OFFICE MUSIC SERVER

Using Music Player Daemon and Icecast

James Pinkster

29 June 2015 / MLUG

Introduction

In the office we share a stereo that people use to plays CD's or songs from an iPod/AUX input device. Throughout the day only few people would put music on.

Introduction

In the office we share a stereo that people use to plays CD's or songs from an iPod/AUX input device. Throughout the day only few people would put music on.

Having a chat with a colleague, we thought it would be cool to have a global playlist that everyone had access to and could add songs to be played through the stereo.

Introduction

In the office we share a stereo that people use to plays CD's or songs from an iPod/AUX input device. Throughout the day only few people would put music on.

Having a chat with a colleague, we thought it would be cool to have a global playlist that everyone had access to and could add songs to be played through the stereo.

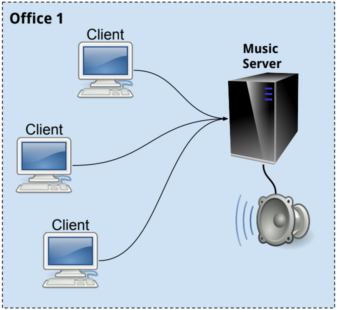
An interweb search revealed that the Music Player Daemon (MPD) could be just what we were looking for.

Music Server Overview

Music Server

- Runs the Music Player Daemon
- Connects directly to stereo
- All clients and server are on same network subnet.

Diagram



Music Server

MPD Installation

Install Debian packages

```
# aptitude install mpd alsa-utils
```

Set mixer levels and ensure channels are unmuted.

```
# alsamixer
```

Store mixer settings

```
# alsactl store
```

Add mpd to group audio

```
# adduser mpd audio
```

Music Server

MPD Configuration

\etc\mpd.conf

```
music_directory    "/zdata/music"
playlist_directory "/var/lib/mpd/playlists"
db_file            "/var/lib/mpd/tag_cache"

log_file           "/var/log/mpd/mpd.log"

user               "mpd"
group              "audio"

bind_to_address    "127.0.0.1"
bind_to_address    "192.168.1.166"
port               "6600"
```

Music Server

MPD Configuration

etc**mpd.conf** continued

```
auto_update        "yes"

max_connections    "30"

audio_output {
    type           "alsa"
    name           "MPD ALSA Device"
}
```

MPD Clients

A few of the MPD clients we tried.

mpc	basic command line client
ncmpcpp	OSX client
Theremin	icecast
Cantata	Qt client, runs on Linux, OSX and Windows

Cool it works!

We can all control the global playlist and keep the tunes going.

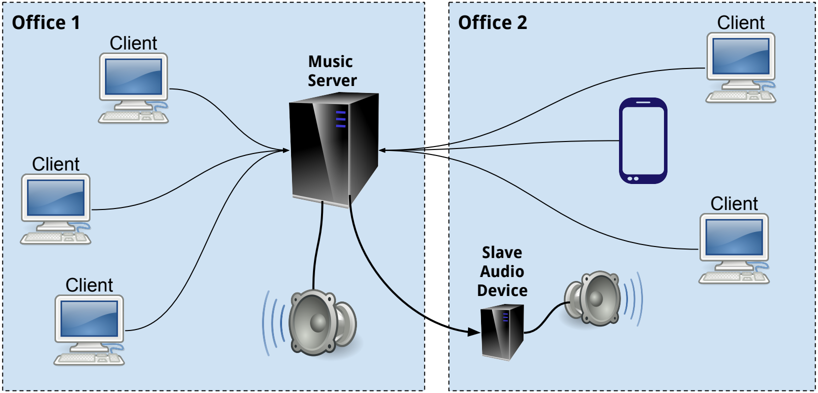
Cool it works!

We can all control the global playlist and keep the tunes going.

Wait...

What about the people in the other section of the office. Maybe they would like to play some music also.

Music Server and Music Slave Diagram



Music Server

Icecast Configuration

Install additional Debian packages on Music Server

```
# aptitude install icecast2
```

Configure icecast2 upon installation

Icecast2 hostname:	192.168.1.166
Icecast2 source password:	icecast
Icecast2 relay password:	icecast
Icecast2 administration password:	icecast

Music Server

Update MPD Configuration With Icecast audio_output

\etc\mpd.conf

```
audio_output {
    type           "alsa"
    name           "MPD ALSA Device"
}
```

Music Server

Update MPD Configuration With Icecast audio_output

\etc\mpd.conf

```
audio_output {
    type             "alsa"
    name             "MPD ALSA Device"
}

audio_output {
    type             "shout"
    encoding         "ogg"
    name             "MPD Shout Stream"
    host             "192.168.1.166"
    port             "8000"
    mount            "/mpd.ogg"
    password         "icecast"
    quality          "5.0"
    format           "44100:16:2"
    user             "source"
}
```

Slave Audio Device

Setup Raspberry Pi

Install Arch packages

```
 # pacman -S mplayer tmux
```

Create regular user pi that belongs to group audio

```
 # useradd -m -G audio -s /bin/bash pi
 # passwd pi
```

Get systemd to auto login user pi on startup
\etc\systemd\system\getty@tty1.service.d\override.conf

```
[Service]
ExecStart=
ExecStart=-/sbin/agetty --autologin pi --noclear %I 38400 linux
Type=idle
```

Slave Audio Device

Configure Mplayer To Connect To Icecast Stream

Script to run mplayer indefinitely
\home\pi\bin\run_mplayer

```
#!/bin/sh

# Keep trying to run mplayer,
# as MPD will stop the stream once a playlist has completed

while [ 1 ]
do
  mplayer -ao alsa -prefer-ipv4 -playlist \
    http://192.168.1.166:8000/mpd.ogg.m3u
  sleep 15
done
```

Slave Audio Device

Configure Mplayer To Connect To Icecast Stream

Run mplayer in a detached TMux session on startup
\home\pi\.bash_profile

```
# Start mplayer for Icecast stream

TMUX_PID=$(pidof tmux)
if [ "${TMUX_PID}x" == "x" ]; then
    echo "Starting tmux with mplayer"
    tmux new-session -d /home/pi/bin/run_mplayer
else
    echo "tmux already running on ${TMUX_PID}"
fi
```

Things To Do

- Hack on Hubot mpd plugin to use with Slack

Music Server Summary

Music Server

- Running Debian Jessie
- Packages: mpd, icecast2 and alsa-utils
- Connected to stereo by AUX input.
- All clients and server are on same LAN subnet.
- MPD client used: Cantata

Music Slave

- Arch linux on Raspberry Pi
- Packages: mplayer and tmux installed
- Connected to another set of speakers in office no. 2

Music Server Demo

Caveats

MPD

- Would try to bind IPv6 be default. Explicitly list IPv4 addresses.
- max_connections defaulted to 5. We hit that limit quickly.
- enable auto_update so client are able to get MPD to update its database.

Icecast

- The Icecast daemon must start before the mpd daemon.

mplayer

- When the MPD playlist reaches the end and stops. It caused mplayer to exit

Slave Audio Device

- Slight delay when pausing or changes tracks on slave audio output due to icecast stream buffering.

References

http://www.musicpd.org https://wiki.archlinux.org/index.php/Streaming_With_Icecast