

При помощи ffmpeg

Описание

ffmpeg - это мощный кроссплатформенный инструмент для обработки и публикации видео- и аудиоконтента. С точки зрения публикации RTMP-потока, ffmpeg позволяет

- очень гибко настраивать параметры кодирования потока;
- передать серверу параметры RTMP-соединения.

Краткое руководство по тестированию

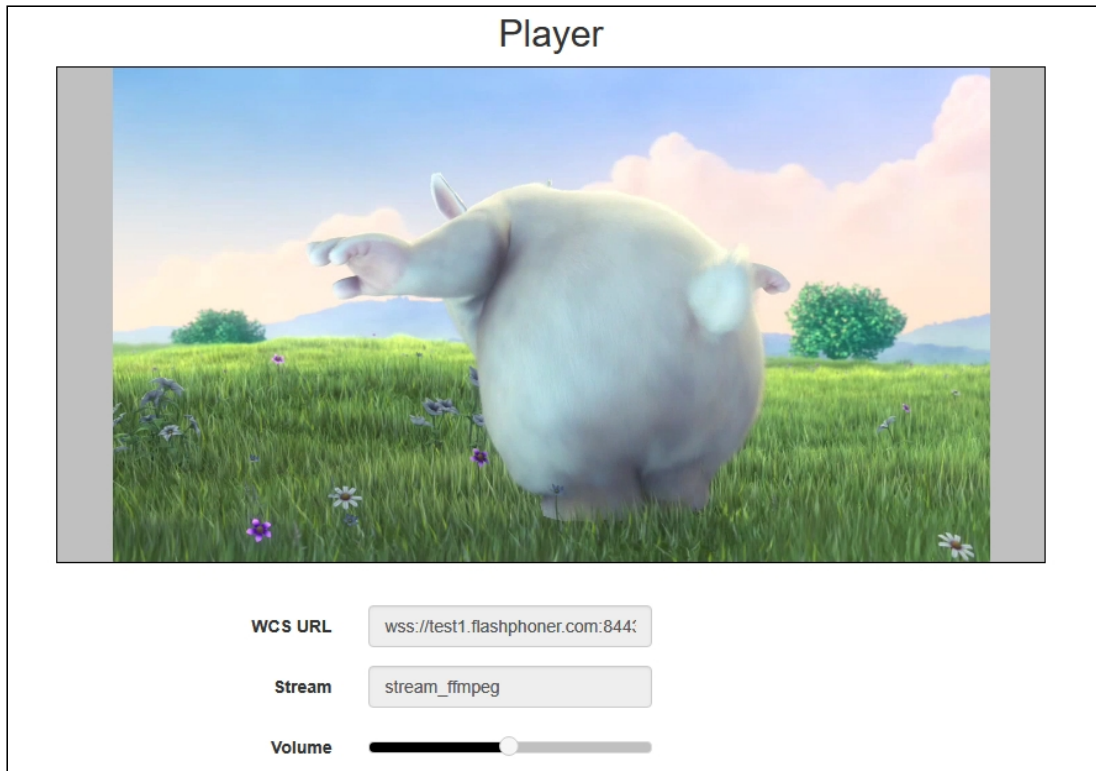
1. Для теста используем:
2. WCS сервер
3. ffmpeg
4. веб-приложение [Player](#) в браузере Chrome для воспроизведения потока
5. Запустите ffmpeg следующим образом

```
ffmpeg -re -i BigBuckBunny.mp4 -preset ultrafast -acodec aac -vcodec h264 -strict -2 -f flv rtmp://test1.flashphoner.com:1935/live/stream_ffmpeg
```

Здесь

6. `BigBuckBunny.mp4` - транслируемый ролик
7. `test1.flashphoner.com` - WCS-сервер
8. `stream_ffmpeg` - имя потока для публикации на сервере
9. Откройте приложение Player в браузере `https://test1.flashphoner.com:8888/client2/examples/demo/streaming/player/player.html` где `test1.flashphoner.com` - WCS-сервер. Укажите имя потока и

