestionar la lista de reproducción.
- ◆ Proporcionar controles sobre la lista de reproducción (sobre lo que reproduce el MPD).
- ◆ Realizar estadísticas.
- ◆ Informar de tus gustos musicales a otras personas.
- ◆ Comportarse como interface a otros dispositivos sin cables como mandos a distancia por infrarojos, bluetooth o access keys para que MPD pueda ser controlado fácilmente.

Por supuesto, en el gráfico el Computador B puede ser un portátil, o una PDA o lo que se os ocurra. Y también, como podeis suponer, no tiene porqué haber dos computadores. Se puede utilizar la interfaz de socket por la interface loopback:





Des esta segunda manera, MPD y MPC se pueden comportar como cualquier reproductor tipo Xmms, con la salvedad de que matar MPC (la parte visual), no afecta para nada a que la música siga su curso.

Ventajas del modelo cliente-servidor

- ♦ El reproductor en sí y el controlador se mantienen totalmente diferenciados e independientes. Su comunicación queda estandarizada a través del protocolo de comunicación de nivel de aplicación.
- ♦ El servidor puede ser considerado un servicio de sistema y gestionado como tal.
- ♦ Al ser la comunicación por red, el cliente gana ubicuidad, el control de la música se puede realizar desde cualquier otro ordenador de la red.
- ♦ La tarea del servidor requiere muy pocos recursos. Descomprimir música es muy sencillo y mantener un listado de ficheros disponibles también. Recordad que las búsquedas se pueden realizar en la parte de cliente. Eso hace que el servidor pueda correr en máquinas con muy pocos recursos.
- ♦ Para soportar nuevos formatos de reproducción, es suficiente añadir soporte en el servidor.
- ♦ Se gana independencia en el cliente. Podemos elegir el cliente que más nos agrade o lo que es mejor, usar varios de ellos a la vez.

Dos comentarios: 1. Este enfoque exige añadir en el servidor un sistema de autenticación. El servidor mantiene un puerto habierto al abismo de la red (quien sabe se Inet). Así que hay que tener cuidado con lo que se hace. La implementación actual que se utiliza en el demonio MPD no me parece muy fiable. 2. Es necesario que el servidor almacene de alguna manera el listado de la música disponible para ser reproducida. Eso implica operaciones para crear y refrescar esa lista. La implementación actual que realiza el MPD no es bastante buena en mi opinión y no creo que de muy bien rendimiento con grandes cantidades de ficheros de música. Supongo que usar algún BD ligera como sqllite no sería muy difícil a la vez que bastante más apropiado. Creo recordar que existe algún fork para incluir ese soporte.

Aplicaciones

El enfoque de MPD-MPC permite escuchar la música como hasta ahora lo veníamos haciendo. De hecho [XMMS2](#) tiene un enfoque también del estilo cliente-servidor. Entonces sus aplicaciones son por lo menos las mismas que las de cualquier reproductor de música: escuchar música. Pero ahora tenemos más posibilidades:

- ♦ Crear un media center, es decir, un equipo con pocos recursos, tarjeta de red, de sonido, amplificador y altavoces, un buen disco duro y sin monitor (ni X). El control de la música a reproducir lo podemos controlar desde cualquier otro ordenador de la red, por ejemplo nuestro portátil, sin renunciar al buen trato que nos proporciona una buena GUI.
- ♦ Una variación de lo anterior son bares o locales con música ambiental, donde el control de la música a reproducir no pueda hacerse desde donde se encuentra realmente la instalación de audio.
- ♦ Por supuesto existe un MPC de línea de comandos, que nos permite realizar comandos del tipo 'mpc play' 'mpc next', 'mpc volumen +10' etc. Entonces, es muy fácil conectar esos comandos con eventos de teclas abreviadas del teclado, del mouse o de mandos a distancia.
- ♦ Añadiendo una pieza más (el streamer), se puede crear una radio personalizada en internet y controlada remotamente, que nos hace disponible en cualquier sitio la música que tenemos en casa.

En breve publicaré un el trquito sobre cómo hacerlo.

Algunos MPC

En el wiki de MPD mantienen una [lista actualizada de MPC's](#). Como he dicho antes, hay mpc para línea de comandos, pero también hay uno para N-curses y otros más gráficos como pypmd (phyton) o gmpc con interfaces gráficas para los más refinados. También es muy interesante tener siempre en marcha mpdscribble, que sondeará el MPD para informar a lastfm de lo que estais escuchando. Incluso guarda en caché lo que habeis reproducido por si no dispone de conexión a Inet! O incluso existe [arthist](#) para actualizar tu weblog a partir de lo que escuches o varios interfaces que se integran con apache para controlar la música desde el web. Echadle un vistazo a todos, valen la pena.



Poniéndolo en marcha

Poner en marcha un servidor MPD es trivial. Casi siempre las configuraciones por defecto os irán perfectamente. En su fichero de configuración `~/mpdconf` encontrareis parámetros para el driver de reproducción (oss o alsa), tamaños de buffer, ficheros de log y otros varios.

Además, permite gestionar el sistema de autenticación y permisos para los clientes que permitamos conectar al servidor. MPD necesita básicamente que se le especifique el puerto en el que escuchar, dónde se encuentra nuestra música y dónde colocar el fichero de su "base de datos".

```
port                                "6600"
music_directory                    "~/musica"
playlist_directory                  "~/mpd/playlists"
log_file                           "~/mpd/mpd.log"
error_file                         "~/mpd/mpd.error"
db_file                            "~/mpd/mpd.db"
state_file                         "~/mpd/mpdstate"
mixer_type                         "alsa"
mixer_device                       "default"
mixer_control                      "Master"
ao_driver                          "alsa09"
ao_driver_options                  "dev=hw:0,0"
```

Me queda probar lo del soporte para [Replaygain](#). Sirve para garantizar que todas las canciones sonarán a un mismo volumen. Antes de lanzar el servidor por primera vez, teneis que lanzarlo con el comando `--create-db` para que genere la base de datos de música disponible. El demonio lo puede lanzar el sistema, por ejemplo en el arranque con el `init.d` (en Debian ya os dan los scripts) o también no hay ningún problema en que lo lance el usuario que desea utilizarlo. En ese caso podeis añadir la llamada a mpd en vuestro `~/bash_profile`

A los MPC que elijais, simplemente hay que indicarles en qué host y puerto pueden encontrar al servidor y qué usuario y contraseña utilizar en el caso de que useis autenticación.

E-mail del autor: lumen_ARROBA_bulma.net

Podrás encontrar este artículo e información adicional en: <http://bulma.net/body.phtml?nIdNoticia=2327>