нажмите **Play**. Начнется воспроизведение потока:



Публикация Sorenson Spark + Speex 16 kHz потока в контейнере FLV

WCS сервер принимает RTMP поток, закодированный в Sorenson Spark + Speex 16kHz в контейнере FLV. Такой поток можно опубликовать, например, при помощи ffmpeg следующим образом:

```
ffmpeg -re -i BigBuckBunny.flv -preset ultrafast -ar 16000 -ac 1 -acodec  
speex -vcodec flv -strict -2 -f flv  
rtmp://test1.flashphoner.com:1935/live/test
```

Ограничения

1. Для дальнейшей обработки на сервере, в том числе для записи, такой поток будет транскодирован в H.264 + AAC.
2. При публикации в SDP для видео и для аудио должны быть указаны payload type 127 и 97 соответственно, например

```
v=0  
o=- 1988962254 1988962254 IN IP4 0.0.0.0  
c=IN IP4 0.0.0.0  
t=0 0  
a=sdplang:en
```

```
m=video 0 RTP/AVP 127
a=rtpmap:127 FLV/90000
a=sendonly
m=audio 0 RTP/AVP 97 8 0
a=rtpmap:97 SPEEX/16000
a=rtpmap:8 PCMA/8000
a=rtpmap:0 PCMU/8000
a=sendonly
```

Дополнительные возможности

Явное указание параметров кодирования

При публикации потока ffmpeg позволяет явно задавать параметры кодирования, например

```
ffmpeg -re -i BigBuckBunny.mp4 -acodec aac -vcodec libx264 -f flv -ar 44100
rtmp://127.0.0.1:1935/live/stream_ffmpeg
```

Более сложный случай с добавлением звуковой дорожки к файлу, не имеющему ее изначально (в примере используется источник `/dev/zero`)

```
ffmpeg -re -f lavfi -i "movie=filename=test.mov:loop=0,
setpts=N/(FRAME_RATE*TB)" -an -s 1280x720 -r 15 -f rawvideo -pix_fmt yuv420p
- | ffmpeg -thread_queue_size 512 -an -f rawvideo -pix_fmt yuv420p -r 15 -s
1280x720 -i - -vn -f s16le -acodec pcm_s16le -ac 2 -i /dev/zero -r 15 -c:v
libx264 -tune zerolatency -profile:v baseline -g 60 -b:v 960k -s 1280x720 -
c:a aac -b:a 64k -f flv rtmp://127.0.0.1/live/test_video
```

Передача параметров серверу при установке RTMP-соединения

При публикации потока ffmpeg позволяет указать в опции `-rtmp_conn` параметры RTMP-соединения, которые должны быть переданы серверу, например:

```
ffmpeg -re -i BigBuckBunny.mp4 -f flv -rtmp_conn "0:1
NS:appKey:flashStreamingApp NS:name:12121212 NS:stream:12121212 N0:custom:0:1
NS:auth:22222222 NS:stream:33333333 0:0 0:0"
rtmp://test1.flashphoner.com:1935/12121212
```

Здесь передаются следующие параметры

- ключ приложения на сервере `flashStreamingApp`
- имя приложения на сервере `12121212`
- имя потока на сервере `12121212`
- объект `custom` с дополнительными данными:

```
{  
  "auth" : "22222222",  
  "stream" : "3333333"  
}
```

Поворот изображения потока, опубликованного при помощи ffmpeg

RTMP кодировщик ffmpeg дает возможность [отправить метаданные ориентации изображения серверу](#) при помощи ключей командной строки:

```
ffmpeg -i input.mp4 -metadata:s:v rotate=90 -vcodec copy -acodec copy -strict  
-2 -f flv rtmp://test1.flashphoner.com:1935/live/stream_ffmpeg
```

Отметим, что настройка поворота для ffmpeg указывается в градусах, при этом на сервер передается соответствующее значение поля orientation.